



**Ressourcen-Provisionierungsverfahren zur Erweiterung
energetischer Freiheitsgrade von Rechenzentren unter
Veränderung der IT-Last**

Von der Fakultät II für Informatik, Wirtschafts- und
Rechtswissenschaften der Carl von Ossietzky Universität Oldenburg zur
Erlangung des Grades und Titels eines

Doktors der Naturwissenschaften (Dr. rer. nat)

angenommene Dissertation

von Herrn Alexander Borgerding

Gutachter: Prof. Dr.-Ing. Wolfgang Nebel
Weiterer Gutachter: Prof. i.R. Dr. Michael Sonnenschein
Tag der Disputation: 15.02.2021

Danksagung

Der Abschluss einer Promotion ist eine ganz besondere Herausforderung und geht zweifelsohne nicht nur auf eine Individualleistung zurück, sondern auch auf zahlreiche Personen, die mich auf diesem Weg begleitet und unterstützt haben.

An erster Stelle danke ich meinem Doktorvater Herrn Prof. Dr.-Ing. Wolfgang Nebel für seine methodische Unterstützung, ermutigenden Ratschläge und Ideen sowie das in mich gesetzte Vertrauen und sein Engagement beim Überwinden diverser Hindernisse. Ich danke Herrn Prof. i.R. Dr. Michael Sonnenschein, der meine Arbeit auch im Ruhestand weiter begleitet hat und insbesondere während der Einschränkungen durch die Corona-Maßnahmen ein wichtiger Diskussionspartner war.

Als Externer bin ich herzlich in der Gruppe „SRI“ des OFFIS aufgenommen worden und konnte an vielen spannenden Themen partizipieren und habe selbst auch viel Unterstützung von der Gruppe erhalten, hierfür danke ich der ganzen Gruppe „SRI“ bzw. „Energieeffiziente Smart Cities“ und den Gruppenleitern Dr. Gunnar Schomaker und Dr.-Ing. Sven Rosinger.

Besonderer Dank gebührt an dieser Stelle meiner Familie und meinem Freundeskreis. Neben dem Verständnis für reduzierte Freizeitkapazität, den offenen Ohren für meine Herausforderungen und hilfreichen Tipps wurde mir auch der Rücken für das Projekt „Dissertation“ freigehalten. Insbesondere danke ich Stephan, Andreas und Bernd.

Zusammenfassung

Durch die voranschreitende Digitalisierung und Etablierung neuer IT-Services ist ein steigender Bedarf an Rechenzentrumsdienstleistungen zu verzeichnen, folglich steigt auch der Energiebedarf in diesem Bereich an und stellt ein relevantes Volumen im Strommarkt dar. Im Energiesektor rückt der Fokus verstärkt auf die erneuerbaren und klimaneutralen Energien. Hierbei ergibt sich neben der Bereitstellung der benötigten Kapazitäten auch die Herausforderung, eine kontinuierliche Stromversorgung sicherzustellen. Je höher der Anteil an volatilen Energielieferanten steigt, umso aufwendiger wird auch die entsprechende Kompensationsleistung.

Ein Ansatz, den Kompensationsbedarf zu reduzieren, ist es, den Energiebedarf entsprechend der aktuellen Verfügbarkeit zu steuern. Der Energiebedarf von Rechenzentren stellt dabei ein attraktives, wachsendes Volumen für diese Aufgabe dar. Allerdings lässt sich die Leistungsaufnahme eines nativen Rechenzentrums nicht oder nur unzureichend steuern. Maßgeblich wird die Leistungsaufnahme von der Rechenleistung bestimmt, die durch die jeweils betriebenen Applikationen abgerufen wird. In dieser Arbeit wird ein Verfahren vorgestellt, das es ermöglicht, die abgerufene Rechenleistung dennoch zu beeinflussen und die energetischen Freiheitsgrade von Rechenzentren zu erweitern.

Auf Basis einer Ressourcen-Überprovisionierung wird ein heuristisches Verfahren vorgestellt, um die Performance der betriebenen Applikationen zu beeinflussen und damit auch die Leistungsaufnahme der Server zu steuern. Mit Server-Virtualisierung als Basistechnologie werden virtuelle Maschinen als Vehikel zum Verschieben von Rechenlast verwendet und so eine Serverauslastung erzeugt, die einer geforderten Leistungsaufnahme entspricht. Mit diesem Ansatz kann eine Steuerung der Server-Leistungsaufnahme erreicht werden, ohne dass Zugriff auf die betriebenen Applikationen benötigt wird oder die fachliche Aufgabe der Applikationen durchdrungen werden muss.

Das in dieser Arbeit entwickelte Online Heuristic Machine Allocation (OHMA)-Verfahren wird als Automaten-Modell in der Simulationsumgebung Uppaal modelliert, um eine transparente Prozessdarstellung zu ermöglichen, und wird anhand von Datensätzen aus der Industrie evaluiert. Es wird gezeigt, dass das heuristische Verfahren vergleichbare Ergebnisse liefert wie bekannte Ansätze, die jedoch auf Scheduling-Technologie basieren und einer fachlichen Implementierung in den betriebenen Applikationen bedürfen.

Das OHMA-Verfahren kann mit linearer Komplexität beschrieben werden und zeigte in synthetischen Experimenten vergleichbare Ergebnisse wie eine bekannte Optimal-Lösung. Es wird eine Leistungsanpassung von bis zu 50% mit der Technik einer Ressourcen-Überprovisionierung in Kombination mit der Migration virtueller Maschinen erreicht.

Abstract

Due to the advancing digitalization and establishment of new IT services, there is a growing demand for data center services, and at the same time the energy demand in this area is also increasing, thus representing a relevant volume in the electricity market. In the energy sector, the focus is now increasingly shifting to renewable and climate-neutral energies. In addition to providing the required capacities, this also poses the challenge of ensuring a continuous supply of electricity. The availability of solar and wind energy can vary greatly. In order to balance out these fluctuations, several types of power plants are always combined for stable grid operation. The higher the proportion of volatile energy suppliers increases, the more expensive the corresponding compensation line becomes.

One approach to reducing the need for compensation is to manage energy demand according to current availability. The energy requirements of data centers represent an attractive, growing volume for this task. However, the power consumption of a native data center cannot be controlled or only insufficiently controlled. The power consumption is largely determined by the data processing line, which is accessed by the applications running in each case. In this thesis, a method is presented which allows to influence the retrieved data line nevertheless and to extend the energetic degrees of freedom of data centers.

Based on resource over-provisioning, a heuristic method is presented for affecting the performance of the applications being operated and thus also controlling the power consumption of the servers. With server virtualization as the base technology, virtual machines are used as a vehicle to shift computational load, creating a server workload that corresponds to a required power consumption. With this approach, control of server power consumption can be achieved without requiring access to the applications being operated or having to penetrate the business task of the applications.

The Online Heuristic Machine Allocation (OHMA) approach developed in this thesis is modeled as an automaton model in the simulation environment Uppaal to enable a transparent process representation, and is evaluated using data sets from industry. It is shown that the heuristic method provides comparable results to known approaches, which, however, are based on scheduling technology and require a specialized implementation in the operated applications.

The OHMA approach can be described with linear complexity and showed comparable results to a known optimal solution in synthetic experiments. A performance adjustment of up to 50% is achieved with the technique of resource overprovisioning in combination with virtual machine migration.

Inhaltsverzeichnis

1. Einleitung und Motivation	1
1.1. Motivation	2
1.2. Nutzungsszenario	5
1.3. Anwendungsfälle und Forschungsfragen	7
1.4. Gliederung der Arbeit	8
1.5. Zusammenfassung	9
2. Grundlagen	11
2.1. Das Rechenzentrum	11
2.1.1. Leistungsaufnahme der Infrastruktur	12
2.1.2. Klimatisierung	14
2.2. Leistungsaufnahme eines Servers	15
2.3. Serverressourcen: Betriebsmittel und deren Charakteristika	16
2.3.1. Prozessor	16
2.3.2. Arbeitsspeicher	17
2.3.3. Festplatte	18
2.3.4. Netzwerkkarte	20
2.4. Servervirtualisierung	20
2.4.1. Hypervisor	21
2.4.2. Migrationsverfahren	21
2.4.3. Kosten einer Live-Migration	22
2.5. Service Level Management (SLM)	23
2.6. Effizienzbewertung von Rechenzentren	23
2.6.1. Effizienz von Servern	24
2.7. Regelleistung	24
2.8. Zusammenfassung	28
3. Stand der Technik	31
3.1. Low-Power Technologie	31
3.2. Throttling	32
3.3. Server-Napping	32
3.4. Optimierung der Server-Klimatisierung	32
3.5. Scheduling-Verfahren zur Steuerung der Server-Leistungsaufnahme	33
3.6. Geografische Migration zur Nutzung lokaler Standortvorteile	34
3.7. Einsatz von Server-Virtualisierungstechnologie	35
3.7.1. Effiziente Nutzung des Arbeitsspeichers	35
3.7.2. Optimierung der Netzwerkkommunikation	36

Inhaltsverzeichnis

3.8. Verwendung von Prognosen zur Kapazitätsplanung	37
3.9. Übersicht zu aktuell verfügbaren Verfahren	37
3.10. Zusammenfassung	39
4. Problemstellung und Abgrenzung	41
4.1. Technologieansatz	42
4.1.1. Prozessor-Ressourcen als Steuerungsinstrument	45
4.2. Abgrenzung	46
4.3. Effizienz und Skalierungseffekte	47
4.4. Komplexität der Aufgabenstellung	47
4.5. Beschreibung des Problems einer Leistungsänderung in Teil-Schritten . . .	48
4.5.1. Ziel-Leistungsaufnahme	48
4.5.2. Teilschritte	49
4.6. Anforderungen	52
4.6.1. Fachliche Anforderungen	52
4.6.2. Nicht-funktionale Anforderungen	52
4.7. Zusammenfassung	53
5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)	55
5.1. Beschreibung des Verfahrens	55
5.1.1. System-Architektur	56
5.2. Voraussetzungen und Annahmen	59
5.2.1. Dynamische Leistungsaufnahme	59
5.2.2. Virtualisierungstechnologie	60
5.2.3. Weitere Randbedingungen	61
5.2.4. Fachliche Voraussetzungen	62
5.3. Modellierung des Verfahrens in einer Uppaal-Umgebung	63
5.3.1. Simulationsframework Uppaal	63
5.4. Modell zur Abbildung einer VM	65
5.4.1. Verhalten einer VM und Ablauf einer Migration	66
5.5. Modell zur Abbildung einer PM	68
5.5.1. Auslastung einer PM	68
5.5.2. Verhalten einer PM	71
5.5.3. Leistungsaufnahme der PM	72
5.5.4. Migrationszeiten	73
5.6. SLM-Komponente	74
5.6.1. Test-Komponente	74
5.7. Auswahl einer VM und Migrationsstrategie	75
5.7.1. Erhöhen der Leistungsaufnahme	77
5.7.2. Absenken der Leistungsaufnahme	79
5.8. Flussdiagramm zum OHMA-Verfahren	80
5.9. Kenngrößen, Dimensionierung und Konfiguration	80
5.9.1. Leistungsreduzierung	80
5.9.2. Aufteilung der Server in Gruppen	80
5.9.3. Konvergenzverhalten	83

5.10. Zusammenfassung	88
6. Potentialanalyse und Evaluation	91
6.1. Ergebnisse	91
6.1.1. Absenken der Leistungsaufnahme	92
6.1.2. Erhöhen der Leistungsaufnahme	94
6.2. Komplexitätsbetrachtung	94
6.2.1. Berechnungsdauer	96
6.3. Bewertungs-Metriken	97
6.3.1. Effizienz und Skalierbarkeit	97
6.3.2. Effektivität	100
6.4. Synthetische Experimente	102
6.5. Einsetzbarkeit des Verfahren im Kontext von Fachapplikationen	104
6.5.1. Billing-Prozess eines Telekommunikationsunternehmens	106
6.5.2. Provisionierungsprozesse	107
6.5.3. Grenzen des Verfahrens	108
6.6. Vergleich mit anderen Verfahren	108
6.7. Zusammenfassung	110
7. Zusammenfassung und Ausblick	113
7.1. Erkenntnisse aus der Bearbeitung der Forschungsfragen	113
7.2. Ausblick	115
7.2.1. Integration von RAM-Ähnlichkeitsprüfungen in das OHMA-Verfahren	116
7.2.2. Adaption zur Nutzung von Docker-Containern	116
7.2.3. Integration thermischer Effekte	116
7.2.4. Weiteres Nutzungspotential durch Microservices	117
A. Anhang	119

1. Einleitung und Motivation

Das aktuelle Energiekonzept der Bundesregierung [Pre12] sieht einen zügigen Ausbau der erneuerbaren Energien und den Ausstieg aus der Kernenergie bis Ende des Jahres 2022 vor. Um diese Ziele zu erreichen, soll der Anteil erneuerbarer Energien an der Stromerzeugung auf mindestens 35% des Strombedarfs im Jahr 2020 steigen. Bis 2030 strebt die Bundesregierung einen Anteil von 50% an, 2040 sollen es 60% und 2050 schließlich 80% sein. Weiterhin ist das „Gesetz zur Reduzierung und zur Beendigung der Kohleverstromung“ beschlossen worden [Pre19a], welches einen schrittweisen Ausstieg aus der Stromerzeugung mit Kohle bis Ende des Jahres 2038 [Pre19b] vorsieht.

Durch die Erhöhung der Anteile von erneuerbaren Energien wie Solar- oder Windenergie ergeben sich diverse technische Herausforderungen. Insbesondere kann die produzierte Leistung in Abhängigkeit von Sonneneinstrahlung und Wind stark schwanken und ist nicht in dem Maße plan- und steuerbar, wie es bei konventionellen Kraftwerken der Fall ist. Die Herausforderung besteht nun darin, das Stromnetz stabil zu halten und gleichzeitig den Anteil volatiler Energieerzeuger zu erhöhen. Einen Eindruck über Komplexität und Umfang dieser Aufgabe geben die Zahlen des Netzbetreibers Tennet. Es mussten nach eigener Aussage für das Jahr 2017 knapp eine Milliarde Euro für Noteingriffe ins Stromnetz ausgegeben werden [WEL18].

Auch der für den Betrieb von Rechenzentren verwendete Energieanteil gewinnt stetig an Relevanz und verdient damit eine gesonderte Betrachtung im Kontext der Energiewende und der Integration in ein intelligentes Netzwerk aus Strom-Erzeugern und Abnehmern.

Die Anzahl von IT-Systemen wächst stetig an und IT-Services werden immer stärker Bestandteil des täglichen Lebens. Die Digitalisierung dringt dabei in weitere Bereiche vor und weckt so den Bedarf an diversen neuen Services. Neue IT-Services werden etabliert, die letztlich einen insgesamt steigenden Bedarf an Rechenleistung nach sich ziehen. An den Fraunhofer-Instituten für Zuverlässigkeit und Mikrointegration sowie für System- und Innovationsforschung [SNP⁺09, SPZ⁺15] werden diese Entwicklungen genauer betrachtet. Allein im Bereich der Nutzung von Video- und TV-Anwendungen wird eine jährliche Zunahme des IP-Datenverkehrs um 46% prognostiziert. Die technologische Weiterentwicklung kompensiert den steigenden Bedarf an Rechenleistung jedoch nicht und es müssen mehr Server verbaut werden, folglich nimmt auch der Energiebedarf von IT-Services und Rechenzentren in Summe zu.

In dieser Arbeit werden die grundlegenden Zusammenhänge bezüglich der Leistungsaufnahme von Servern in Rechenzentren erläutert und ein neues Konzept zur Realisierung einer adaptiven Leistungsaufnahme von Servern vorgestellt. Die Evaluation des Verfahrens erfolgt anhand von realen Datensätzen, die freundlicherweise von einem Industrieunternehmen zur Verfügung gestellt wurden.

1.1. Motivation

Den Betrieb einer Applikation in einem professionellen Rechenzentrum ausführen zu lassen, hat diverse Gründe. Hauptsächlich werden besondere Anforderungen an die Verfügbarkeit, Leistung, aber auch an die Betriebssicherheit gestellt. In dem Zuge werden Systeme oftmals in einem Maße dimensioniert, die die schieren Anforderungen der Applikationen übersteigen.

Das Unternehmensziel eines IT-Serviceanbieters ist die ökonomische und gegebenenfalls ökologische Bereitstellung seiner Dienste. Daher gilt es, die aufgewendete Energie möglichst effizient zu nutzen, günstig einzukaufen und sich einer verändernden Marktsituation zu stellen. Dabei sehen sich die Serviceanbieter mit Fragestellungen bezüglich des ökologischen Fußabdrucks (CO₂-Footprint) konfrontiert [Her11], dessen Bilanz mit dem Einsatz erneuerbarer Energien und effektiver Energienutzung positiv beeinflusst werden kann. Bei der Umsetzung entsprechender Maßnahmen ist weiterhin die vertraglich vereinbarte Servicequalität zu gewährleisten.

Der Herausforderung einer effizienten Energienutzung kann beispielsweise mit dem Lastmanagement eines Rechenzentrums begegnet werden. Mit Hilfe von Virtualisierungstechnologien lassen sich Server virtualisieren und es können so mehrere virtuelle Server auf einem physikalisch vorhandenen Server betrieben werden. Das angesprochene Lastmanagement kann nun über die Verteilung der virtuellen Server ein Ressourcenmanagement vornehmen und konsolidieren. Die Anzahl physikalischer Server, die andernfalls nicht optimal ausgelastet wären, kann so reduziert werden.

Über den Einsatz virtueller Server gewinnt ein Rechenzentrum auch an Flexibilität bezüglich der Kapazitätsplanung. Durch die Virtualisierungstechnologie ist es verhältnismäßig einfach möglich, Serverressourcen zu erweitern oder zu verringern. Die eingangs beschriebene Überdimensionierung aus den Befürchtungen von Ressourcenengpässen heraus kann so reduziert werden.

Aktuell können Virtualisierungslösungen über geeignete Messverfahren die Auslastung einzelner Server feststellen, es kann eine Konsolidierung veranlasst und gegebenenfalls physikalische Server zur Energieeinsparung abgeschaltet werden. Weiterhin gibt es Prognoseverfahren, wie in [Hoy11] beschrieben, über die eine vorzuhaltende Serverkapazität geplant wird, damit die Servicequalität gehalten und Engpässe vermieden werden.

Ein Problem hierbei ist, dass diese Verfahren die auf den Servern vorherrschende CPU-Auslastung als Vorgabe hinnehmen und auf dieser Basis entsprechende Kapazitäten und Optimierungsschritte planen. In dieser Konstellation werden alle Ressourcenanforderungen gleich behandelt, denn weitere Informationen über die fachliche Dringlichkeit der durchzuführenden Operationen sind nicht verfügbar. Es wird die auf den Servern evozierte CPU-Auslastung als fixe, unbeeinflussbare Größe verstanden, anhand derer ein optimaler Zustand des Rechenzentrums gesucht wird.

Andere Verfahren, wie in [BF07] und [CCC14] vorgeschlagen, setzen genau an diesem Punkt an und arbeiten mit Scheduling-Verfahren. Es werden im Vorfeld die Fälligkeiten für anstehende Rechenoperationen bestimmt und die Ausführung entsprechend der

Dringlichkeiten gesteuert. Passend zu den Fälligkeiten der jeweiligen Operationen wird eine priorisierte Liste erstellt, die mit einem Scheduling-Verfahren abgearbeitet wird. Für eine derartige Vorgehensweise sind allerdings fachliche Kenntnisse bezüglich der Rechenoperationen essentiell, die nicht immer zur Verfügung stehen. Bei diesen Verfahren werden die Aspekte der Energieverfügbarkeit derzeit ebenfalls nicht hinreichend berücksichtigt.

Im Zusammenhang mit effizienter Energienutzung in Rechenzentren ergeben sich mit dem zunehmenden Einsatz erneuerbarer Energien weitere Aspekte, die es ebenfalls zu betrachten gilt. Es ist zukünftig mit steigenden Energiepreisen und aufgrund der variablen Energieverfügbarkeit mit einem agilen Preismodell seitens der Stromanbieter zu rechnen. Um variable Strompreise nutzen zu können, muss die Leistungsaufnahme des Rechenzentrums anpassbar sein. Über sogenannte Service Level Agreements (SLAs) wird eine bestimmte Service-Leistung des Rechenzentrums in einem definierten Zeitraum festgelegt. Es werden beispielsweise Zeitfenster definiert, in denen eine Applikation einen bestimmten Durchsatz produzieren muss. Hier besteht nun die Möglichkeit bei geringem Strompreis in Vorproduktion zu gehen und diesen Puffer in Perioden mit deutlich höherem Strompreis wieder abzubauen, ohne Verletzung der SLAs. Der Strompreis orientiert sich dabei an der Verfügbarkeit und kann stark schwanken. Eine Speicherung des Stroms kann mit Hilfe weiter Technologien erfolgen, ist aber mit hohen Verlusten behaftet.

Der elektrische Strom ist eine Energieform, die nicht im Netz selbst gespeichert oder gepuffert werden kann, daher muss die Stromeinspeisung stets im selben Maße des Strombedarfs erfolgen. Um dieses Gleichgewicht herzustellen, werden unterschiedliche Kraftwerkstypen eingesetzt, die in Kombination sowohl die Grundlast als auch Spitzenlasten bedienen können. Mit dem fortschreitenden Ausbau der erneuerbaren Energien gewinnt der Anteil dieser Energiequellen zunehmende Relevanz bezüglich der Netzstabilität. Die Abgabeleistung ist beispielsweise bei Solaranlagen von der Sonneneinstrahlung abhängig und kann daher nicht erhöht, nur gesenkt werden. Aktuell werden große Solaranlagen über Rundsteueranlagen bei einem Energieüberangebot in ihrer Abgabeleistung an das Stromnetz reduziert oder von vornherein auf 70% ihrer Maximalleistung beschränkt [Bun12]

In Abbildung 1.1 ist die Energie-Säule auf dem EWE-Campus an der Cloppenburg Straße in Oldenburg zu sehen. Die Säule wird mit aktuellen Daten direkt aus der Netzleitstelle versorgt und bringt diese zur Anzeige. In Abbildung 1.1a ist die Netzbelastung durch den Leuchtbalken dargestellt, in Abbildung 1.1b ist der momentane Anteil an Windstrom abzulesen.

Gerade die erneuerbaren Energien wie Solar- und Windenergie können keinen kontinuierlichen Energiefluss sicherstellen und sind im Kontext der Produktionssteuerung allenfalls in der Abgabeleistung reduzierbar. Eine steuerbare Erhöhung der Abgabeleistung ist in der Regel nicht möglich. Ziel ist es nun, in Rahmen dieser Arbeit auch Belange von Netzbetreibern und Stromlieferanten von erneuerbaren, intermittierenden Energiequellen in das Lastmanagement für Rechenzentren zu integrieren. So soll Energie bei hoher Verfügbarkeit zu günstigen Konditionen genutzt werden können. In diesem Umfeld könnte ein Rechenzentrum als regelbarer Abnehmer und ebenso als Instrument zur Netzstabilisierung dienen. Indem die Energieaufnahme in einem gewissen Maße regulierbar ist, könnte ein Rechenzentrum als wertvoller Teil eines sogenannten Smart-Grid fungieren.

1. Einleitung und Motivation



Abbildung 1.1.: Energie-Informationssäule am Campus Cloppenburg Straße der EWE AG

Smart-Grids vernetzen Stromproduzenten stärker mit Stromkonsumenten und bieten einen Ansatz zur Optimierung des Zusammenspiels zwischen Erzeugung, Konsum, Speicherung und Netzmanagement. Somit erlauben Smart-Grids eine Abstimmung der Teilnehmer untereinander und ermöglichen so eine gleichmäßigere Auslastung der Netze und effizienteren Betrieb von Kraftwerken. Auch Rechenzentren ließen sich mit entsprechenden Verfahren in ein Smart-Grid integrieren.

Aktuelle Verfahren zur Steuerung der Leistungsaufnahme von Rechenzentren, wie in [AVZR11, BR11, SGIS10] und [Hoy11] beschrieben, arbeiten mit vordefinierten, festen Lastkurven. Diese Lastkurven werden auf Basis von Prognoseverfahren erstellt und im Nachgang abgefahren. In dem tatsächlichen Szenario ergibt sich jedoch oftmals eine Abweichung, die einerseits aus der Genauigkeit des Prognoseverfahrens und andererseits aus der Umsetzungsqualität durch den Lastverteilungsalgorithmus resultiert. Neben den in der Realität abweichenden Lastparametern kann es weitere Umstände geben, die ad hoc eine Anpassung der Leistungsaufnahme eines Rechenzentrums erforderlich machen können. Beispiele hierfür können Ausfälle in der Infrastruktur des Rechenzentrums selbst sein oder auch eine unerwartete Änderung der Energieverfügbarkeit.

In diesem Kontext wird eine temporäre Leistungsanpassung des Rechenzentrums ohne applikationsinvasive Maßnahmen benötigt, wobei die Betriebssicherheit der Applikationen nicht beeinträchtigt werden darf. Verfahren, die einen fachlichen oder technischen Eingriff an Applikationen vornehmen, werden in dieser Arbeit als applikationsinvasiv bezeichnet.

Insbesondere die Betriebssicherheit und Hochverfügbarkeit von Applikationen sind die ausschlaggebenden Faktoren, sie in Rechenzentren zu betreiben. Diese Faktoren dürfen durch etwaige Maßnahmen nicht in Frage gestellt werden. Zusammenfassend können hier zwei große Handlungsfelder beschrieben werden: zum einen die effiziente Nutzung der aufgewendeten Energie generell und zum anderen die Anpassung der Leistungsaufnahme

des Rechenzentrums, um auf die Stromverfügbarkeit bzw. den Arbeitspreis in einem agilen Preisgefüge zu reagieren.

1.2. Nutzungsszenario

Ziel dieser Arbeit ist es, eine Methodik bereitzustellen, mit der die Leistungsaufnahme von Rechenzentren gesteuert werden kann. Insbesondere soll ein reaktives Verhalten auf die Energieverfügbarkeit ermöglicht werden, die durch erneuerbare Energien stark schwanken kann. In einigen Ansätzen [CHT12, LLW⁺11] wird beschrieben, wie in großen Verbunden von Rechenzentren die Last in jene verschoben wird, deren Standorte eine hohe Energieverfügbarkeit oder günstige Strompreise aufweisen. Derartige Ansätze werden hier nicht betrachtet, die angestrebte Lösung soll explizit mit den lokalen Bedingungen arbeiten und auch im regionalen Rahmen Anwendung finden können. Die Leistungsaufnahme soll konkret beeinflusst werden, ohne die Rechenlast an andere Standorte zu verschieben.

In einem konkreten Nutzungsszenario verwaltet der Rechenzentrumsbetreiber die geschlossenen Service Level Agreements (SLAs) gegenüber Kunden und Lieferanten in einer Service Level Management (SLM)-Applikation, die fortwährend die Einhaltung der SLAs und Service Level Objectives (SLOs) überwacht. Dies wird in Kapitel 2 ausführlich beschrieben. Nun soll im Sinne des Smart-Grid-Gedankens auf das aktuell verfügbare Stromangebot reagiert werden können, wobei sich die Stromverfügbarkeiten im Strompreis niederschlagen. Diese Lieferantenbeziehung lässt sich ebenfalls in der SLM-Applikation abbilden. Der Anreiz besteht aus ökonomischer Sicht nun darin, bei einem günstigen Strompreis die Produktion des Rechenzentrums zu erhöhen und im umgekehrten Falle die Leistungsaufnahme abzusenken.

Wird eine geringere Energieverfügbarkeit über einen steigenden Strompreis signalisiert, kann die SLM-Applikation den möglichen Spielraum zur Leistungsreduzierung ermitteln und einen Zielwert vorgeben. Analog lässt sich auch die Steigerung der Leistungsaufnahme betrachten, wobei hier die Gefährdung von SLAs eher an Bedeutung verliert. Die Maßnahmen zur Erhöhung der Leistungsaufnahme bedingen eine höhere Applikationsperformance und Verletzungen der geschlossenen SLAs sind in dem Zuge eher unwahrscheinlich.

Es gibt verschiedene Geschäftsmodelle zum Betrieb bzw. Einkauf von Rechenzentrumsdienstleistungen, die sich in der Serviceebene unterscheiden. Bei dem Modell IaaS (Infrastructure as a Service) wird dem Kunden die Infrastruktur des Rechenzentrums zur Verfügung stellt, wobei der Betrieb von Applikationen hier nicht zur Dienstleistung gehört. Das Modell SaaS (Software as a Service) schließt genau diese Dienstleistung mit ein, dem Kunden wird der Applikationsbetrieb inklusive Hardwarebetrieb als Komplettpaket angeboten. Der entscheidende Unterschied zwischen diesen Betriebsmodellen liegt im Zugriff auf und der Verantwortung für die betriebenen Applikationen. Bei einem SaaS-Modell besteht zwar Zugriff auf die Applikationen durch den Dienstleister, Methoden zur Steuerung der Applikationsperformance müssten jedoch implementiert werden. Darüber hinaus setzt dies fundierte Kenntnisse über die fachlichen Zusammenhänge der Applikationen voraus. Bei einem IaaS-Modell besteht kein Zugriff auf die Applikationen. Das Vorhaben

1. Einleitung und Motivation

einer Steuerung der Leistungsaufnahme kann folglich nicht mittels direkter Beeinflussung der Applikationen durch den Dienstleister erreicht werden, applikationsinvasive Methoden scheiden somit aus.

In dem verbreiteten Betriebsmodell des SaaS und IaaS soll es nun ermöglicht werden, Applikationen, die keine Realzeitanforderungen haben und deren Performance in einem Zeitintervall von min. 60 Minuten angepasst werden kann, für eine Leistungssteuerung zu verwenden. Hierzu wird ein Verfahren benötigt, das eine geringe, leicht überschaubare Komplexität und Transparenz aufweist, damit das Vertrauen in die Technologie geschaffen werden kann. Über die Leistungsanpassung soll in der Folge die Leistungsaufnahme des Rechenzentrums reduziert bzw. erhöht werden können, um beispielsweise einen Beitrag zur Netzstabilisierung leisten zu können. Die Anforderungen hierzu lehnen sich an die der Prozesse zur Regelleistung an.

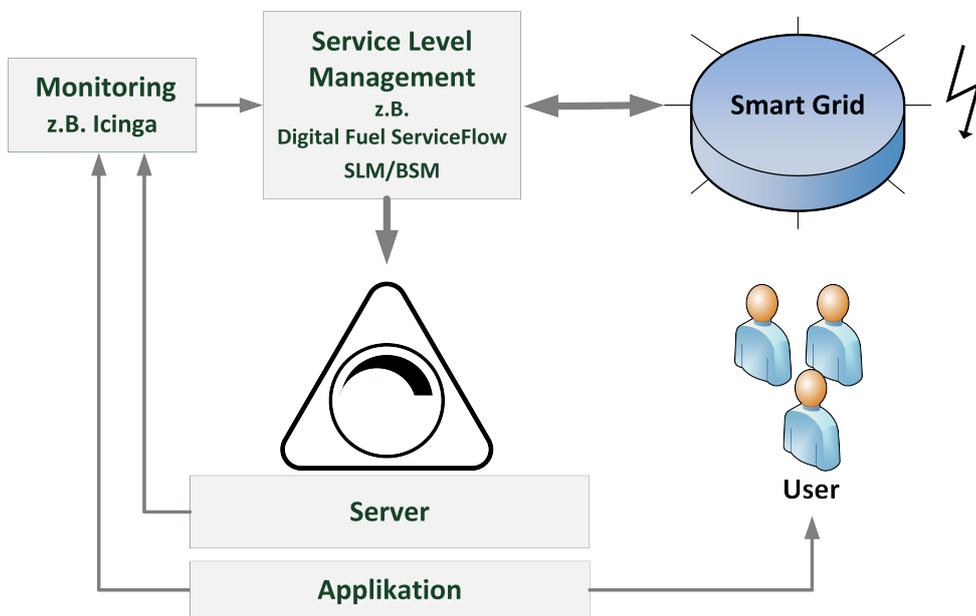


Abbildung 1.2.: Nutzungsszenario für regelbare Leistungsaufnahme von Servern in einem Rechenzentrum

Abbildung 1.2 stellt das avisierte Nutzungsszenario dar. Die Anforderungen von Kunden und Nutzern der IT-Services ebenso wie die Energieverfügbarkeiten werden an die SLM-Applikation übergeben. Über Monitoring-Systeme werden Informationen über Applikationen und Server zugelifert. Auf Basis dieser Daten kann ein neuer Zielwert für die Leistungsaufnahme der Server erstellt und an das zu implementierende Verfahren übergeben werden.

Im Rahmen dieser Arbeit soll ein *Leistungsregler für Server* entwickelt werden, der eine zugeliferte Ziel-Leistungsaufnahme der Server herstellt und so die energetischen Freiheitsgrade von Rechenzentren erweitern kann. Die Beeinflussung der Leistungsaufnahme soll auf Grundlage einer Ressourcen-Überprovisionierung erfolgen, wie später genauer erläutert wird.

1.3. Anwendungsfälle und Forschungsfragen

Die in dieser Arbeit untersuchten Aspekte sollen eine Basis zur Implementierung und Erweiterung von Lastmanagement-Systemen liefern. Neben den inhärenten Prozessen und Anforderungen an ein Rechenzentrum sollen die Belange externer Stakeholder im Lastmanagement abbildbar sein. Insbesondere werden die durch den verstärkten Einsatz erneuerbarer Energieträger mit intermittierender Leistungsabgabe entstehenden variablen Energieverfügbarkeiten berücksichtigt. Mögliche Anwendungsfälle sind:

- *Bedienung externer Stakeholder:* Abgesehen von inhärenten Prozessen eines Rechenzentrums können über SLAs externe Faktoren eine Rolle spielen. Ein Rechenzentrum könnte beispielsweise auf die Anforderungen eines Netzbetreibers reagieren und entsprechende Strommengen nach dessen Vorgabe abnehmen. So kann das Stromnetz stabil gehalten werden, ohne auf das Zuschalten von schnell verfügbaren Energiequellen wie Gaskraftwerken zurückgreifen zu müssen. Ebenso kann bei einem Überangebot an Energie die elektrische Last des Rechenzentrums erhöht werden. Auch dieser Fall gewinnt mit dem steigenden Anteil an erneuerbaren Energien – wie Wind- und Sonnenenergie – stetig an Relevanz.
- *Ökonomischer Rechenzentrumsbetrieb:* In der Kombination aus effizienter Energienutzung und der Möglichkeit einer variablen Lastaufnahme lässt sich Strom zu günstigen Konditionen einkaufen und optimal verwerten. Durch eine gleichmäßige und planbare Server-Auslastung können alle Systeme in den Bereichen ihres optimalen Wirkungsgrades gefahren werden. Die Effizienz des Rechenzentrums kann so gezielt beeinflusst werden.
- *Ökologischer Rechenzentrumsbetrieb:* Die Leistungsaufnahme kann in Abhängigkeit zum aktuellen Strommix gesteuert werden, um den CO_2 -Fußabdruck zu reduzieren.
- *Unterstützung interner Prozesse des Rechenzentrums:* Durch gezielte Umverteilung lassen sich die Server auf einem definierten Auslastungsniveau halten. Vorteilhaft für den Betreiber von Rechenzentren sind hier die aktive Vermeidung von Hotspots und die kalkulierbare Auslastung und Lebensdauer der eingesetzten Server inklusive Backend-Technik. Auch unvorhersehbare Umstände können die Anpassung des Auslastungsniveaus erfordern. Fällt beispielsweise ein Kühlsystem aus, kann die entstehende Abwärme der Server nicht abgeführt werden. Mit Hilfe eines adaptierenden Verfahrens könnte die Serverleistung reduziert werden, ohne IT-Services selbst unterbrechen zu müssen. Denkbar sind diverse weitere Szenarien, bedingt durch Ausfall, Störung oder Wartungsarbeiten.

Hauptaspekt des zu entwickelnden Verfahrens ist die Erweiterung der energetischen Freiheitsgrade eines Rechenzentrums bezüglich der elektrischen Leistungsaufnahme. Es gibt eine Reihe nachgelagerter Vorteile und Einsatzmöglichkeiten, die sich durch eine Erweiterung ergeben, im Rahmen dieser Arbeit jedoch nicht weiter betrachtet werden:

- Integration und dadurch verstärkte Nutzung von erneuerbaren Energien
- Optimierte Nutzung von Energieträgern (z. B. kein Abschalten von Solaranlagen über Rundsteueranlage)

1. Einleitung und Motivation

- Green-IT Label: Nachhaltigkeit auch als Werbeträger, z.B. Blauer Engel[RAL15]
- Erhöhung der Energie-Versorgungssicherheit durch Nutzung von Rechenzentren als regelbares Element zur Netzstabilisierung
- Erweiterung der Freiheitsgrade, um weitere Anforderungen zu bedienen: Beispielsweise die Nutzung der von den Systemen produzierten Abwärme zum Heizen von Gebäuden.

Aus dem beschriebenen Nutzungsszenario leiten sich folgende Forschungsfragen ab:

1. Mit welcher Methode lässt sich die Leistungsaufnahme von Servern im Betriebskontext von SaaS und IaaS sowohl senken als auch erhöhen und gleichzeitig reaktives Verhalten auf eine akute Anforderung mit nur einem Rechenzentrumsstandort realisieren?
2. Welcher Effekt lässt sich mit Ressourcen-Überprovisionierung auf die Leistungsaufnahme von Servern erzielen?
3. Kann ein skalierendes Verfahren modelliert werden, das mit der Anzahl von betriebenen VMs und PMs erweiterbar ist?
4. Wie kann die Komplexität der Aufgabenstellung reduziert werden, um ein simples Verfahren zu erstellen, und die Verfahrensweise transparent dargestellt werden?
5. An welchen Stellen kann bei einem Fokus auf Leistungssteuerung dennoch die Effizienz bezüglich der eingesetzten Energie Berücksichtigung finden?

Zum Zeitpunkt der Erstellung dieser Arbeit ist kein Verfahren bekannt, das die beschriebenen Themenfelder in Gänze abdeckt. In Abschnitt 3 werden die bekannten Verfahren zur Steuerung der Leistungsaufnahme beschrieben und die Lücke in der Reihe dieser Verfahren dargestellt.

1.4. Gliederung der Arbeit

Zur Bearbeitung des in diesem Abschnitt beschriebenen Themenfeldes werden in Kapitel 2 die grundlegenden Technologien und Zusammenhänge beschrieben. In Kapitel 3 werden Verfahren beschrieben, die den aktuellen Stand der Technik repräsentieren und im Kontext der Einflussnahme auf Effizienz und Leistungsaufnahme von Servern in Rechenzentren stehen. Die genaue Beschreibung der Problemstellung erfolgt in Kapitel 4. Es werden die Handlungsfelder aufgezeigt und eine thematische Abgrenzung vorgenommen. Das Problem wird formal charakterisiert und eine Analogie zu bestehenden Problemen aufgebaut sowie eine Abschätzung der Komplexität vorgenommen. Auf Basis der gewonnenen Erkenntnisse wird in Kapitel 5 die Modellierung des Verfahrens vorgestellt. Dazu wird das Verfahren zunächst informell beschrieben und die notwendigen Voraussetzungen und Annahmen werden definiert. Im Anschluss wird die Simulation des Verfahrens im Framework Uppaal vorgestellt und die Details der Implementierung dargestellt. Eine Bewertung des Verfahrens und die Diskussion der erzielten Ergebnisse erfolgen in Kapitel 6.

Kapitel 7 fasst die Ergebnisse der Arbeit zusammen und gibt einen Ausblick auf mögliche Erweiterungen des Verfahrens.

1.5. Zusammenfassung

Einleitend wurde das aktuelle Energiekonzept der Bundesregierung beschrieben, das einen weitreichenden Ausbau der erneuerbaren Energien vorsieht. Dem wurden die Entwicklungen im Bereich der IT-Services gegenübergestellt und das Potential einer Verknüpfung beider Stränge aufgezeigt. Es wurde ein konkretes Nutzungsszenario für einen neuen Ansatz skizziert und die möglichen Anwendungsfälle mit ihren resultierenden Vorteilen beschrieben. Aus den Teilen zur Bearbeitung des Themenfeldes wurde die Gliederung der Arbeit abgeleitet und vorgestellt.

2. Grundlagen

In diesem Kapitel werden die technischen Gegebenheiten und Zusammenhänge im Umfeld der elektrischen Leistungsaufnahme von Rechenzentren beleuchtet. Es wird ein Überblick über die Komponenten eines Rechenzentrums und die Architektur gegeben. Um die Leistungsaufnahme eines Servers, dessen Komponenten und deren Einfluss auf die Leistungsaufnahme bewerten zu können, werden deren Arbeitsweisen detailliert beschrieben.

In Abschnitt 2.3 werden die Ressourcen eines Server sowie deren charakteristische Eigenschaften beschrieben. Diese Informationen sind in Kapitel 4 relevant, wo die Eignung der Ressourcen bezüglich einer Überprovisionierung und Beeinflussung der Leistungsaufnahme betrachtet wird.

Eine häufig verwendete Basistechnologie ist die Server-Virtualisierung, die auch im Rahmen dieser Arbeit zum Einsatz kommen soll und deren Modalitäten kurz vorgestellt werden. Weiterhin wird das Service Level Management beschrieben, das einen Rahmen um die erbrachten Leistungen eines Rechenzentrums bildet.

Abschnitt 2.8 beschreibt die etablierten Prozesse der Regelleistung, um das Stromnetz stabil zu halten. Für eine aktive Teilnahme oder Unterstützung dieser Prozesse ergeben sich bereits konkrete Anforderungen, die ein entsprechendes Verfahren abdecken muss.

2.1. Das Rechenzentrum

Der Begriff *Rechenzentrum* ist kein allgemein definierter, feststehender Begriff, es soll daher zunächst kurz beschrieben werden, wie er im Rahmen dieser Arbeit verwendet wird.

Als Rechenzentrum wird in dieser Arbeit ein Gebäudekomplex bezeichnet, in dem die Rechen- und Support-Technik untergebracht ist. Ein Rechenzentrum kann aus mehreren sogenannten IT-Räumen bestehen, in denen die Server untergebracht sind. Es erfolgt in der Regel eine Trennung der Server- und Support-Infrastruktur. Im Kontext dieser Arbeit ist die Leistungsaufnahme eines Rechenzentrums von besonderer Bedeutung. Sie setzt sich aus den Leistungsaufnahmen mehrerer Komponenten zusammen, zudem gibt es systematische Zusammenhänge bezüglich der Leistungsaufnahme, die in diesem Abschnitt erläutert werden.

Abbildung 2.1 zeigt eine aktuell übliche Verteilung des Energiebedarfs im Rechenzentrum [DRTB14]. Es ist erkennbar, dass etwa die Hälfte der eingesetzten Leistung von den Servern aufgenommen wird.

2. Grundlagen

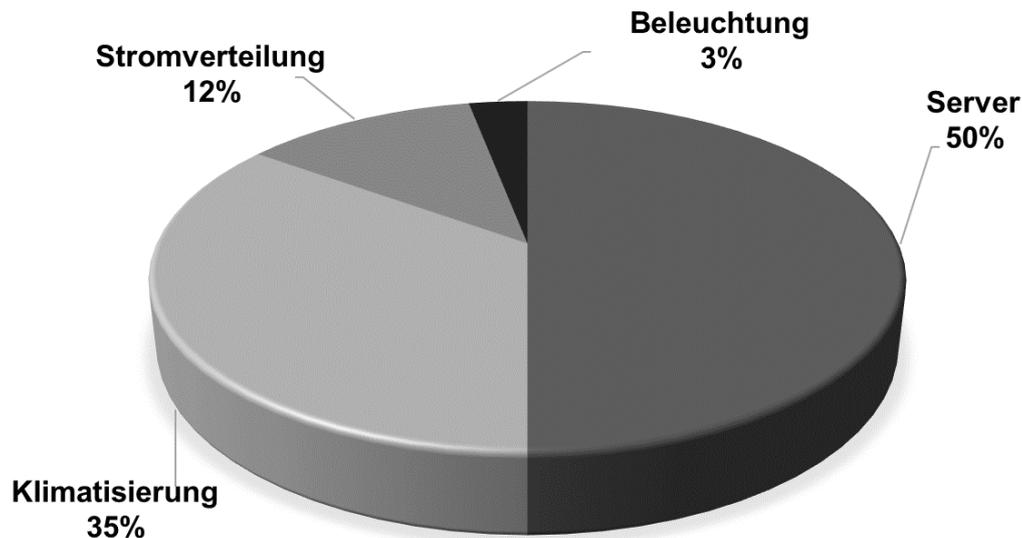


Abbildung 2.1.: Anteilige Darstellung des Energiebedarfs in Rechenzentren; eigene Darstellung, Daten entnommen aus [DRTB14]

Neben der Leistungsaufnahme der Server entfällt ein großer Anteil auf die Klimatisierung des Rechenzentrums.

2.1.1. Leistungsaufnahme der Infrastruktur

Bei der Betrachtung der Leistungsaufnahme eines Rechenzentrums sind zunächst die Server selbst und deren Leistungsaufnahme zu nennen. Der elektrische Strom wird an die Server über Power Distribution Units (PDUs) verteilt, ähnlich einer Mehrfachsteckdose. Managebare PDUs benötigen selbst Energie wie auch die Netzwerkinfrastruktur mit Switchen, Routern usw.

Um eine unterbrechungsfreie Stromversorgung der Server zu gewährleisten, werden Netzersatzanlagen (NEA) und unterbrechungsfreie Stromversorgungen (USV) benötigt, die ihrerseits ebenfalls Strom benötigen.

Unterbrechungsfreie Stromversorgung

Der Bereich der USV-Technik wird in drei Stufen eingeteilt, die sich wiederum in unterschiedliche Klassen gliedern. Eine häufig verwendete Art der USV-Technik ist die sogenannte Online-USV, auch als Doppelwandler-, Double-Conversion- oder Dauerbetrieb-USV bezeichnet. Das schematische, stark vereinfachte Schaltbild einer solchen Online-USV ist in Abbildung 2.2 dargestellt.

Die Netzspannung wird zunächst mit einem Gleichrichter in Gleichspannung umgewandelt, wie sie auch von Akkumulatoren abgegeben wird. Es werden nun die aus dem

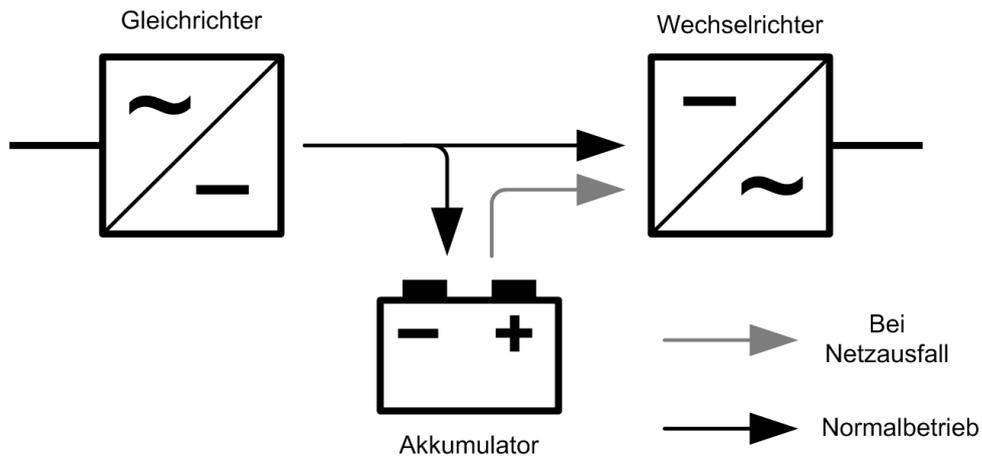


Abbildung 2.2.: Schematisches Blockschaltbild einer Online-USV

Stromnetz bezogene Gleichspannung und die Akkumulatoren parallel geschaltet. Über die Gleichspannung aus dem Stromnetz wird gleichzeitig eine Erhaltungsladung der Akkumulatoren sichergestellt[Kre05].

Das Ergebnis ist eine netzautarke Gleichspannungsversorgung. Die so erhaltene Gleichspannung muss nun für die weitere Verwendung wieder in eine Wechselspannung mit 50 Hz umgewandelt werden. Diese Aufgabe übernimmt ein Wechselrichter, wie dargestellt. Der Wechselrichter kann eine konstante Frequenz und Spannung sicherstellen.

Durch diesen Aufbau der Online-USV werden Abweichungen in der Netzfrequenz und Spannung direkt kompensiert. Nachteilig bei dieser Art der USV ist der permanente Betrieb der USV und die damit verbundene Leistungsaufnahme der Komponenten. Andere USV-Typen, die erst bei einem Netzausfall in Betrieb gehen, haben diese zusätzliche Leistungsaufnahme nicht. Allerdings können Spannungsschwankungen und Netzfrequenzschwankungen nicht abgefangen werden und es kann beim Umschaltvorgang zu einer Spannungsunterbrechung von mehreren Millisekunden kommen. Es ist noch zu erwähnen, dass bei der Verwendung einer Online-USV die Auslastung mit der abgerufenen Leistung steigt und in gleicher Weise die Verlustleistung.

Werden für einen Versorgungsausfall Netzersatzanlagen (NEAs) vorgehalten, ergeben sich weitere Aspekte bezüglich der Leistungsaufnahme. Die Dieselgeneratoren von NEAs müssen beispielsweise mittels elektrischer Heizung auf einer Mindesttemperatur gehalten werden, um im Bedarfsfall sicher und schnell gestartet werden zu können. Auch die Leistungsaufnahme, die zur Beheizung der Generatoren benötigt wird, addiert sich zur Leistungsaufnahme des Rechenzentrums. Diese Leistungsaufnahme ist unabhängig von der Auslastung des Rechenzentrums und in diesem Kontext auch nicht beeinflussbar.

2. Grundlagen

2.1.2. Klimatisierung

Zur Kühlung der Server und weiterer Komponenten wird Klimatisierungstechnik eingesetzt, deren Leistungsaufnahme ebenfalls in die Gesamtbetrachtung einfließt.

Die in einem Rechenzentrum betriebenen Server und Komponenten produzieren Abwärme, die bedarfsgerecht abgeführt werden muss, um eine Überhitzung der Komponenten zu vermeiden. Hierzu kommen verschiedene Technologien zum Einsatz. Bei der Freikühlung wird die Außenluft zur Kühlung des Rechenzentrums verwendet. Unterschieden wird hierbei zwischen direkter und indirekter Freikühlung:

- *Direkte Freikühlung*: Bei der direkten Freikühlung wird die Außenluft gefiltert und ersetzt die im Rechenzentrum zuvor erwärmte Luft. Bei diesem Verfahren wird auch die Luftfeuchtigkeit im Rechenzentrum von der Außenluft bestimmt. Zu feuchte oder zu trockene Luft kann die Wärmeabfuhr und die Lebensdauer der Server negativ beeinflussen.
- *Indirekte Freikühlung*: Die beschriebenen negativen Effekte direkter Freikühlung können mittels indirekter Freikühlung vermieden werden, bei der Wärmetauscher zum Einsatz kommen. Die kühlere Außenluft und die warme Luft des Rechenzentrums werden über Wärmetauscher geführt, sodass lediglich eine Wärmeabgabe an die Außenluft erfolgt ohne einen Luftaustausch.

Die Freikühlung ist eine sehr effiziente Methode, um die anfallende Abwärme abzuführen, und wird daher intensiv untersucht [ZWZ17, NLM18].

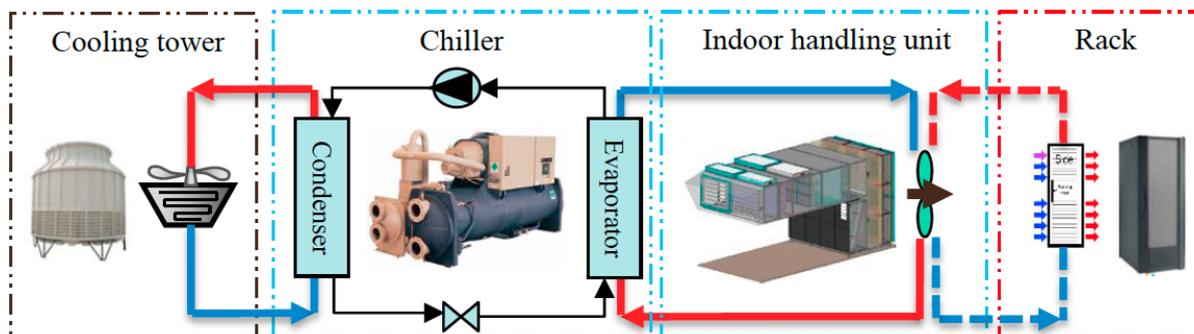


Abbildung 2.3.: Gesamtübersicht eines Klimatisierungsstrangs mit Einsatz von Kompressoren; Grafik unverändert übernommen aus [ZWZ17]

In Abbildung 2.3 sind die einzelnen Komponenten eines klassischen Klimatisierungssystems mit Kompressor dargestellt. Die Abwärme der Server wird nach dem Prinzip der Verdichtung von Kältemittel durch einen Kompressor abgeführt.

Steigt die Auslastung der Server, erhöhen sich deren Leistungsaufnahme und die abgegebene Wärme. Mit einem gewissen zeitlichen Verzug erhöht sich auch die Leistungsaufnahme des Klimatisierungssystems. Der Verzug kommt durch Regelung der Klimatisierung zustande und beschreibt die Dauer von der Erhöhung der Leistungsaufnahme der Server,

bis die Sensorik der Klimatisierung die erhöhte Temperatur erfasst hat und ebenfalls die Kühlleistung erhöht.

Das Klimatisierungssystem ist typischerweise so konfiguriert, dass eine bestimmte Lufttemperatur gehalten werden soll. Die Server des Rechenzentrums geben je nach deren Leistungsaufnahme mehr oder weniger erwärmte Luft an die IT-Räume ab. Die Leistungsaufnahme der Klimatisierung ist somit mittelbar an die Leistungsaufnahme der Server gekoppelt.

Hotspots

Zum Abtransport von Wärme muss es eine Temperaturdifferenz zwischen der eingeblasenen Luft und der zu kühlenden Komponenten geben. Je größer die Temperaturdifferenz ist, umso mehr Wärme kann transportiert werden. Idealerweise haben die Server und IT-Komponenten in einem IT-Raum einen ähnlichen Kühlbedarf, sodass mit einer angemessenen Lufttemperatur ein ausreichender Abtransport der entstehenden Wärme erfolgen kann.

Eine immer wiederkehrende Herausforderung bei der Klimatisierung von IT-Räumen ist das Auftreten von sogenannten Hotspots. Unter einem Hotspot wird beispielsweise ein Server verstanden, der sich überproportional erwärmt hat. Dieser Server benötigt für eine ausreichende Kühlung deutlich kühlere Zuluft als die umliegenden Server. In der Regel muss in so einem Fall die Temperatur der Zuluft eines Racks oder sogar das ganze IT-Raums abgesenkt werden, um den Hotspot ausreichend zu kühlen. Die Effizienz der Klimatisierung ist durch eine derartige Maßnahme stark rückläufig.

2.2. Leistungsaufnahme eines Servers

Die Leistungsaufnahme eines Servers setzt sich aus den Leistungsaufnahmen der einzelnen Komponenten des Servers zusammen. Ausgehend vom Netzteil wird der Strom auf die Komponenten verteilt. Ein Kabelstrang ist direkt am Mainboard, der Hauptplatine, angeschlossen. Weitere Stränge versorgen Laufwerke, Lüfter und andere Komponenten. Alle diese Komponenten haben ein spezifisches Verhalten bezüglich ihrer Leistungsaufnahme, das sich je nach Lastsituation ändern kann. Die Leistungsaufnahme eines Servers setzt sich daher aus einem dynamischen und einem statischen Teil zusammen. Der statische Teil ($P_{server\ Idle}$) beschreibt eine Grund-Leistungsaufnahme, die bei eingeschaltetem Server nicht unterschritten werden kann. Der dynamische Teil ($P_{server\ dyn.}$) addiert sich zu der Grund-Leistungsaufnahme und ist abhängig von der Auslastung eines Servers bzw. dessen Komponenten und Betriebsmitteln. Die Leistungsaufnahme eines Servers (P_{Server}) setzt sich daher folgendermaßen zusammen:

$$P_{Server} = P_{server\ Idle} + P_{server\ dyn.} \quad (2.1)$$

2. Grundlagen

In Abbildung 2.4 ist beispielhaft die Leistungsaufnahme eines Servers in Abhängigkeit zur Auslastung dargestellt. Ohne Auslastung (Active Idle) hat der Server eine Leistungsaufnahme von 86 Watt, bei voller Auslastung sind es 562 Watt. Somit ergibt sich für diesen Server ein Leistungsaufnahmebereich von:

$$P_{Server} = 86 + [0, 476]. \quad (2.2)$$

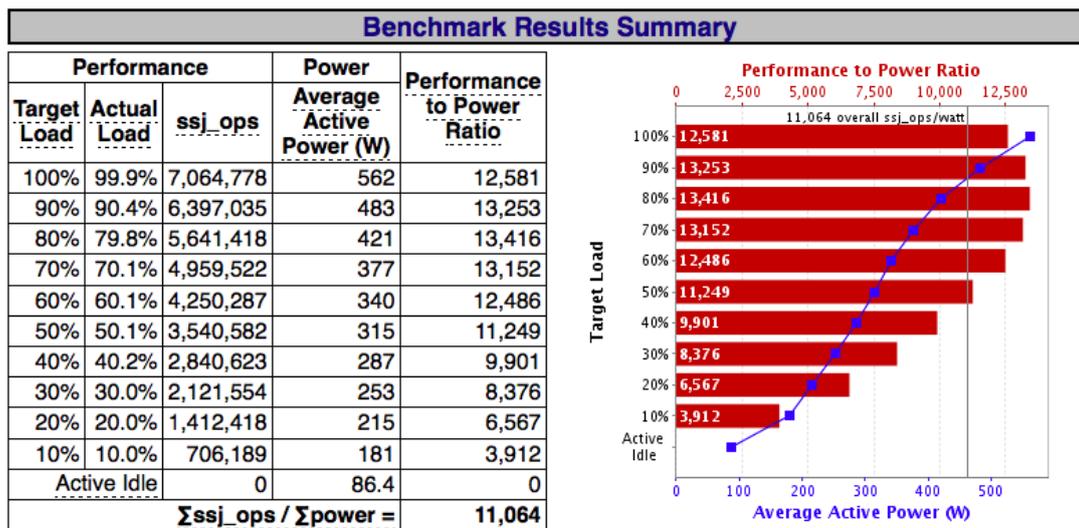


Abbildung 2.4.: Benchmark-Ergebnisse eines Servers mit dynamischer Leistungsaufnahme; Grafik unverändert aus [Sta16] übernommen

2.3. Serverressourcen: Betriebsmittel und deren Charakteristika

Die beschriebene, dynamische Leistungsaufnahme eines Servers geht auf die Betriebsmittel zurück, die für die Ausführung von Prozessen bereitgestellt werden. Diese Betriebsmittel haben bezüglich ihrer Leistungsaufnahme und des Verhaltens unter zunehmender Last sehr unterschiedliche Charakteristika, die im Folgenden aufgezeigt werden.

2.3.1. Prozessor

Mit der Bezeichnung Prozessor oder Central Processing Unit (CPU) eines Computers oder Servers ist im Allgemeinen auch der Chip selbst gemeint. Ein reiner Prozessor besteht in der Regel aus einem Rechenwerk, Speicher und einem Steuerwerk, sowie den Busleitungen für die Kommunikation. In dieser Arbeit soll der Begriff CPU ebenfalls den Chip selbst umfassen.

2.3. Serverressourcen: Betriebsmittel und deren Charakteristika

Die Leistungsaufnahme P_{CPU} der CPU kann mit:

$$P_{CPU} = C * f * U^2 + P_{static} \quad (2.3)$$

beschrieben werden, wobei C die Kapazität, f die Taktfrequenz, U die Betriebsspannung und P_{static} die Grundleistungsaufnahme ist. Besonderes Augenmerk liegt auf der Spannung, die mit quadratischer Auswirkung in die Leistungsaufnahme einfließt.

Um die Energie-Effizienz von Prozessoren zu erhöhen, haben die Hersteller reagiert. Intels SpeedStep[®] [Int04] und AMDs PowerNow[®] [Adv00] nutzen Techniken zur Reduzierung der Leistungsaufnahme bei verringerter CPU-Auslastung. Diese Technologien basieren auf dem Dynamic Voltage and Frequency Scaling (DVFS).

Mit Hilfe von DVFS-Techniken werden Frequenz und Spannung gesenkt, um die Leistungsaufnahme der CPU zu reduzieren und an den tatsächlichen Bedarf anzupassen und Energie zu sparen [LSH10, LNY07]. Diese Technologie ist die Basis für die Realisierung einer Server-Leistungsaufnahme in Korrelation zur Auslastung. Die Implementierung des Verfahrens setzt jedoch ein enges Zusammenwirken von Hard- und Software voraus.

2.3.2. Arbeitsspeicher

Der Arbeitsspeicher eines Servers stellt den Teil des Speichers dar, in den Programme inklusive der zugehörigen Daten für eine Ausführung geladen werden müssen.

Der Arbeitsspeicher ist häufig aus Dynamic Random Access Memory (DRAM) Speicherzellen aufgebaut und ein sogenannter flüchtiger Speicher, der technisch mittels Kapazitäten als Speichermedium realisiert wird. Im Gegensatz zu einer Festplatte werden die Daten nicht dauerhaft gespeichert. Wird die Spannungsversorgung des Arbeitsspeichers abgeschaltet, entladen sich die Kapazitäten und die Daten gehen verloren. Der Ladezustand der Kapazitäten des Arbeitsspeichers muss zyklisch geprüft und aufgefrischt werden, um die Speicherinhalte zu erhalten.

Der Großteil der Leistungsaufnahme entfällt hier auf die Controller, die im Bereich von Millisekunden den Ladezustand prüfen und diesen gegebenenfalls erhöhen müssen [CBJ10]. Aufgrund dieser technischen Realisierung ist die Leistungsaufnahme des Arbeitsspeichers nahezu konstant, unabhängig vom jeweiligen Datenfüllstand bzw. Ladezustand der Kapazitäten. Relevant für die Leistungsaufnahme ist vielmehr der Power-Status (Active, Idle oder Powerdown), in dem sich der Arbeitsspeicher befindet [EMW09, HSL⁺04].

Auch die Zugriffszeiten sind unabhängig vom Datenfüllstand des Arbeitsspeichers. Ob nur wenige Daten gespeichert sind oder der Speicher komplett belegt ist, beeinflusst die Zugriffszeiten nicht. Wird allerdings mehr Arbeitsspeicher benötigt, als physikalisch vorhanden ist, müssen Teile des Arbeitsspeichers auf die Festplatte ausgelagert und der Arbeitsspeicher um einen virtuellen Bereich erweitert werden, um weitere Daten aufnehmen zu können. Dieser Prozess ist Aufgabe des jeweiligen Betriebssystems.

2. Grundlagen

Die im physikalischen Arbeitsspeicher vorhandenen Daten sind innerhalb weniger Taktzyklen von der CPU zu verarbeiten. Bei den Daten im virtuellen Teil des Arbeitsspeichers, der auf die Festplatte ausgelagert wurde, verhält es sich anders. Die Zugriffszeiten sind durch den Technologie-Wechsel auf ein langsames Speichermedium erheblich verlängert. Werden Daten angefordert, die in dem virtuellen Teil des Arbeitsspeichers auf der Festplatte lagern, muss zunächst Platz im physikalischen Teil geschaffen werden, indem andere Daten auf die Festplatte ausgelagert werden. In einem zweiten Schritt können die angeforderten Daten in den physikalischen Teil kopiert werden.

Um diesen Effekt gering zu halten, existieren zahlreiche Verfahren, die eine möglichst geschickte Auswahl der ein- oder auszulagernden Daten vornehmen. Das definierte Ziel von heuristischen Verfahren in diesem Bereich ist es, Daten für die Auslagerung zu ermitteln, die seltener benötigt werden. Mit diesen Verfahren kann der Prozess des Ein- und Auslagerns von Speicherinhalten zwar optimiert werden, die Performance beim Zugriff auf den virtuellen Arbeitsspeicher ist jedoch um ein Vielfaches verlangsamt und bedeutet einen entsprechenden Effizienzrückgang.

2.3.3. Festplatte

Auf der Festplatte eines Systems werden Daten auf einem magnetischen Medium dauerhaft gespeichert. Auch nach Abschalten des Systems gehen die Daten nicht verloren. Festplatten können folglich in einen Standby-Modus versetzt und abgeschaltet werden, ohne einen Datenverlust zu erleiden.

Ähnlich wie bei einem Drucker kann auch der Schreib-Lese-Kopf einer Festplatte stets nur ein Datum lesen oder schreiben, nicht mehrere parallel. Besteht die Anforderung, mehrere Dateien parallel zu schreiben, wird ein Kontextwechsel durchgeführt. Tatsächlich werden beide Dateien im Wechsel geschrieben. Dabei ist die Geschwindigkeit beim parallelen Schreiben oder Lesen mehrerer Dateien deutlich langsamer als bei einer seriellen Verarbeitung. Verzögernd wirken sich hier die Zeiten aus, in denen der Kontextwechsel durchgeführt wird. In diesen Intervallen wird effektiv weder gelesen noch geschrieben, es erfolgt die Neupositionierung des Schreib-Lesekopfes auf der Magnetoberfläche, bei der die Speicherorte der jeweiligen Daten angefahren werden müssen.

Wie beschrieben, können Festplatten in einen Standby-Modus versetzt oder abgeschaltet werden. Wird von einer Festplatte über einen längeren Zeitraum keine Zulieferung oder Persistenz von Daten benötigt, kann dies eine sinnvolle Maßnahme sein. Wird zu einem späteren Zeitpunkt wieder Zugriff auf die Daten benötigt, muss die Festplatte zunächst reaktiviert werden. Bevor auf die Daten zugegriffen werden kann, muss die Spindel auf die korrekte Drehzahl beschleunigt werden. Dieser Vorgang kann mehrere Sekunden in Anspruch nehmen und die Zugriffszeit verlängert sich entsprechend. Bei Server-Systemen wird daher üblicherweise kein Standby-Modus für Festplatten vorgesehen. Bei konventionellen Festplatten in Serversystemen ist damit keine Varianz in der Leistungsaufnahme zu erwarten, zudem ergibt sich ein Effizienzverlust bei der Verarbeitung paralleler Schreib- und Lesevorgänge.

Storage Area Network

In größeren Rechenzentren wird überwiegend ein Storage Area Network (SAN) betrieben und ein zentrales Speichersystem verwendet. In den Servern werden zusätzliche Host Bus Adapter (HBA) eingebaut. Der SAN-HBA eines Server wird zusammen mit den SAN-HBAs weiterer Server an einen SAN-Switch angeschlossen, an dem wiederum Storage-Boxen angeschlossen sind, in denen sich die Festplatten befinden.

Diese Technologie bietet Performance- und weitere administrative Vorteile. Üblicherweise wird einem Server nicht eine konkrete Platte der Box zugewiesen, sondern nur ein logischer Speicherbereich.

Die Verwaltung und Optimierung des Speicherbereichs sowie die Allokation zu konkreten Festplatten in der Box übernimmt das SAN-Management-System. Der Speicherbereich eines Servers kann auf mehrere Festplatten verteilt werden, um so ein paralleles Lesen und Schreiben in der Performance zu optimieren. Denkbar wäre, dass einerseits häufig geschriebene und andererseits häufig gelesene Daten auf unterschiedlichen Festplatten allokiert werden. Auf die gleiche Weise lassen sich parallele Lese- und Schreibvorgänge trennen.

Der administrative Vorteil eines SAN-Systems besteht darin, dass die logischen Festplatten der Server im laufenden Betrieb verkleinert und erweitert werden können, man ist damit nicht an die fixe Größe einer im Server verbauten Festplatte gebunden.

Solid State Disk

Eine Weiterentwicklung der persistierenden Speichersysteme ist die Solid State Disk (SSD). Bei dieser Technologie ist das Speichermedium aus Flash-Speicherchips aufgebaut, es sind keine beweglichen oder mechanischen Komponenten vorhanden. Eine Verzögerung beim Lesen oder Schreiben von Daten aufgrund von erforderlichen Positionsänderungen des Schreib-Lesekopfes gibt es nicht. Durch diese Technologie bietet eine SSD gegenüber der rotierenden Festplatte erhebliche Performancevorteile. SSDs liegen aktuell preislich jedoch noch über konventionellen Festplatten, sodass der betriebene Anteil konventioneller Festplatten noch überwiegt.

Aufgrund betriebswirtschaftlicher Gesamtbetrachtungen wird bei der Anschaffung neuer Hardware allerdings vermehrt auf SSDs gesetzt. Der stetig fallende Preis, die geringere Abwärme und die höhere Speicherdichte zeigen in der Gesamtkalkulation einen nur marginal höheren Preis im Vergleich zur konventionellen Festplatte. Der Konzern, der dankenswerterweise die Datensätze für die Evaluation zur Verfügung gestellt hat, setzt daher bei Servern und auch im Bereich der SAN-Storage-Boxen auf SSD-Technologie.

Durch den Einsatz von SAN und SSDs kann die Effizienz bezüglich paralleler Prozesse optimiert und die Leistungsaufnahme des Speichersystems insgesamt reduziert werden. Eine signifikante, auslastungsbezogene Varianz in der Leistungsaufnahme ergibt sich jedoch nicht.

2. Grundlagen

2.3.4. Netzwerkkarte

Die Netzwerkkarte oder Network Interface Card (NIC) stellt die Kommunikationsschnittstelle eines Systems zum lokalen Netzwerksegment dar. Über die Netzwerkkarte eines Servers werden die Applikationsdienste zur Verfügung gestellt, ebenso erfolgt die Überwachung und Administration über die Netzwerkschnittstelle.

Bei der Leistungsaufnahme von Netzwerkkarten zeigt sich aktuell kaum Varianz, die an den übertragenen Datenverkehr gekoppelt wäre. Die Leistungsaufnahme hängt insbesondere von der maximal möglichen Übertragungsrates ab, die beim Verbindungsaufbau einmalig ausgehandelt und festgelegt wird und nicht von der tatsächlichen Transferrate [OP16]. Bei PCs mit aktivierter Wake On Lan Funktion, muss die Ethernet-Schnittstelle auch im ausgeschalteten Zustand weiter betrieben werden, um ein Wiedereinschalten über das Ethernet zu ermöglichen. Allerdings wird gerade bei Gigabit-Ethernet-Netzwerkkarten häufig der Wechsel auf die kleinste, mögliche Übertragungsgeschwindigkeit vorgenommen. Ist der PC abgeschaltet, wird daher oftmals nur eine 10MBit Schnittstelle weiter betrieben und die Leistungsaufnahme kann gegenüber dem Betrieb einer Gigabit-Schnittstelle reduziert werden.

Die Diskrepanz zwischen maximal möglicher Übertragungsrates und tatsächlich benötigter Bandbreite ist offensichtlich im privaten Haushalt besonders hoch. Laut einem Bericht des Spiegel [Spi07] werden hier Gigabit-Switche eingesetzt, obwohl ein 100-Megabit-Switch vollkommen ausreichend wäre und weniger Energie benötigt. Insbesondere wird der Anschluss von IP-TV-Boxen und Netzwerkdruckern kritisiert, da hier permanent eine Netzwerkschnittstelle mit hoher Bandbreite betrieben wird und die Nutzdauer nur gering ausfällt.

Bisher ist eine Geschwindigkeitsanpassung nur möglich, während die Verbindung hergestellt wird und kann im laufenden Betrieb nicht mehr geändert werden. Die so unnötigerweise aufgenommene Strommenge soll sich in den USA auf 5,8 Terawattstunden belaufen. Es wird auch berichtet, dass Cisco ein Methode vorgestellt hat, die Bandbreite dynamisch anzupassen, was in der Folge eine zur Datenübertragung korrespondierende Leistungsaufnahme bedeuten könnte. Auch Gunaratne et. al [GCNS08] haben ein Konzept zur Reduzierung der Leistungsaufnahme vorgestellt. Dabei wird die benötigte Datenrate bestimmt und mit dem angeschlossenen Switch eine Neusynchronisation mit angepasster Datenrate durchgeführt. Im Gegensatz zu der permanent mit maximaler Geschwindigkeit bestehenden Verbindung werden Einspareffekte über eine Reduzierung der Bandbreite erzielt.

Die beschriebenen Technologien befinden sich noch in der Entwicklung, aktuell weisen konventionelle Netzwerkschnittstellen keine signifikante Varianz in der Leistungsaufnahme auf.

2.4. Servervirtualisierung

Unter dem Begriff der Virtualisierung wird in diesem Kontext die Nachbildung eines Computersystems mit Hilfe von Software verstanden. Die Virtualisierungssoftware simuliert

das Vorhandensein von Hardwarekomponenten mit Hilfe eines Abstraktionslayers. Durch diese Technologie lassen sich virtuelle Server erzeugen und auch der Betrieb mehrerer virtueller Server auf einem physikalisch vorhandenen Server wird ermöglicht.

2.4.1. Hypervisor

Technisch betrachtet fußt die Virtualisierungstechnologie auf einem physikalisch vorhandenem Server. Die Ressourcen dieses Servers werden von einem sogenannten Hypervisor verwaltet, der somit die Abstraktionsschicht zwischen einem virtuellen Server und der physikalisch tatsächlich vorhandenen Hardware darstellt. Ein Hypervisor kann mehrere virtuelle Server beherbergen, wobei der Ressourcenzugriff vom Hypervisor geregelt wird. Der virtuelle Server stellt sich nun wie ein Hardware-Server dar, auf dem ein Betriebssystem installiert werden kann.

Der Begriff virtuelle Maschine (VM) wird im weiteren Verlauf der Arbeit häufiger verwendet. Mit VM wird hier die virtuelle Serverumgebung inklusive Betriebssystem und darin ausgeführten Applikationen verstanden.

Eine VM kann auf einem physikalischen Server (PM) betrieben werden, der einen Hypervisor zur Verfügung stellt. Im Rahmen dieser Arbeit wird unter dem Begriff PM generell ein physikalischer Server verstanden, der einen Hypervisor zur Verfügung stellt. Weiterhin ermöglicht der Hypervisor den parallelen Betrieb mehrerer dieser VMs. Das Konstrukt der VM kann mit Hilfe von Migrationsverfahren auf unterschiedliche PMs verschoben werden.

2.4.2. Migrationsverfahren

Wie beschrieben, stellt der Hypervisor eine generische Abstraktionsschicht zur eigentlichen Server-Hardware dar. Aufgrund dieses Umstands ist es möglich, eine VM auf verschiedenen PMs zu betreiben. Dafür muss lediglich das Dateisystem der VM auf dem jeweiligen Server zur Verfügung stehen. In großen Rechenzentren wird häufig ein SAN (vgl. 2.3.3) verwendet, über welches der Zugriff auf die Dateisysteme verschiedener VMs ermöglicht werden kann. Es besteht somit die Möglichkeit, eine VM auf unterschiedlichen PMs zu betreiben, wobei der Vorgang des PM-Wechsels als Migration einer VM bezeichnet wird.

Migration einer VM

Um eine VM von einer PM auf eine andere PM zu migrieren, gibt es unterschiedliche Verfahrensweisen. Bei der Offline-Migration wird das Betriebssystem der VM angehalten bzw. heruntergefahren, dieser Zustand entspricht einem abgeschalteten, physikalischen Server. In einem zweiten Schritt wird die VM auf dem Ziel-Server wieder gestartet. Während dieser Migration ist die VM allerdings nicht verfügbar, die Migration findet mit einer Betriebsunterbrechung (offline) statt. Angestrebt wird jedoch die Migration möglichst ohne Betriebsunterbrechungen (online) durchzuführen und die Applikationen verfügbar zu halten.

2. Grundlagen

Mit dem Verfahren der sogenannten Live-Migration, wird die Migration einer VM vorgenommen, ohne die VM oder das Betriebssystem herunterzufahren. Dieses Verfahren ist jedoch ungleich komplexer.

Live-Migration

Die Live-Migration einer VM ist in mehrere Phasen unterteilt. Grundsätzlich ist der RAM-Inhalt der VM von der Quell-PM auf die Ziel-PM zu kopieren. Da die VM während des Kopiervorgangs in Betrieb bleibt, kommt es zu Änderungen des RAM-Inhaltes. Geänderte RAM-Seiten müssen fortlaufend auf der Ziel-PM aktualisiert werden. Schließlich wird die VM kurz angehalten, um einen synchronen Zustand herstellen zu können, im Anschluss erfolgt die Umschaltung auf die VM-Instanz der Ziel-PM. Dabei liegt das Intervall zwischen Anhalten der VM und Aktivierung auf der Ziel-PM im Rahmen weniger Millisekunden, wie die Untersuchungen von Semenov et. al [SVT11] und Clark et. al [CFH⁺05] gezeigt haben.

2.4.3. Kosten einer Live-Migration

Bei der Migration von VMs werden mehrere Arbeitsschritte ausgeführt, wie beschrieben. Zur Ermittlung der im Rahmen von Live-Migrationen entstehenden Kosten wurden bereits diverse Untersuchungen durchgeführt. Es konnte gezeigt werden, dass sich bei älterer Hardware signifikante Effekte, wie eine deutliche Verlangsamung der Migration bei Volllastung der Server zeigen. Diese Effekte treten durch VM-Migrationen auf aktueller Hardware eher in den Hintergrund, wie Semenov et. al [SVT11] herausgefunden haben. Es wurde Hardware verwendet die über Erweiterungen speziell für Virtualisierungstechniken verfügt, die von Intel VT-x genannt wird [NSL⁺06].

Um die Auswirkungen einer Migration auf die Leistungsaufnahme von Servern zu betrachten, haben Strunk et. al [SD13] Untersuchungen an Hardware vorgenommen, die Virtualisierungstechnologien direkt unterstützt. In einem untersuchten Beispiel konnte eine Zunahme der Server-Leistungsaufnahme von 3 - 5 Watt registriert werden. Als Referenz für den Anstieg der Leistungsaufnahme wurde die Leistungsaufnahme desselben Servers im Idle-Status herangezogen. Die Untersuchungen wurden mit mehreren VMs unterschiedlicher RAM-Größe durchgeführt. Die Ergebnisse sind in Abbildung 2.5 dargestellt.

Neben der Leistungsaufnahme ist auch die Dauer einer Migration von Bedeutung. Insbesondere ist die Frage relevant, inwieweit die Auslastung der Server die Migrationskosten beeinflusst. Hierzu haben Verma et. al [VAN08] Untersuchungen durchgeführt und beobachtet, dass die Migrationskosten lediglich von den Charakteristika der VM abhängen und somit im Vorfeld berechenbar sind, die Lastsituation des Servers spielt hier keine Rolle.

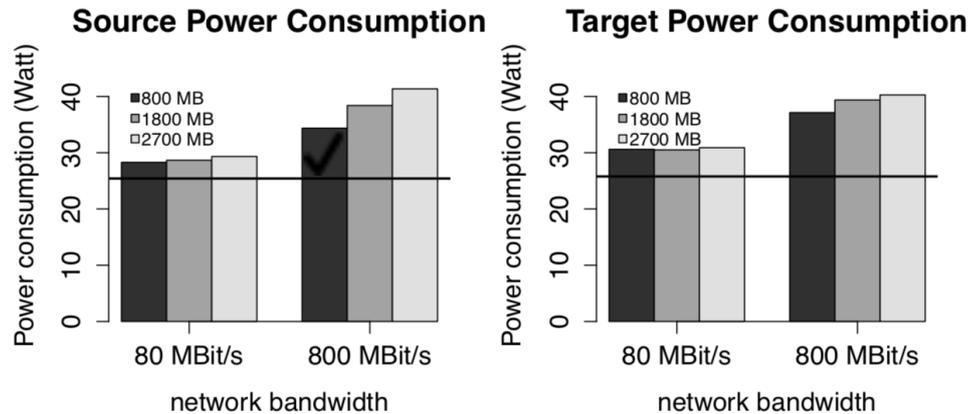


Abbildung 2.5.: Durchschnittliche Leistungsaufnahme von Quell- und Ziel-Server während der Migration, verglichen mit der Idle-Leistungsaufnahme (horizontale Linie); Grafik unverändert aus [SD13] zitiert

2.5. Service Level Management (SLM)

Wie bereits in Abschnitt 1.1 beschrieben, sind Service Level Agreements im Kontext des Rechenzentrumsbetriebs ein wichtiger Vertragsbestandteil.

Die Entscheidung, eine Applikation in einem Rechenzentrum zu betreiben, ist oftmals einem erhöhten Sicherheitsbedürfnis bezüglich Ausfallsicherheit und Performance geschuldet. Zur vertraglichen Absicherung dieser Beweggründe werden Service Level Agreements (SLAs) geschlossen. In den SLAs werden sogenannte Pönale (Vertragsstrafen) festgelegt, die bei Nichteinhaltung der zugesicherten Leistung an den Kunden zu zahlen sind. Konkreter werden in den Service Level Objectives (SLOs) messbare Größen definiert, deren Einhaltung im SLA festgeschrieben wird. Die Ausgestaltung und Einhaltung von SLAs kann sich als sehr komplex erweisen, daher erfolgt die Verwaltung und Überwachung oftmals in einer SLM-Applikation, wie *Digital Fuel ITBM* [Ama19]. Die SLM-Applikation aggregiert Monitoring-Daten, aus denen sich die Performance einer Applikation bzw. die Einhaltung der vereinbarten SLAs ableiten lässt. Die zentrale Verwaltung der SLAs und Auswertung der aktuellen Servicekennzahlen schafft eine Transparenz über den Service-Status und kann gegebenenfalls Handlungsbedarf aufzeigen. Weiterhin können Berichte über die SLA-Erfüllung mit Abbildung von Geschäftszeiten, Feiertagen, Wartungsintervallen usw. erzeugt werden. Letztlich ist auch die automatische Veranlassung von Gutachten entsprechend der festgesetzten Pönalen bei einer Verletzung von SLAs möglich.

2.6. Effizienzbewertung von Rechenzentren

Zur Bewertung der Effizienz von Rechenzentren existieren eine ganze Reihe an Metriken. Die Power Usage Effectiveness (PUE) ist inzwischen in die ISO-Norm ISO/IEC 30134-

2. Grundlagen

2:2016 eingeflossen und ist durch:

$$PUE = \frac{TotalFacilityEnergy}{IT - EquipmentEnergy} \quad (2.4)$$

definiert. Bei der PUE wird die Leistungsaufnahme der IT-Geräte ins Verhältnis zur Leistungsaufnahme der Geräte gesetzt, die alle weiteren unterstützenden Leistungen erbringen, wie Klimatisierung, USV usw. Weitere Faktoren fließen nicht ein. Wie auch in [Sch16] beschrieben, berücksichtigen diverse Metriken zur Bewertung der Effizienz (wie PUE) die tatsächlich von den IT-Systemen erbrachte Rechenleistung jedoch nicht, die Effizienz der Server ist ausgeklammert. Die PUE bewertet somit die Effizienz der Support-Infrastruktur, die Effizienz der eingesetzten Server ist folglich separat zu betrachten.

2.6.1. Effizienz von Servern

Der Wirkungsgrad von technischen Maschinen, Geräten und Komponenten kann über ihren Leistungsbereich stark variieren. Häufig existiert ein spezifischer Leistungsbereich, in dem der Wirkungsgrad ein Maximum annimmt. Im Leistungsbereich darüber oder darunter ist der Wirkungsgrad rückläufig. Bei Servern ist der Wirkungsgrad im unteren Leistungsbereich oftmals geringer, steigt mit zunehmender Leistung an und fällt zum Erreichen der maximalen Leistung wieder ab.

In Abbildung 2.6 ist die Performance eines Fujitsu Servers dargestellt, der genau das beschriebene Verhalten zeigt. Die Leistung des Servers ist in *Server Side Java Operations Per Second* (ssj_ops) angegeben, der erste Industriestandard-Benchmark zur Messung sowohl der Leistung als auch der Leistungsaufnahme von Servern [Sta12]. Es ist erkennbar, dass sich die maximale Effizienz des Servers bei etwa 80% Auslastung erzielen lässt. Die Leistungsaufnahme des Servers skaliert mit der Auslastung. Ist ein möglichst effizienter Betrieb der Server ein definiertes Betriebsziel, wäre für Server dieses Typs eine Serverauslastung von 80% anzustreben.

In Abbildung 2.7 sind die Daten Sun Netra X4250 dargestellt. Der Sun Netra X4250 ist ein Server, der kaum hinsichtlich der Auslastung skaliert. Eine effiziente Nutzung setzt hier eine möglichst hohe Auslastung voraus.

2.7. Regelleistung

Der elektrische Strom ist eine Energieform, die nicht im Netz selbst gespeichert oder gepuffert werden kann, wie bereits in Abschnitt 1.1 beschrieben. Die Stromeinspeisung muss daher stets im selben Maße des Strombedarfs erfolgen. Um die Stabilität des Stromnetzes zu gewährleisten, sind die Übertragungsnetzbetreiber in ihrer jeweiligen Regelzone in der Verantwortung für ein Leistungsgleichgewicht zu sorgen [Bun19]. Abweichungen zwischen Erzeugung und Entnahme müssen durch den Einsatz von sogenannter Regelleistung



Abbildung 2.6.: Benchmark des Servers Fujitsu PRIMERGY TX2540 M1; Grafik unverändert aus [Sta15] übernommen

2. Grundlagen



Abbildung 2.7.: Benchmark des Servers Sun Netra X4250; Grafik unverändert aus [Sta09] übernommen

oder Regelleistung ausgeglichen werden. Die aktuelle Balance zwischen erzeugter und entnommener Energie lässt sich an der Netzfrequenz erkennen. Wird mehr Energie aus dem Stromnetz entnommen, als eingespeist wird, sinkt die Netzfrequenz. Im umgekehrten Fall steigt die Netzfrequenz.

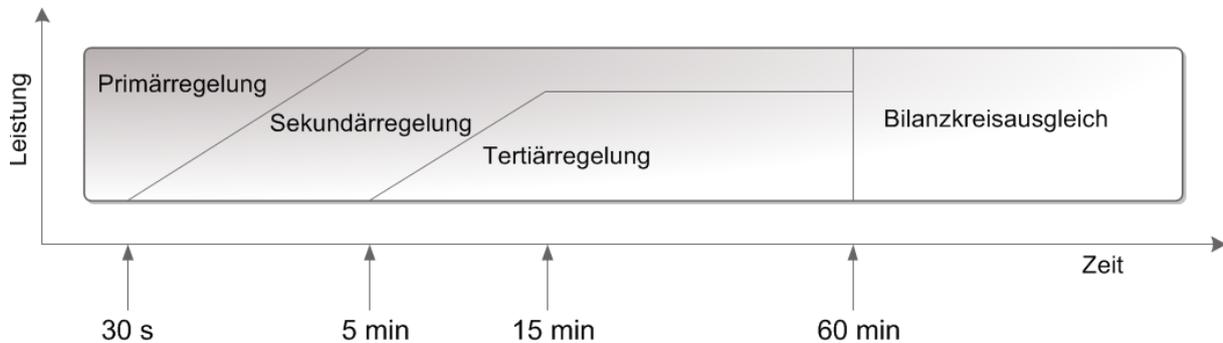


Abbildung 2.8.: Zeitliche Abfolge der Aktivierung von Regelleistung

Mit Regelleistung wird die Energie bezeichnet, die ein Netzbetreiber für unvorhersehbare Leistungsschwankungen bereitstellen muss. Da die Leistungsschwankungen sowohl positiv als auch negativ ausfallen können, muss auch die Regelleistung für beide Fälle vorgehalten werden. Es stehen drei Arten von Regelleistung zur Verfügung, die in einem kaskadierten Verfahren (s. Abbildung 2.8) die Stabilisierung des Stromnetzes sicherstellen:

- *Primärregelenergie*: Sie ist die in der ersten Stufe zu aktivierende Energie und wird zur schnellen Stabilisierung des Netzes innerhalb von 30 Sekunden benötigt.

Anbieter von Primärregelenergie messen die Netzfrequenz und aktivieren die Regelleistung automatisch, sobald die Netzfrequenz über 50,01 Hertz steigt oder unter 49,99 Hertz fällt. Ist die Netzfrequenz innerhalb des beschriebenen Intervalls, erfolgen keine Maßnahmen. Beim Verlassen des Intervalls werden die Leistungsreserven proportional zur registrierten Netzfrequenzabweichung aktiviert, um das Netz zu stabilisieren.

Nicht alle Kraftwerke sind zur Vorhaltung von Regelleistung geeignet. Zur Prüfung der Voraussetzungen gibt es eine entsprechende Präqualifikation [50H19], es muss ein entsprechender Nachweis erbracht werden.

Der Reaktor von Kernkraftwerken lässt sich aus technischen Gründen nicht in der geforderten Zeitspanne regulieren. Kernkraftwerke nehmen daher typischerweise nicht am Prozess der Primärregelung teil. Für die Bereitstellung von Primärregelenergie kommen insbesondere thermische und hydraulische Großkraftwerke in Betracht, die eine schnelle Leistungsänderung umsetzen können [Con14].

- *Sekundärregelenergie*: Die Primärregelenergie muss nach 30 Sekunden in voller Höhe erbracht und für 15 Minuten zur Verfügung stehen. Innerhalb dieser Zeitspanne löst die Sekundärregelenergie die Primärregelenergie vollständig ab, um die Primärregelenergie wieder für neue Stabilisierungsmaßnahmen zur Verfügung zu stellen.

2. Grundlagen

Bereits nach 30 Sekunden muss eine Reaktion mit Sekundärregelenergie erkennbar sein und innerhalb von 5 Minuten muss die gesamte Leistung erbracht werden können. Kraftwerke, die in der Lage sind, derartige Anforderungen zu bedienen, sind z.B. Gas-Kraftwerke oder Pumpspeicherwerke.

- *Tertiärregelung*: Die Tertiärregelung oder Minutenreserve dient zur weiteren Stabilisierung des Stromnetzes. Die Regelleistung muss innerhalb von 15 Minuten erbracht werden können und löst die Sekundärregelenergie bei Netzschwankungen ab, die über 15 Minuten hinausgehen. Das Ziel ist es auch hier, die vorgelagerte Regelenergie abzulösen und für weitere Ereignisse im Stromnetz bereitzuhalten.

Die Aktivierung der Minutenreserve erfolgt nicht automatisiert, sie wird von den verantwortlichen Übertragungsnetzbetreibern fallweise aktiviert. Es wird die tatsächliche Inanspruchnahme der Sekundärregelenergie beurteilt und die absehbare weitere Entwicklung berücksichtigt, bevor die Minutenreserve abgerufen wird. Ereignisse, die zu einer Aktivierung führen, sind z.B. Prognosefehler oder Kraftwerksausfälle [Con14].

- *Bilanzkreisausgleich*: Die kleinste Einheit des Energiemarktmodells wird als Bilanzkreis bezeichnet und ist ein virtuelles Energiemengenkonto. Prinzipiell soll das Stromnetz durch ein Fahrplanmanagement zur Ein- und Ausspeisung von Energiemengen stabil gehalten werden. Es bildet die Basis für ein stabiles Stromnetz, während die zuvor beschriebenen Regelungen die Auswirkungen von unvorhergesehenen Ereignissen kompensieren müssen. Verantwortlich für den Bilanzkreisausgleich sind die Übertragungsnetzbetreiber.

2.8. Zusammenfassung

Die Technologien und Methoden, die diese Arbeit tangieren, wurden in diesem Kapitel kurz vorgestellt.

Die Leistungsaufnahme von Rechenzentren setzt sich aus den Leistungsaufnahmen von mehreren relevanten Komponenten zusammen. Dabei gibt es Komponenten, deren Leistungsaufnahme unabhängig von der Auslastung der betriebenen Server ist, wie die Beleuchtung oder die Bereitschafts-Heizung von NEAs. Andere Komponenten, wie die Klimatisierungssysteme, adaptieren ihre Leistungsaufnahme und stehen im Einfluss der Serverauslastung. Damit ist die Auslastung der Server und die damit verbundene Leistungsaufnahme der Ausgangspunkt zur Beeinflussung der Leistungsaufnahme eines Rechenzentrums.

Die Betriebsmittel eines Servers zeigen je nach Auslastung unterschiedliche Charakteristika, was Performance und die Leistungsaufnahme betrifft. Die CPU zeigt in Kombination mit DFVS ein adaptives Verhalten, entsprechend der Auslastung variiert auch die Leistungsaufnahme.

Eine Basistechnologie für das Ressourcenmanagement in Rechenzentren ist die Servervirtualisierung, deren Funktionsweise und das Verfahren der Live-Migration von VMs wurden

erläutert.

Weiterhin wurden die Betriebsmodelle für Rechenzentrumsdienstleistungen beschrieben, sowie die im Rahmen des SLM definierten SLAs.

Um die Anforderungen an das zu implementierende Verfahren ableiten zu können, wurden die etablierten Maßnahmen und Prozesse zur Stabilisierung des Stromnetzes erläutert.

3. Stand der Technik

In diesem Kapitel werden verschiedene Ansätze zur effizienten Energienutzung und adaptiven Leistungssteuerung in Rechenzentren vorgestellt. Das Hauptaugenmerk liegt dabei auf den eingesetzten Methodiken zur Beeinflussung der elektrischen Leistungsaufnahme.

Zunächst werden Technologien beleuchtet, die an der Server-Hardware und Komponenten-Steuerung ansetzen. Weitere Ansätze steuern die Server-Auslastung über den Ressourcenbedarf der Applikationen, wobei der Fokus hierbei auf unterschiedliche Optimierungsziele gerichtet ist. Zur Beeinflussung des Ressourcenbedarfs werden ebenfalls verschiedene Ansätze verfolgt, die hier aufgezeigt werden.

Zusammenfassend lassen sich die Ansätze in methodisch verwandte Kategorien einteilen, jede Kategorie wird bezüglich ihrer Verfahrenseigenschaften erläutert und zugehörige Arbeiten benannt.

In Abschnitt 3.9 wird ein Kriterienkatalog zur Bewertung der Ansätze bezüglich ihrer Eignung für das Nutzungsszenario aufgebaut. Die vorgestellten Verfahren werden hinsichtlich dieser Kriterien betrachtet und tabellarisch dargestellt.

3.1. Low-Power Technologie

Bei der Suche nach Methoden zur Reduzierung der Leistungsaufnahme von Rechenzentren stellt sich die Frage, ob sogenannte Low-Power-Technologien, wie sie auch bei mobilen Endgeräten Anwendung finden, für diese Aufgabe geeignet sind und Effizienz-Vorteile bieten. Meisner et. al [MW11] haben die Leistungsaufnahme von Servern untersucht und dabei auch eine Bewertung der Effizienz durchgeführt. Für eine effiziente Energienutzung ist der Einsatz von Low-Power Technologie allein nicht zielführend, in konkreten Fällen kann die Energieaufnahme sogar steigen und die Effizienz rückläufig sein. Der Fokus liegt hier auf der Bereitstellung einer Rechenleistung generell, mit möglichst geringer Leistungsaufnahme. Bei High Performance Systemen dagegen ist das Design auf Höchstleistung ausgelegt, hier zeigt sich die beste Effizienz bei einer hohen Serverauslastung.

Während es unter dem Gesichtspunkt der Effizienz sinnvoll sein kann, bestimmte Komponenten hoch auszulasten, kann dieses Szenario bei anderen Komponenten negativ zu Buche schlagen. In [SPV09] werden diverse Modelle für Komponenten von Rechenzentren vorgestellt und deren Leistungsaufnahme analysiert. Beispielsweise nimmt die Effizienz einer USV bei höherer Auslastung ab. Es sind etwa 9% Verlust-Leistung bei einer hundertprozentigen Auslastung der USV zu erwarten. Eine separate (Effizienz-) Betrachtung einzelner Komponenten scheint daher sinnvoll.

3.2. Throttling

Throttling ist eine Technik, mit der über den CPU-Takt direkter Einfluss auf die Leistungsaufnahme der Server und somit auch mittelbar auf das gesamte Rechenzentrum inklusive Klimatisierung, USV usw. genommen werden kann. Hierbei handelt es sich jedoch um einen groben Eingriff in die Abläufe auf einem Serversystem. Die CPU wird hier ohne Beteiligung des Betriebssystems ausgebremst. Als Folge wird hier nicht nur die Leistung der auf dem Server befindlichen Applikationen reduziert, auch betriebssystemeigene Prozesse und Dienste werden gebremst. Somit ist diese Technik kein probates Mittel zur Steuerung der Energieaufnahme, da es hier zu unkontrollierbaren Performanceeinbrüchen der Applikationen kommen kann, wie in [NHSS09] beschrieben wird.

Throttling dient vorrangig dem Zweck, im Störfall eventuellen Hardwareschäden vorzubeugen. Für einen kontrollierbaren Einsatz müssten weitere Techniken kombiniert werden, um Applikation und Betriebssystem in das Verfahren mit einzubeziehen.

3.3. Server-Napping

Analog zur DVFS-Technologie (vgl. Abschnitt 2.3.1) haben Meisner et. al [MGW09] ein Verfahren entwickelt, das sie „PowerNap“ nennen. Dabei werden Komponenten des Servers in einen Ruhezustand versetzt. Wird der Server angefordert, werden die Komponenten wieder aktiviert und die anstehenden Tasks können verarbeitet werden. Damit der Server entsprechende Signale empfangen kann, wird die Netzwerkkarte nicht in den Napping-Zustand versetzt.

Bei diesem Ansatz gibt es zwei Betriebszustände des Servers, den Napping-Zustand und den Normalbetrieb. Ziel des Ansatzes ist es, die Leistungsaufnahme von Servern im Idle-Zustand weiter zu senken. Im konkreten Beispiel eines HP-Blade-Servers mit einer maximalen Leistungsaufnahme von 450 Watt und 270 Watt im Idle-Zustand konnte die Leistungsaufnahme im PowerNap-Zustand auf 10,4 Watt gesenkt werden.

In Abbildung 3.1 sind die Ergebnisse eines Server-Benchmarks dargestellt. Die Leistungsaufnahme variiert je nach Auslastung und liegt im Idle-Zustand bei 86,4 Watt, wie in der letzten Zeile der Ergebnistabelle ersichtlich. Das Konzept des „PowerNap“ ermöglicht es, diese Leistungsaufnahme weiter zu senken, wie beschrieben. Eine gezielte Steuerung der Leistungsaufnahme ermöglicht das Verfahren jedoch nicht.

3.4. Optimierung der Server-Klimatisierung

Wie in Abschnitt 2.1 dargestellt, entfällt ein Großteil der Leistungsaufnahme eines Rechenzentrums auf die Klimatisierung der Server-Räume. Viele Ansätze konzentrieren sich daher auf die Klimatisierung von Rechenzentren [PSBB02, BPS06, BF07, PP09].

3.5. Scheduling-Verfahren zur Steuerung der Server-Leistungsaufnahme

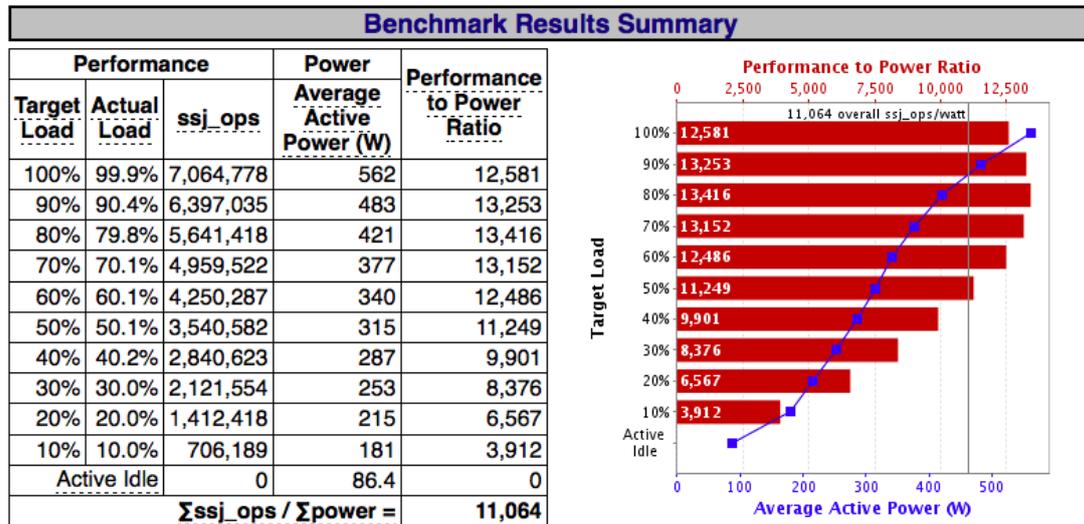


Abbildung 3.1.: Leistungsaufnahme eines Servers in Abhängigkeit zur Auslastung; Grafik unverändert aus [Sta16] übernommen

In [TGV08] wird explizit die Effizienz der Klimatisierung betrachtet. Durch gezieltes, temperaturabhängiges Platzieren von Jobs auf der Serverlandschaft werden Temperaturspitzen in den Server-Racks vermieden. Es wird eine gleichmäßige Auslastung der betriebenen Server angestrebt. Die Server im Rechenzentrum haben durch diese Technik eine annähernd gleiche Temperatur und die Entstehung von Hotspots kann verhindert werden. Um Hotspots ausreichend kühlen zu können, wäre eine entsprechend kühle Luftzufuhr nötig, da hier nun keine Hotspots mehr entstehen, kann die Temperatur der Kühlkreisläufe insgesamt erhöht werden und die Klimatisierungssysteme können effizienter arbeiten. Es ergibt sich eine höhere Temperaturdifferenz an den Wärmetauschern, die zu einer insgesamt höheren Effizienz des Klimatisierungssystems führt. Simulationsergebnisse zeigen mögliche Energieeinsparungen von 20% bis 30% durch den Einsatz des Verfahrens.

Bei dem beschriebenen Verfahren wird über eine Steuerung der Serverauslastung die Effizienz des Klimatisierungssystems erhöht, eine gezielte Steuerung der Server-Leistungsaufnahme wird nicht realisiert. Die hier verwendete Methodik wird jedoch in weiteren Verfahren mit Scheduling-Ansatz aufgegriffen.

3.5. Scheduling-Verfahren zur Steuerung der Server-Leistungsaufnahme

Um die Leistungsaufnahme von Rechenzentren an die Verfügbarkeit von Windenergie zu adaptieren, wird in [KGA⁺11] ein Algorithmus vorgestellt, der je nach verfügbarer Windenergie Jobs auf einen Serververbund verteilt. Für jeden eintreffenden Job ist eine Deadline und die Ausführungsdauer bekannt. Aus der Deadline und der Ausführungsdauer ergibt sich ein Zeitpunkt, zu dem der Job spätestens gestartet werden muss, um die

3. Stand der Technik

rechtzeitige Verarbeitung sicherzustellen. Ein Scheduler übernimmt dabei die Aufgabe, die Jobs in einer priorisierten Reihenfolge zu starten. Das Starten und Stoppen wird je nach aktueller Energieverfügbarkeit initiiert.

Als problematisch haben sich die raschen Wechsel in der Energieverfügbarkeit erwiesen, was unter anderem dazu führte, dass Jobs nicht bis zum Erreichen der Deadline verarbeitet wurden. Dieses Verhalten kann dazu führen, dass SLOs nicht eingehalten werden können und Vertragsstrafen drohen. Um diese Problematik zu entschärfen arbeiten andere Ansätze mit mehreren Rechenzentrumsstandorten.

In [CCC14] wird ein Ansatz beschrieben, mit dem Rechenzentren die Stabilität des Stromnetzes unterstützen können. Es soll eine Regelbarkeit der Leistungsaufnahme von Rechenzentren erreicht werden, um die raschen Wechsel in der Verfügbarkeit von erneuerbaren Energien kompensieren zu können. Weiterhin soll die variable Leistungsaufnahme als Service zur Regulation angeboten werden. In einem Schedulingverfahren wird eine FIFO (First In First Out)-Queue implementiert. Je nach energetischer Anforderung werden Jobs aus dieser FIFO-Queue auf den Servern zur Ausführung gebracht und so die Auslastung der Server gesteuert. Die Leistungsaufnahme der Server adaptiert sich dabei an die Auslastung mittels DVFS. Es wird eine mögliche Leistungsreduzierung von etwa 50% beschrieben, ohne maßgebliche Einbußen in der Servicequalität.

3.6. Geografische Migration zur Nutzung lokaler Standortvorteile

Ein Ansatz, um die Energiekosten für IT-Services zu reduzieren, kann die geografische Verlagerung der Services sein. Wie bereits beschrieben, kann so ein lokaler Vorteil durch einen niedrigen Energiepreis bestehen. Es existieren jedoch weitere Aspekte, die einen Standort attraktiv für den Betrieb von IT-Services machen. Xu et.al [XFL13] nutzen Temperaturunterschiede an unterschiedlichen Standorten verteilter Rechenzentren. Die Verarbeitung von Jobs wird an Standorte mit kühlerer Außentemperatur verlagert, um so den Energiebedarf der Klimatisierung zu reduzieren.

Der Ansatz von Rao et. al [RLL10] hat das Ziel, die Stromkosten zu senken, und orientiert sich direkt an einem lokalen Energiepreis, um die IT-Last auf verschiedene Rechenzentrumsstandorte zu verteilen. Dabei sollen jedoch weiterhin die vereinbarten SLOs gehalten werden. Es werden die Energiepreise an unterschiedlichen Standorten verglichen und die IT-Last entsprechend des Preisgefälles verteilt. Durch das vorgestellte Verfahren konnte im konkreten Fall eine Kostenersparnis von bis zu 30,15% erreicht werden. Allerdings setzt dieses Verfahren voraus, dass mehrere Rechenzentrumsstandorte zur Verfügung stehen. Weiterhin müssen diese geographisch so verortet sein, dass sich ein relevanter Unterschied in der Energieverfügbarkeit und dem Preisgefüge ergibt.

Liu et. al [LCB⁺12] verfolgen das Ziel, den CO₂-Fußabdruck von Rechenzentren zu reduzieren. Sie berücksichtigen in ihrem Verfahren sowohl die Verfügbarkeit von erneuerbaren

3.7. Einsatz von Server-Virtualisierungstechnologie

Energien als auch die klimatischen Gegebenheiten. Insbesondere ist die Frage von Bedeutung, ob der Einsatz einer Freiluftkühlung (siehe Abschnitt 2.1.2) an einem potentiellen Rechenzentrumsstandort möglich ist. Sie kombinieren in ihrem Verfahren die Verfügbarkeit von erneuerbaren Energien und die klimatischen Gegebenheiten. Bei der Platzierung von Batch-Jobs werden diese Faktoren berücksichtigt. Es wird eine mögliche Reduktion des Einsatzes von nicht-erneuerbaren Energien um 60% beschrieben.

In [LLW⁺11] und [ZWW11] werden Verfahren vorgeschlagen, bei denen die Rechenlast an Standorte verlagert wird, an denen erneuerbare Energie zur Verfügung steht. Andere Ansätze konzentrieren sich auf „follow the sun“ wie in [MDKS12] beschrieben. Diverse weitere regionale Faktoren sind denkbar. Die geografische Verschiebung von Rechenleistung ist jedoch erst in größeren Szenarien mit verteilten Rechenzentren sinnvoll einsetzbar, da sich die Gegebenheiten bei dicht beieinander gelegenen Standorten üblicherweise nicht signifikant unterscheiden.

3.7. Einsatz von Server-Virtualisierungstechnologie

Das Ressourcenmanagement in Rechenzentren ist bereits in diversen Arbeiten [Alb10, KSK14, GJN⁺10] detailliert untersucht worden. Die Ansätze verfolgen das Ziel einer effizienten Ressourcenverteilung mit Hilfe von Server-Virtualisierungstechnologie.

Über die Einführung von virtuellen Maschinen in Verbindung mit einem Lastmanagement können die darunterliegenden physikalischen Server bezüglich ihrer Auslastung optimiert werden. Bestehende Algorithmen verfolgen die Konsolidierung möglichst vieler virtueller Maschinen auf einem physikalischen Server. So können ungenutzte physikalische Server abgeschaltet werden, um Energie einzusparen. Ghribi et. al [GHZ13] optimieren explizit die Leistungsaufnahme von Rechenzentren unter Berücksichtigung des Ressourcenbedarfs der einzelnen VMs und Kapazitäten der PMs.

In [CFF12] wird ein Modell auf Basis des Open-Source Produkts OpenStack vorgestellt. Die Platzierung der virtuellen Maschinen wird hauptsächlich anhand von Leistungsaufnahme, Netzwerkauslastung und Server-Ressourcen vorgenommen. Ein Scheduler hat Zugriff auf eine zentrale Datenbank, in der die relevanten Serverparameter gepflegt werden. Im Bedarfsfall entscheidet der Scheduler, auf welchem physikalischen Server eine neue virtuelle Maschine instantiiert wird. Die Entscheidung über den Server, der zur Initialisierung der neuen virtuellen Maschine ausgewählt wird, basiert auf der jeweils aktuellen Ressourcenlage. Eine spätere Justierung, falls sich Änderungen ergeben, findet jedoch nicht statt.

3.7.1. Effiziente Nutzung des Arbeitsspeichers

Bei der Konsolidierung von virtuellen Maschinen auf einem physikalischen Server kann der Arbeitsspeicher ein limitierender Faktor sein. Der physikalische Server muss mindestens so viel freien Arbeitsspeicher aufweisen, wie die virtuelle Maschine, die zusätzlich auf

3. *Stand der Technik*

diesen Server migriert werden soll, benötigt. Ist nun der Arbeitsspeicher des physikalischen Servers belegt, können keine weiteren virtuellen Server auf diesen physikalischen Server konsolidiert werden. Nachteilig ist dies insbesondere, wenn die virtuellen Maschinen auf diesem physikalischen Server die Serverressourcen nicht auslasten, die CPU beispielsweise nur mäßig ausgelastet wird und somit noch Kapazitäten ungenutzt bleiben.

Um nun weitere virtuelle Server auf diesen physikalischen Server konsolidieren zu können, kann die Technologie des Memory-Sharing eingesetzt werden. Virtuelle Server werden mit Hilfe von sogenannten Hypervisoren betrieben, die den virtuellen Servern die Systemressourcen zur Verfügung stellen, wie in Abschnitt 2.4.1 beschrieben. Der Hypervisor kann nun identische Seiten im Arbeitsspeicher identifizieren und zu einer Seite zusammen fassen. Es existieren in der Folge zwei Referenzen, die auf dieselbe Seite verweisen. Soll eine der Seiten im Nachhinein verändert werden, wird der gemeinsam genutzte Speicher aufgegeben und es erfolgt eine separate Ablage mit einer neuen Referenz.

Das Potential für Konsolidierungen kann sehr hoch ausfallen, wie Barker et. al in [BWSS12] beschrieben haben. Oftmals werden Bibliotheken im Arbeitsspeicher vorgehalten, die zum Betriebssystem gehören, also öfter vertreten sind. Andere Bibliotheken sind Teil von Software-Umgebungen, wie z.B. Java, die ebenfalls häufiger vorkommen, eine gewisse Größe haben und somit Konsolidierungspotenzial bieten.

Wood et al. [WTL⁺09] nutzen diese Technologie für ihren Konsolidierungsansatz. Im Vorfeld der Konsolidierung wird eine Analyse der virtuellen Maschinen vorgenommen und ihr Memory-Sharing-Potential untersucht. Auf diese Weise kann bei einer anschließenden Konsolidierung von virtuellen Maschinen deren Memory-Sharing-Potential berücksichtigt und Kandidaten mit hohem Potential bevorzugt werden. So können entsprechend mehr virtuelle Maschinen konsolidiert, der physikalische Server besser ausgelastet und damit effizienter betrieben werden.

3.7.2. Optimierung der Netzwerkkommunikation

Große Applikationen, die in einem Rechenzentrum betrieben werden, bestehen oftmals aus mehreren Teil-Applikationen. Häufig gibt es logische Applikationsanteile, die einzelne fachliche Aufgaben abbilden und zu einer komplexen Applikation orchestriert werden. Als Persistenz-Schicht wird häufig eine Datenbank verwendet. Die Schnittstellen dieser Komponenten werden über das Netzwerk des Rechenzentrums erreicht und es kann zu kaskadierten Aufrufen der Schnittstellen kommen. Die hierbei entstehenden Netzwerk-Latenzen können die Effizienz und Performance der Applikationen beeinträchtigen.

Um die Effizienz von Rechenzentren zu steigern, haben Piao et al. [PY10], Mann et al. [MGD⁺12], Dalvandi et al. [DGC13] und Pu et.al [PLM⁺13] eine VM-Platzierung vorgeschlagen, die die Netzwerkkommunikation der virtuellen Server untereinander berücksichtigt. Das Ziel ist eine Performance-Steigerung der Applikationen durch eine optimierte Netzwerkkommunikation. In Simulationen konnten Optimierungen bezüglich der Bandbreite von 80-100% erreicht werden. Weitere Server-Ressourcen oder Energie-Aspekte wurden in diesen Ansätzen nicht berücksichtigt.

3.8. Verwendung von Prognosen zur Kapazitätsplanung

Eine weiterer Ansatz mit veränderlicher Energieverfügbarkeit umzugehen, ist die Verwendung von Vorhersagen zur Planung von Serverressourcen wie in [CGJ⁺10] vorgeschlagen. Auf der Basis von historisierten Daten werden Prognosen für einen zukünftig zu erwartenden Ressourcenbedarf erstellt.

Entsprechend des prognostizierten Ressourcenbedarfs werden Allokationen erstellt, die korrespondierend zum zeitlichen Verlauf eine Art betrieblichen Fahrplan darstellen.

Dieser Ansatz einer Ressourcenplanung wird sowohl mit Hilfe von VM-Allokationen und Virtualisierungstechnologie, als auch mit Scheduling-Verfahren umgesetzt.

3.9. Übersicht zu aktuell verfügbaren Verfahren

In Tabelle 3.1 sind die relevanten Ansätze noch einmal dargestellt und auf Basis der verwendeten Methoden zu Clustern zusammengefasst.

Aus dem Umfeld des beschriebenen Nutzungsszenarios und den Forschungsfragen ergeben sich Kriterien, die ein Verfahren erfüllen muss, damit es den Anwendungsfall abdecken kann:

- *Leistungsaufnahme beeinflussbar*: Als Grundvoraussetzung muss das Verfahren eine Beeinflussung der Leistungsaufnahme sowohl positiv als auch negativ umsetzen können.
- *agnostisch*: In Abschnitt 1.2 sind die Betriebsmodelle für Rechenzentrumsdienstleistungen beschrieben. Insbesondere beim Modell des IaaS besteht kein Zugriff auf die Applikationen seitens des Dienstleisters. Applikationsinvasive Methoden sind somit nicht anwendbar. Aus Applikationssicht wird daher ein agnostisches Verfahren benötigt, sodass die betriebenen Applikationen keine Kenntnis über das Verfahren erlangen oder gar eine Kooperation nötig ist. Ein agnostisches Verfahren kann im Bereich des IaaS als auch SaaS Anwendung finden.
- *lokal einsetzbar*: In der Reihe von aktuellen Verfahren werden zur Optimierung auch Techniken eingesetzt, die eine Verschiebung von Rechenleistung vorsehen, um einen lokalen Effekt zu erzeugen. Insgesamt betrachtet, wird die Leistungsaufnahme jedoch nicht geändert, sondern lediglich in ein anderes Rechenzentrum verlagert. Bei nur einem Rechenzentrumsstandort oder geringer geographischer Ausbreitung sind diese Ansätze nicht verwendbar.
- *Effizienz im Fokus*: Bei der Implementierung der vorgestellten Verfahren werden diverse Faktoren betrachtet, die jeweils im Fokus stehen. Häufig steht alleinig die Leistungsaufnahme im Fokus. Insbesondere sollte es durch die Anwendung eines Verfahrens jedoch nicht zu Einbußen bezüglich der Effizienz kommen. Die Ansätze von Krioukov et. al [KGA⁺11] und Chen et. al [CCC14] verfolgen den Ansatz mit

3. Stand der Technik

Hilfe eines Task-Schedulers die Leistungsaufnahme zu beeinflussen, dabei werden auch Tasks abgebrochen. Hier kommt es beispielsweise zu Effizienz-Verlusten, da die investierte Rechenzeit zu keinem Ergebnis bzw. nicht zum Abschluss dieser Tasks führt. Unter dem Kriterium *Effizienz im Fokus* wird daher die Berücksichtigung von Effizienzgesichtspunkten bewertet.

Die Auswertung der genannten Kriterien zeigen die jeweiligen Spalten der Tabelle 3.1. Keines der Verfahren, die repräsentativ für die jeweils verwendete Methode aufgeführt sind, erfüllt alle gestellten Anforderungen.

Kurz-Beschreibung des Ansatzes	Effizienz im Fokus	lokal einsetzbar	Leistungsaufnahme beeinflussbar	agnostisch	Quellen
EDF-Schedule, Tasks werden entsprechend Energieverfügbarkeit gestartet oder beendet	✗	✓	✓	✗	[KGA ⁺ 11], [CCC14]
Effizienzsteigerung der Klimatisierung. Taskverteilung anhand Servertemperatur	✓	✓	✗	✗	[TGV08], [BF07], [MCRS05], [TGSC06], [TGV07], [XFL13], [LCB ⁺ 12]
Effizienzsteigerung durch optimierte Ressourcennutzung. Migration von VMs	✓	✓	✗	✓	[DGC13], [PY10], [WTLS ⁺ 09], [CFF12], [MGD ⁺ 12], [PLM ⁺ 13]
Geografische Verschiebung. Rechenlast wird an Standorte mit höherer Energieverfügbarkeit verlagert. Migration von VMs	✗	✗	nur lokaler Effekt (✓)	✓	[LLW ⁺ 11], [ZWW11], [MDKS12]

Tabelle 3.1.: Vergleich aktuell verfügbarer Verfahren

3.10. Zusammenfassung

Im Bereich der Leistungsaufnahme von Servern und Rechenzentren gibt es bereits eine Reihe von Ansätzen, die sich mit der Optimierung und Steuerung der Leistungsaufnahme beschäftigen. Die vorgestellten Verfahren bedienen sich unterschiedlicher Techniken zur Beeinflussung der Leistungsaufnahme, die in diesem Abschnitt vorgestellt wurden und eine große Bandbreite verschiedener Herangehensweisen aufzeigen.

Verfahren, die auf der Basis von Servervirtualisierung arbeiten, ermöglichen ein Verschieben von Applikationen inklusive der durch den Ressourcenbedarf hervorgerufenen Serverauslastung. Ein Verfahren, welches auf akute Anforderungen zur Anpassungen der Gesamt-Leistungsaufnahme reagieren kann und gleichzeitig ohne direkten Zugriff auf die Applikationen agieren kann, fehlt in dieser Reihe jedoch.

4. Problemstellung und Abgrenzung

Ziel dieser Arbeit ist die Bereitstellung einer Methodik, die es ermöglicht, die Leistungsaufnahme eines Rechenzentrums im laufenden Betrieb zu ändern und die eingesetzten Serverressourcen effizient zu nutzen. Die genaue Beschreibung der Anforderung und Abgrenzung des Problems erfolgt in diesem Abschnitt.

In Anlehnung an die in Abschnitt 1.3 vorgestellten Forschungsfragen soll eine hinreichende Lastaufnahmevarianz des Rechenzentrums ohne applikations-invasive Methoden erreicht werden, um auf die raschen Wechsel der Energieverfügbarkeit aus erneuerbaren Energieträgern wie Wind- und Solarenergie reagieren zu können. Weiterhin können Anforderungen eines Strom-Netzbetreibers eine Rolle spielen, wobei die adaptive Leistungsaufnahme des Rechenzentrums als Netz-Stabilisierungsinstrument dienen kann. Bei der Implementierung des Verfahrens soll dessen Effizienz ebenfalls Beachtung finden.

Das Ressourcenmanagement eines Rechenzentrums hat die Aufgabe, die betriebenen IT-Dienste derart auf die vorhandenen Server zu verteilen, dass die mit dem Kunden vereinbarten SLAs eingehalten werden. Mit dieser Forderung sind die Belange des Kunden bereits hinreichend beschrieben. Für einen ökonomischen Rechenzentrumsbetrieb sind jedoch weitere Aspekte zu berücksichtigen. Durch den zunehmenden Einsatz erneuerbarer Energien ist zukünftig mit Strompreismodellen zu rechnen, die sich an der Energieverfügbarkeit orientieren. Darüber hinaus rückt auch die Frage der CO₂-Bilanz von IT-Services mehr in den Fokus, wie in Abschnitt 1.1 beschrieben. Um auf aktuelle Verfügbarkeiten von elektrischer Energie reagieren zu können, wird eine passende Methodik für Rechenzentren gesucht. In Abschnitt 4.1 wird beziehend auf die Betriebsmodelle von Rechenzentrumsdienstleistungen zunächst der Rahmen für technische Einflussmöglichkeiten abgesteckt, aus denen sich der Technologie-Ansatz für das zu implementierende Verfahren ableitet. Die genaue Abgrenzung hierzu erfolgt in Abschnitt 4.2 und 4.3.

Die Komplexität der Aufgabenstellung wird in Abschnitt 4.4 beleuchtet und eine Analogie zu bekannten Problemen und zugehöriger Komplexitätsklasse hergestellt. In Abschnitt 4.5 wird die Aufgabenstellung in Teilschritte zerlegt und eine formale Darstellung des Lösungsraums aufgezeigt.

Neben den offenkundigen Anforderungen werden in Abschnitt 4.6 auch die nicht-funktionalen Anforderungen beschrieben, die ein entsprechendes Verfahren abdecken muss. Im Anschluss erfolgt eine Zusammenfassung dieses Kapitels.

4.1. Technologieansatz

In Abschnitt 1.2 sind die verschiedenen Betriebsmodelle für Rechenzentren vorgestellt worden. Im IaaS- und Cloudservice-Bereich hat der Dienstleister keinen Zugriff auf die betriebenen Applikationen, somit ergibt sich die technologische Prämisse, ein Verfahren zu entwickeln, welches ohne applikationsinvasive Methoden arbeitet.

Bei dem beschriebenen Betriebsmodell hat der Dienstleister keinen Zugriff auf die Applikationen und kann an ihnen somit auch keine Manipulationen vornehmen. Es besteht folglich auch keine Notwendigkeit, die auf den Servern laufenden Applikationen fachlich und inhaltlich kennen oder berücksichtigen zu müssen, wie es bei den Verfahren auf Basis von EDF-Scheduling der Fall ist. Dieser administrative Aufwand entfällt. Es sollen explizit keine (fachlichen) Eingriffe an einer Applikation vorgenommen werden, sodass die Applikationen bezüglich des Verfahrens agnostisch sind.

Eine Einflussnahme auf Applikationen ohne direkten Zugriff kann beispielsweise über die der Applikation zur Verfügung stehenden Ressourcen umgesetzt werden. Steht innerhalb einer Applikationstransaktion beispielsweise eine Schreiboperation an, bei der eine Datei auf der Festplatte abgelegt werden soll, so wird die Festplatte als Ressource angefordert. Sobald die Festplatte zur Verfügung steht, wird die Schreiboperation ausgeführt. Hierbei ist das Warten auf die Zuteilung der Festplatte ein ganz gewöhnlicher Vorgang, mit dem eine Applikation umzugehen weiß. Das Warten auf eine Ressource kann also als Mittel zur Einflussnahme dienen, wobei es für eine Applikation nicht feststellbar ist, ob die Wartesituation zufällig, also durch die vorhergehende Benutzung der Ressource von einer anderen Applikation entstanden ist, oder bewusst herbeigeführt wurde. Unter der Verwendung derartiger Methoden kann ein agnostisches Verfahren entwickelt werden.

Um die eingangs erwähnten Anforderungen abdecken zu können, soll das Lastmanagement eines Rechenzentrums genauer betrachtet werden, insbesondere die Verteilung der virtuellen Maschinen auf die physikalischen Server. Mit Hilfe von Virtualisierungstechnologien können virtuelle Maschinen online migriert (wie in Abschnitt 2.4.2 beschrieben) und so eine Verlagerung der Ressourcennutzung bewirkt werden.

Das gesuchte Verfahren soll auf Basis einer gegebenen Ausgangssituation (Belegung von virtuellen Maschinen auf physikalischen Servern) eine passende Belegung finden und mit der Anforderung einer Änderung der Leistungsaufnahme konfrontiert werden können. Es wird unter Berücksichtigung der aktuellen Verteilung der virtuellen Maschinen eine Permutation gesucht, die die gewünschte Leistungsaufnahme des Rechenzentrums hervorruft und gleichzeitig eine effiziente Ressourcennutzung gewährleistet.

Bei den bestehenden Verfahren [CFF12] wird die auf den Servern durch Applikationen evozierte Auslastung der Serverressourcen (IT-Last) als Vorgabe hingenommen und eine optimale Konstellation für diese vorherrschende IT-Last gesucht. Die Auslastung der Server wird hier nun nicht als fixe, naturgegebene Größe verstanden und soll in den jeweiligen Zielkonstellationen der Verteilung von virtuellen Maschinen bewusst geändert werden. Es soll explizit die Konkurrenz um Serverressourcen zum Beeinflussen der Serverauslastung und damit der Leistungsaufnahme des gesamten Rechenzentrums herangezogen

werden. Untersuchungen, wie sie in [BS16] durchgeführt wurden, zeigen mögliche Effekte auf. Durch geschicktes Verteilen von virtuellen Maschinen auf physikalischen Servern soll die Leistungsaufnahme des Rechenzentrums in hinreichendem Umfang beeinflussbar gemacht werden, um die Leistungsaufnahme den Anforderungen entsprechend zu erhöhen oder zu senken.

Die auf den virtuellen Maschinen laufenden Applikationen benötigen zur Abarbeitung ihrer Aufgaben Systemressourcen. Stehen die benötigten Systemressourcen zur Verfügung, kann die Applikation optimal arbeiten. Aus diesem Grund konsolidieren konventionelle Algorithmen zur Platzierung von virtuellen Maschinen gerade jene auf einem physikalischen Server, die nicht um dieselben Systemressourcen konkurrieren. Werden nun alle durch den physikalischen Server bereitgestellten Systemressourcen stark belastet, ist auch die Energieaufnahme entsprechend hoch. Der größte Einfluss auf die Leistungsaufnahme obliegt hier der CPU-Auslastung, wie Chen et al. [CHL⁺08] gezeigt haben.

Um nun die Leistungsaufnahme eines Rechenzentrums zu verringern, soll eine Konstellation gefunden werden, in der weniger Energie aufgenommen wird. Es soll eine Absenkung der Applikationsleistung erreicht werden, wobei die vereinbarten SLAs jedoch gehalten werden. In dieser Konstellation könnte beispielsweise eine Konkurrenz um CPU, RAM oder Festplatten-I/O bestehen, die bewusst herbeigeführt wird, um den Applikationsdurchsatz zu verringern und in der Folge die Leistungsaufnahme der Server zu reduzieren.

Neben den Hardware-Ressourcen kommt ein weiterer Aspekt zur Einflussnahme in Betracht, die Netzwerkkommunikation. In diversen Ansätzen [MLP⁺13, PY10] wurde die Netzwerkkommunikation in Rechenzentren untersucht.

Der Ansatz von Mann et al. [MGD⁺12] sieht dabei vor, VMs so zu platzieren, dass die Netzwerkkommunikation entzerrt wird. Ein Großteil von Applikationen, die in einem Rechenzentrum betrieben werden, besteht aus mehreren Komponenten, zwischen denen ein intensiver Datenaustausch stattfindet. Selbst einfache Webapplikationen bestehen aus einem Applikationsserver und einer zugehörigen Datenbank, aus der Informationen abgerufen und persistiert werden. Auch hier könnte eine Warte-Situation durch verlangsamte Kommunikation provoziert werden, die wiederum dazu führt, dass die Leistungsaufnahme des betreffenden Servers sinkt, da die Applikation auf eine Antwort ihres Nachbarsystems oder einer Komponente, die auf einem anderen Server betrieben wird, wartet.

Indem aus Applikationssicht eine mehr oder weniger ressourcen-optimale Verteilung der virtuellen Maschinen auf die physikalischen Server vorgenommen wird, soll die Leistungsaufnahme des Rechenzentrums beeinflusst werden. Für die Verteilung der virtuellen Maschinen soll genau jene Permutation gefunden werden, die auf der Basis von Auslastungseffekten der Server-Ressourcen den energetischen Anforderungen entspricht. Es gilt, eben jene möglichen Zielkonstellationen zu eruieren.

Maßgebliche Faktoren, die über die Zielkonstellation entscheiden, sind zum einen die nötige Zeitspanne zum Erreichen der gewünschten Konstellation (Permutationsdauer) und zum anderen Faktoren, die sich aus dem Ressourcenbedarf der einzelnen virtuellen Maschinen ergeben. Ein Rechenzentrum von einem in einen anderen Zustand zu überführen, kann eine gewisse Zeit in Anspruch nehmen. Nach der Entscheidung für eine neue Konstellation gibt es eine Latenz, bis der neue Zustand erreicht ist. Diese Latenz ergibt sich

4. Problemstellung und Abgrenzung

aus der Dauer, die für die Migration der virtuellen Maschinen benötigt wird und aus den System-Zuständen der beteiligten Komponenten. Die Klimatisierung reagiert beispielsweise verzögert auf eine Veränderung der Wärmeabgabe durch die Server.

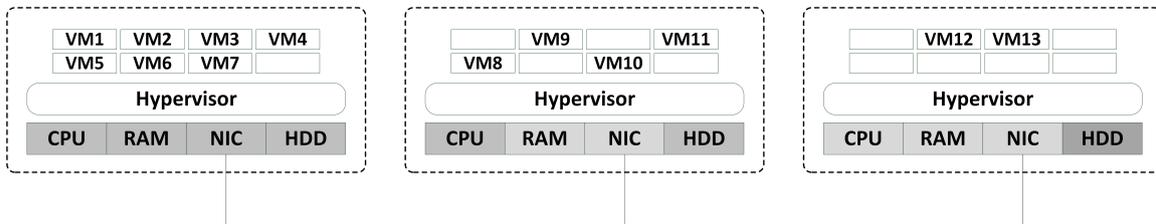


Abbildung 4.1.: Ausgangssituation einer VM-Verteilung auf physikalische Server mit Hypervisor

In Abbildung 4.1 sind schematisch drei physikalische Server mit den auf ihnen aktuell betriebenen virtuellen Maschinen (VM1, ..., VM13) dargestellt. Die einzelnen VMs haben einen gewissen Ressourcenbedarf, der von dem physikalischen Server auf dem sie sich befinden, bereitgestellt wird. Die Auslastung der Ressourcen wird durch die Einfärbung der zugehörigen Komponente dargestellt. Eine dunkle Färbung bedeutet eine hohe Auslastung. In dieser gegebenen Konstellation ergibt sich ein Energiebedarf des Rechenzentrums, der beeinflusst werden soll.

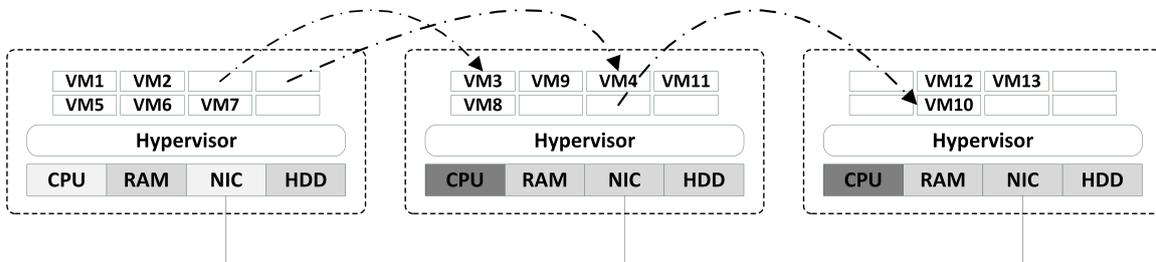


Abbildung 4.2.: Durch Permutationen von VMs geänderte Ressourcenauslastung

Die Abbildung 4.2 zeigt nun die durch Permutation der VMs erreichte Zielkonstellation, die den Energiebedarf in gewünschtem Umfang im Vergleich zur Ausgangssituation in Abbildung 4.1 adaptiert. Konkret wurden die virtuellen Maschinen VM3, VM4 und VM10 migriert und eine veränderte Ressourcennutzung herbei geführt, die die geforderte Energieaufnahme des Rechenzentrums liefert. Deutlich wird die geänderte Ressourcennutzung durch die Einfärbung der jeweiligen Server-Komponente.

Die gegebene Verteilung der virtuellen Maschinen wird durch Permutation geändert und die Ressourcennutzung variiert. Somit kann die Leistungsaufnahme des Rechenzentrums kontrolliert beeinflusst werden. Der zu implementierende Algorithmus soll auf diesem Prinzip aufbauen und erhält als Eingabe eine feste Menge virtueller Maschinen, verteilt auf eine feste Menge physikalischer Server. Zudem wird eine bestimmte Leistungsaufnahme gefordert.

Diese Arbeit stützt sich auf aktuelle Virtualisierungstechniken und soll die Potentiale weiterer Freiheitsgrade untersuchen und darlegen. Rechenzentren, die Lastmanagementverfahren einsetzen und beispielsweise mit Hilfe von Virtualisierungslösungen nicht-genutzte Server abschalten können, haben bereits einen energetischen Handlungsspielraum, der hier als Freiheitsgrad bezeichnet wird. Durch die Kombination mit weiteren Verfahren soll dieser Freiheitsgrad weiter erhöht und eine verbesserte Adaption an energetische Anforderungen möglich werden.

4.1.1. Prozessor-Ressourcen als Steuerungsinstrument

Wie in Abschnitt 2 beschrieben, existieren DVFS-Technologien, die eine Anpassung der Leistungsaufnahme an die Auslastung eines Servers erlauben. Diese Technologie ist für das Absenken und Erhöhen der Leistungsaufnahme von Bedeutung. In dieser Arbeit soll in Kombination mit DVFS das Prinzip der Überprovisionierung genutzt werden. Als Überprovisionierung wird hier eine Auslastung einer Ressource verstanden, die über 100% hinausgeht. Zur Überprüfung der Theorie, dass sich die CPU ohne Effizienz-Verlust überprovisionieren lässt, wurden Experimente durchgeführt. Es wurde eine Muster-VM mit CPU Benchmark aufgesetzt und weitere Klone dieser VM hergestellt. In einer Folge von Experimenten wurde die Anzahl parallel betriebener VMs sukzessive erhöht. Der verwendete physikalische Server mit Hypervisor ist bei allen Iterationen derselbe. In Abbildung 4.3 sind die Ergebnisse dieser Experimente dargestellt.

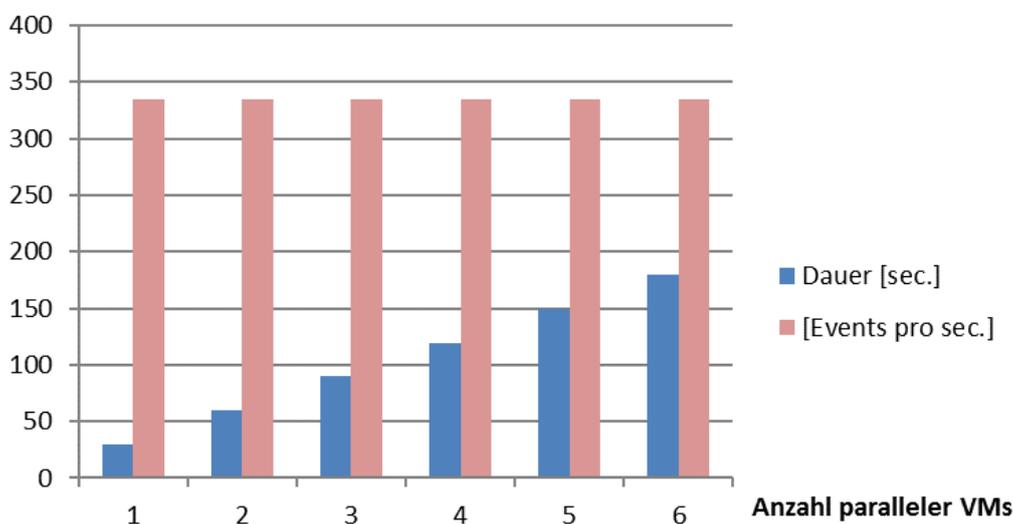


Abbildung 4.3.: Performance-Ergebnisse aus Experimenten mit Überprovisionierung der CPU-Ressourcen

Die Ergebnisse zeigen eine mit der Anzahl an VMs linear wachsende Ausführungszeit, die Performance je VM verringert sich entsprechend. Darüber hinaus bleibt die Anzahl Events

4. Problemstellung und Abgrenzung

pro Sekunde, die verarbeitet wurden, jedoch konstant. Durch den Betrieb der weiteren VMs und Überprovisionierung der CPU ist somit kein Effizienz-Verlust erkennbar, wie in Forschungsfrage Nr. 5 gefordert. Erwähnenswert ist auch die Tatsache, dass es zumindest in diesem Szenario keinen feststellbaren Overhead durch die zusätzlichen VMs zu geben scheint. In weiteren Experimenten wurde die Konkurrenz um die CPU in mehreren Ebenen untersucht. Den Ergebnissen ist zu entnehmen, dass sich eine lineare Gleichverteilung der CPU-Ressource auf die jeweiligen VMs ergibt. Dies ist sowohl bei der Verteilung durch die VMs als auch durch konkurrierende Prozesse zu beobachten, sofern kein explizites Ressourcenmanagement aktiviert wurde.

Mit Hilfe der Servervirtualisierung (s. Abschnitt 2.4) können VMs auf wenige PMs konsolidiert werden. Es kommt zu einer CPU-Überprovisionierung auf der Ziel-PM und die Auslastung der abgebenden PM sinkt. Die Leistungsaufnahme eines bereits zu 100% ausgelasteten Servers kann durch die Aufnahme weiterer VMs nicht mehr erhöht werden. Mittels DVFS-Technologie (s. Abschnitt 2.3.1) adaptiert sich die Leistungsaufnahme der abgebenden PM an die nun reduzierte Auslastung, die Leistungsaufnahme sinkt folglich. Im umgekehrten Fall erhöht sich die Leistungsaufnahme entsprechend wieder. Mit der CPU-Überprovisionierung ist damit eine passende Technik für die Steuerung der Leistungsaufnahme gefunden und beantwortet Forschungsfrage Nr. 2. Diese Technik ist mit nur einem Rechenzentrumsstandort realisierbar, wie in Forschungsfrage Nr. 1 gefordert, und kann im Betriebskontext von IaaS und SaaS Anwendung finden.

Die Simulation und das Verhalten von Multiprozessorsystemen ist ein weites Forschungsfeld und soll nicht Gegenstand dieser Arbeit sein. Im Kontext dieser Arbeit ist vorrangig die Leistungsaufnahme eines Systems unter einer bestimmten Auslastung von Bedeutung. Für die Verwendung dieser Daten wird daher auf reale Daten von Servern zurückgegriffen, die von der Standard Performance Evaluation Corporation (SPEC) zur Verfügung gestellt werden. Die Daten werden mit Hilfe von Benchmarks ermittelt und sind für diverse Server-Systeme verfügbar [Sta].

4.2. Abgrenzung

Kernaspekt einer Integration des Rechenzentrums in ein Smart-Grid ist die Fähigkeit, den eigenen Energiezustand an die jeweilige Situation im Grid anzupassen. Dabei besitzt jedes Rechenzentrum bereits bestimmte Freiheitsgrade, in denen es energetisch regelbar ist. Werden beispielsweise Server ein- oder ausgeschaltet, ändert sich die Leistungsaufnahme des gesamten Rechenzentrums. Sind solche Aktionen planbar und grundsätzlich ausführbar, ist dadurch ein energetischer Freiheitsgrad beschrieben.

Ein Lösungsansatz zur Beeinflussung der Leistungsaufnahme mit überregionalem Fokus, wie in [CHT12] beschrieben, soll explizit ausgeschlossen werden. Ebenso kommt eine dynamische Standortverlagerung, wie in [QWB⁺09] vorgeschlagen, nicht in Betracht. Diese Ansätze sind nur in größeren Szenarien mit Rechenzentren an mehreren Standorten realisierbar, so würde der mögliche Einsatzbereich des Verfahrens stark reduziert. Darüber

hinaus wird der Energieumsatz der Server in Summe nicht verändert, es erfolgt lediglich eine Verlagerung der Leistungsaufnahme.

Durch den Einsatz der Server-Virtualisierung kann der energetische Freiheitsgrad deutlich erweitert werden, denn sie erlaubt die Ausführung von unterschiedlichen Anwendungskombinationen auf verschiedenen Servern. Eben dieser Aspekt und deren Auswirkungen auf Effizienz und Leistungsaufnahme des Rechenzentrums werden in dieser Arbeit beleuchtet.

Auf diesen Effekten aufbauend, wird ein schlankes Verfahren zur Regelung der Leistungsaufnahme von Servern entwickelt. Dabei wirkt sich eine geänderte Serverauslastung und das Ausschalten von Servern auch auf die Leistungsaufnahme von weiteren Komponenten im Rechenzentrum, wie beispielsweise die von Kühlgeräten oder Stromversorgungssystemen aus. Diese sekundären Aspekte und deren komplexe Zusammenhänge werden in dieser Arbeit jedoch nicht näher betrachtet.

4.3. Effizienz und Skalierungseffekte

Wie in Abschnitt 2.6.1 beschrieben, haben Server ganz unterschiedliche Charakteristika bezüglich ihrer Effizienz. Je nach eingesetztem Typ findet eine Skalierung der Leistungsaufnahme statt und es gibt eine Korrelation zwischen Auslastung und Leistungsaufnahme. Als Grundlage dieser Arbeit wird von einem Servertyp ausgegangen, der bezüglich der Leistungsaufnahme skaliert und ein entsprechendes Verhalten aufweist.

In Abschnitt 3 dieser Arbeit sind neben den grundsätzlichen Methodiken zur Leistungsanpassung von Servern auch eine ganze Reihe Nebeneffekte angesprochen worden. Ein weiterer Punkt von Relevanz ist das Thema der Klimatisierung. Es wurde insbesondere die Problematik der Entstehung von Hotspots angesprochen. Treten Hotspots auf, kommt es zu nennenswertem Rückgang der Effizienz bei den Klimatisierungssystemen. Dieser beschriebene Aspekt soll ebenfalls beim Design des Verfahrens Berücksichtigung finden.

In Abschnitt 2 wurden unter anderem Verfahren beschrieben, die Tasks abbrechen, was folglich zu einem Effizienzverlust führt. Derartiges Vorgehen soll vermieden und insgesamt ein Verfahren gefunden werden, das mit möglichst wenig Rechenoperationen auskommt.

4.4. Komplexität der Aufgabenstellung

Zu der hier gestellten Aufgabe des Verteilens von VMs auf PMs lässt sich eine Analogie zum sogenannten Rucksackproblem oder Behälterproblem aus dem Bereich der Kombinatorik aufbauen [HMGW07]. Beim Behälterproblem besteht die Aufgabe darin, eine gewisse Anzahl von Objekten unterschiedlicher Größe in möglichst wenige Behälter zu verstauen. Die Behälter sollen effizient genutzt werden, ohne Platz zu verschwenden.

Ähnlich verhält es sich bei der Verteilung der VMs auf die PMs, wobei hier die VMs die Objekte und die PMs die Behälter darstellen. Die VMs müssen mit ihrem Ressourcenbedarf von den PMs aufgenommen werden können, der Ressourcenbedarf der VMs

4. Problemstellung und Abgrenzung

ist damit analog zur Größe der Objekte zu betrachten. Eine VM muss bezüglich ihres Ressourcenbedarfs zum Betrieb auf einer bestimmten PM geeignet sein. Dies ist der Fall, wenn die benötigten Ressourcen von der PM zur Verfügung gestellt werden können. Da sich der gesamte Ressourcenbedarf einer virtuellen Maschine aus mehreren Komponenten zusammensetzt und jede Ressource einzeln zu bewerten ist, kann man hier von einem mehrdimensionalen Behälterproblem sprechen. Hinzu kommt, dass die VMs bereits in Betrieb sind und in Fortführung der Analogie die Rucksäcke bereits gefüllt sind.

Bereits das einfache Behälterproblem gehört zur Klasse der NP-schweren Probleme. Eine Aufgabenstellung der hier beschriebenen Komplexität lässt sich folglich nicht in angemessener Zeit lösen. Anstelle einer absoluten Berechnung müssen Heuristiken und gegebenenfalls weitere Methoden zur Lösung herangezogen werden.

4.5. Beschreibung des Problems einer Leistungsänderung in Teil-Schritten

Wie bereits beschrieben, gibt es diverse Effekte, die die Leistungsaufnahme eines Rechenzentrums beeinflussen. Primärer Ausgangspunkt ist die Leistungsaufnahme der Server und deren Abwärme, an die sich alle weiteren Support-Systeme adaptieren. Daher konzentriert sich dieser Ansatz auf die Leistungsaufnahme der Server.

4.5.1. Ziel-Leistungsaufnahme

Die Leistungsaufnahme der Server-Systeme hängt von deren Auslastung ab und wird somit durch den Ressourcenbedarf der betriebenen VMs bestimmt. Der Lösungsraum, um eine vorgegebene Leistungsaufnahme der Server zu erreichen, wird folglich über die Kombinationsmöglichkeiten von PMs und VMs definiert.

Gegeben sei ein Rechenzentrum mit einer festen Anzahl k an physikalisch vorhandenen Servern (PMs) und der Anzahl n hierauf betriebenen virtuellen Servern (VMs). Die Zuordnung, die beschreibt, wie die VMs auf die PMs verteilt sind, wird im Folgenden als Allokation \mathcal{A} bezeichnet. Insbesondere unterscheiden sich Allokationen durch die Zuweisung von VMs und PMs, wobei die Gesamtzahl der VMs und PMs jedoch gleich bleibt.

Die Leistungsaufnahme der in einem Rechenzentrum befindlichen Server unter einer Allokation \mathcal{A} ist definiert durch:

$$P_{Server}(\mathcal{A}) = \sum_{i=1}^j P_{PM_i}(\mathcal{A}), \quad (4.1)$$

wobei $P_{PM_i}(\mathcal{A})$ die Leistungsaufnahme des i -ten Servers unter der Allokation \mathcal{A} ist und insgesamt j Server betrieben werden.

4.5. Beschreibung des Problems einer Leistungsänderung in Teil-Schritten

Zu einer gewünschten Leistungsaufnahme P_{target} wird daher eine Allokation \mathcal{A} für $\epsilon \rightarrow 0$ gesucht:

$$\left| \sum_{i=1}^j P_{PMi}(\mathcal{A}) - P_{target} \right| + \epsilon = 0. \quad (4.2)$$

4.5.2. Teilschritte

Die zu lösende Aufgabe lässt sich in mehrere Teilschritte gliedern und wird in den folgenden Punkten beschrieben:

- Zunächst ist die Menge aller möglichen Zuordnungen von VMs zu PMs zu bilden. Für ein Rechenzentrum mit einer festen Anzahl k an physikalisch vorhandenen Servern und der Anzahl n hierauf betriebenen virtuellen Server kann eine Allokation als eindeutige Abbildung beschrieben werden:

$$\mathcal{A} : n \mapsto k, n \in VMs; k \in PMs. \quad (4.3)$$

Die Anzahl der möglichen Allokationen, die ein Rechenzentrum annehmen kann, entspricht der Anzahl der Möglichkeiten, eine n -elementige Menge in k Partitionen zu teilen, wobei die entstehenden k Mengen disjunkt und nichtleer sind.

Diese Bedingungen werden durch die Stirling-Zahl zweiter Art $\mathcal{S}_{n,k}$ beschrieben:

$$\mathcal{S}_{n,k} = \frac{1}{k!} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} j^n. \quad (4.4)$$

Für ein beispielhaftes Szenario mit 10 virtuellen Servern und drei physikalisch vorhandenen Servern ergeben sich somit $\mathcal{S}_{10,3} = 9330$ mögliche Allokationen. Bei 40 virtuellen Servern und 8 physikalisch vorhandenen Servern ergeben sich bereits $\mathcal{S}_{40,8} = 3,17 * 10^{31}$ mögliche Allokationen. Wird darüber hinaus die Option betrachtet, Server auch komplett abzuschalten, erhöht sich die Anzahl der Möglichkeiten weiter auf:

$$\sum_{i=1}^k \mathcal{S}_{n,k-i} \quad (4.5)$$

- In einem weiteren Schritt ist die Menge der Allokationen auf ihre Erreichbarkeit hin zu untersuchen. Da zum Zeitpunkt einer eintreffenden Anforderung bereits eine Allokation vorliegt (die VMs sind in Betrieb), ist zu prüfen, welche Allokationen durch iterative Migrationen oder Permutationen von VMs erreichbar sind. Bei der Herstellung einer initialen Allokation besteht diese Einschränkung nicht.

4. Problemstellung und Abgrenzung

Gesucht wird die Teilmenge $A_R \subsetneq A$ aller Allokationen, die von der Ausgangsallokation \mathcal{A}_0 aus erreichbar sind. Es soll die Allokation \mathcal{A}_0 mit einer endlichen Anzahl von n VM-Migrationen in die Allokation \mathcal{A}_i abgebildet werden. Hierfür wird eine Migrationsfunktion benötigt.

Die Funktion $f_{\hat{v} \rightarrow \hat{m}}$ bildet eine Allokation \mathcal{A} auf eine Allokation \mathcal{A}' durch die Migration der VM \hat{v} auf PM \hat{m} ab.

Die Spezifikation von Ressourcen sei ein 4-Tupel $RS = (R_{CPU}, R_{RAM}, R_{HDD}, R_{NIC})$, wobei R_{CPU} die Ressource CPU, R_{RAM} die Ressource RAM, R_{HDD} die Ressource HDD und R_{NIC} die Ressource NIC spezifiziert. Die Menge aller Spezifikationen $\mathbb{R}\$$ ist damit $\mathbb{R}\$ \subsetneq \mathbb{R}^4$.

Sei $R_{VM} \in \mathbb{R}\$$ der Vektor des Ressourcenbedarfs einer Menge von n VMs und $R_C \in \mathbb{R}\k der Vektor der Ressourcen-Kapazitäten von k PMs. Die Migration einer VM \hat{v} auf eine PM \hat{m} wird durch eine Migrationsfunktion $f_{\hat{v} \rightarrow \hat{m}}$ beschrieben:

$$f_{\hat{v} \rightarrow \hat{m}}: \begin{cases} A \times \mathbb{R}^4 \rightarrow A \times \mathbb{R}^4 \\ (\mathcal{A}, U) \rightarrow (\mathcal{A}', U') \end{cases} \quad (4.6)$$

wobei sich durch die Migration der VM \hat{v} auf PM \hat{m} die Allokation \mathcal{A} zu \mathcal{A}' ändert.

$$\mathcal{A}'(v): \begin{cases} \mathcal{A}(v) & \text{falls } v \neq \hat{v}, 1 \leq v \leq n \\ \hat{m} & \text{falls } v = \hat{v} \end{cases} \quad (4.7)$$

Für die Ressourcenauslastung der PMs

$$U = (R_{CPU_1}, R_{RAM_1}, R_{HDD_1}, R_{NIC_1}), \dots (R_{CPU_k}, R_{RAM_k}, R_{HDD_k}, R_{NIC_k}) \quad (4.8)$$

vor der Migration und

$$U' = (R'_{CPU_1}, R'_{RAM_1}, R'_{HDD_1}, R'_{NIC_1}), \dots (R'_{CPU_k}, R'_{RAM_k}, R'_{HDD_k}, R'_{NIC_k}) \quad (4.9)$$

nach der Migration mit dem vorausgesetzten Ressourcenbedarf der n VMs

$$R_{VM} = (\widetilde{R_{CPU_1}}, \widetilde{R_{RAM_1}}, \widetilde{R_{HDD_1}}, \widetilde{R_{NIC_1}}), \dots (\widetilde{R_{CPU_n}}, \widetilde{R_{RAM_n}}, \widetilde{R_{HDD_n}}, \widetilde{R_{NIC_n}}) \quad (4.10)$$

gilt offensichtlich für alle nicht betroffenen PMs i , $1 \leq i \leq k$, $i \neq \mathcal{A}(\hat{v})$ und $i \neq \mathcal{A}'(\hat{v}) = \hat{m}$:

$$R_{CPU_i} = R'_{CPU_i}, R_{RAM_i} = R'_{RAM_i}, R_{HDD_i} = R'_{HDD_i}, R_{NIC_i} = R'_{NIC_i} \quad (4.11)$$

und für alle betroffenen PMs:

$$R'_{CPU_{\mathcal{A}(\hat{v})}} = R_{CPU_{\mathcal{A}(\hat{v})}} - \widetilde{R_{CPU_{\hat{v}}}} \quad (4.12)$$

4.5. Beschreibung des Problems einer Leistungsänderung in Teil-Schritten

und analog für die anderen Ressourcen.

Für die Ziel-PM $\hat{m} = \mathcal{A}'(\hat{v})$ gilt entsprechend:

$$R'_{CPU_{\hat{m}}} = R_{CPU_{\hat{m}}} + \widetilde{R_{CPU_{\hat{v}}}} \quad (4.13)$$

und analoge Bedingungen für die weiteren Ressourcen.

Der Vektor der PM-Kapazitäten R_C ist beschrieben durch:

$$R_C = (\widehat{R_{CPU_1}}, \widehat{R_{RAM_1}}, \widehat{R_{HDD_1}}, \widehat{R_{NIC_1}}), \dots, (\widehat{R_{CPU_k}}, \widehat{R_{RAM_k}}, \widehat{R_{HDD_k}}, \widehat{R_{NIC_k}}). \quad (4.14)$$

Eine Migration ist nur möglich, wenn die Ziel-PM ausreichend Ressourcen für den zusätzlichen Betrieb der VM bereitstellen kann oder eine Überprovisionierung vom Grad λ gestattet wird, wobei λ den Grad einer möglichen Überprovisionierung der CPU-Ressource beschreibt. Folglich gilt mit dem Vektor der PM-Kapazitäten:

$$\begin{aligned} R'_{CPU_i} &\leq \lambda * \widehat{R_{CPU_k}} \text{ und} \\ R'_{RAM_i} &\leq \widehat{R_{RAM_k}} \text{ und} \\ R'_{HDD_i} &\leq \widehat{R_{HDD_k}} \text{ und} \\ R'_{NIC_i} &\leq \widehat{R_{NIC_k}} \end{aligned} \quad (4.15)$$

für alle $1 \leq i \leq k$, wobei es genügt, den Fall der neuen Ziel-PM \hat{m} mit $i = \hat{m}$ zu betrachten.

Die Gesamtmenge der erreichbaren Allokationen kann nun bestimmt werden. Gesucht werden alle Allokationen \mathcal{A}_i für die gilt: Es existiert eine eindeutige Abbildung mit $n < \infty$ Migrationen:

$$A_R = \{\mathcal{A}_i \in A \mid \exists f_1, \dots, f_n : \mathcal{A}_0(f_n \circ \dots \circ f_1) = \mathcal{A}_i\} \quad (4.16)$$

Wobei f die Migrationsfunktion $f_{\hat{v} \rightarrow \hat{m}}$ zur Migration einer VM und \circ der bekannte Operator zur Anwendung einer Komposition für die Darstellung einer Migrationsfolge ist.

- Ist die Menge aller Allokationen auf die Menge A_R aller erreichbaren Allokationen reduziert worden, sind für die verbliebene Teilmenge nun jene Allokationen zu bestimmen, die alle gestellten Anforderungen bezüglich der Leistungsaufnahme erfüllen. Die Leistungsaufnahme, die eine Allokation hervorruft, muss sich in einem Korridor ϵ um die angeforderte Leistungsaufnahme befinden. Diese Anforderung reduziert die Teilmenge der erreichbaren Allokationen weiter.

$$A_P = \{\mathcal{A}_i \in A_R \mid P_{Server}(\mathcal{A}_i) \leq P_{target} \pm \epsilon\} \quad (4.17)$$

4. Problemstellung und Abgrenzung

- Die Teilmenge von Allokationen A_P , die erreichbar sind und die Leistungsaufnahmekriterien erfüllen, kann nun nach Effizienz-Gesichtspunkten sortiert werden. Gesucht wird die Allokation, die sich mit dem kleinsten Migrationsaufwand erreichen lässt. Hier ist die Anzahl bezüglich der VM-Migrationen am geringsten und ein Effekt auf die Leistungsaufnahme schneller sichtbar. Gesucht wird eine Allokation, zu der eine Folge von n Permutationen existiert, gleichzeitig gibt es jedoch keine Folge die kürzer ist.

$$\begin{aligned} \mathcal{A}_i \in A_P \mid \exists f_1, \dots, f_n : \mathcal{A}_0(f_n \circ, \dots, \circ f_1) = \mathcal{A}_i \wedge \\ \nexists f'_1, \dots, f'_{n-1} : \mathcal{A}_0(f'_{n-1} \circ, \dots, \circ f'_1) = \mathcal{A}_i \end{aligned} \quad (4.18)$$

- Ist die anzustrebende Allokation gefunden, sind die Migrationen zur Herstellung dieser Allokation durchzuführen.

4.6. Anforderungen

Aus dem angestrebten Nutzungsszenario leiten sich die Anforderungen an das Verfahren ab, die in diesem Abschnitt beschrieben werden.

4.6.1. Fachliche Anforderungen

Zur Teilnahme am Prozess der Regelleistung, wie in Abschnitt 2.8 beschrieben, und Nutzung von variablen Strompreis-Tarifen ergeben sich folgende fachliche Anforderungen:

- Eine erkennbare Reaktion auf die gestellte Anforderung zur Leistungsaufnahme muss schnell (je nach angebotener Regelenergie) in Form einer Änderung der Leistungsaufnahme erfolgen.
- Das Verfahren muss nicht-invasiv und agnostisch agieren können, um für das Betriebsmodell des IaaS einsetzbar zu sein.
- Das Optimierungsziel darf sich nicht über lokale Standortvorteile in einem großen Gesamtszenario definieren, es soll eine lokale Änderung der Server-Leistungsaufnahme ermöglicht werden.
- Für eine Anforderung des Absenkens oder Erhöehens muss eine Lösung geboten werden.

4.6.2. Nicht-funktionale Anforderungen

Neben den fachlichen Anforderungen, die die Funktionalität eines Verfahrens ausmachen, gibt es weitere, sogenannte nicht-funktionale Anforderungen. Diese Anforderungen sind

aus Sicht des Anwenders oftmals inhärent, beziehen sich zumeist auf qualitative Aspekte und verdienen daher besondere Aufmerksamkeit. Wie bereits Eingang erwähnt, soll das Verfahren in Rechenzentren Anwendung finden, wo Kunden besonderen Wert auf hohe Betriebsqualität legen. Das zu implementierende Verfahren soll daher folgende Punkte aus dem Bereich der nicht-funktionalen Anforderungen abdecken:

- *Skalierbarkeit*: Basistechnologie in Rechenzentren ist die Servervirtualisierung, die sich flexibel bei der Ressourcenverwaltung und dem Instanzieren neuer virtueller Server zeigt. Das Verfahren darf dieser Flexibilität daher nicht entgegenstehen und muss die Erweiterbarkeit und Skalierbarkeit unterstützen.
- *Zuverlässigkeit*: Der Ausfall einzelner Komponenten oder des Gesamtsystems darf die Betriebssicherheit nicht gefährden.
- *Stabilität*: Wurde eine Vorgabe für eine Leistungsaufnahme gegeben und erreicht, soll diese auch bei unvorhersehbaren Ereignissen gehalten werden. Das Verfahren soll die Leistungsaufnahme überwachen und gegebenenfalls weitere Schritte zur Stabilisierung einleiten.
- *Effizienz*: Mit Migrationen von VMs ist sparsam umzugehen. Die Effekte von Live-Migrationen fallen zwar gering aus, sind jedoch feststellbar, wie in [VBVB09] beschrieben.
- *Transparenz*: Wie bereits dargestellt, ist der Applikationsbetrieb und die Sicherstellung der Servicequalität ein sensibles Thema im Kontext des Rechenzentrumsbetriebs. Unsicherheiten und hohe Systemkomplexität können zu einem Vertrauensverlust auf Kundenseite führen. Die grundlegende Methodik soll daher möglichst trivial und insbesondere jede durchgeführte Transaktion transparent und nachvollziehbar sein.

4.7. Zusammenfassung

Für die Aufgabe einer Anpassung der Server-Leistungsaufnahme wurden die technischen Handlungsfelder diskutiert, die sich im Rahmen eines Betriebsmodells von IaaS ergeben. Basierend auf dem aktuellen Stand der Technik aus Abschnitt 3 wurde der Technologieansatz, wie er in dem zu implementierenden Verfahren Anwendung finden soll, beschrieben. In diesem Umfeld eignen sich virtuelle Maschinen als Vehikel zur Verschiebung von Applikationen inklusive ihres Ressourcenbedarfs. Es erfolgte eine Abgrenzung des Verfahrens auf die durch Migrationen von VMs erzielbaren Effekte. Verschiedene Möglichkeiten zur Beeinflussung der Ressourcennutzung wurden beschrieben, wobei die CPU-Nutzung mit direkter Einflussnahme auf die Leistungsaufnahme im Fokus steht.

Die Aufgabenstellung wurde hinsichtlich ihrer Komplexität betrachtet und ist in die Klasse der NP-schweren Probleme einzuordnen. Es wurden weitere relevante Aspekte, die bei der Implementierung zu berücksichtigen sind, benannt und die nicht-funktionalen Anforderungen an das Verfahren ausgeführt.

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Nach der technologischen Abgrenzung und Konkretisierung der fachlichen und nicht-funktionalen Anforderungen erfolgt in diesem Abschnitt zunächst die Beschreibung des OHMA-Verfahrens und die Vorstellung der System-Architektur.

Für die Umsetzung des Verfahrens müssen diverse Voraussetzungen technischer und fachlicher Natur erfüllt sein. In Abschnitt 5.2 werden die technischen Randbedingungen sowie weitere Technologien beschrieben, auf denen das Verfahren fußt. Die Modellierung des Verfahrens erfolgt in einer Uppaal-Umgebung. Die Motivation hierzu sowie die grundlegende Funktionsweise von Uppaal selbst werden in Abschnitt 5.3 vorgestellt.

In den darauf folgenden Abschnitten 5.4- 5.6 werden die Automaten beschrieben, die zur Abbildung von VMs und PMs verwendet werden.

Bei der Auswahl einer VM für eine zielorientierte Migration sind diverse Kriterien zu berücksichtigen. Wie sich daraus die Migrationsstrategie ableitet, wird in Abschnitt 5.7 beschrieben. Den prozessualen Ablauf einer Leistungsänderung zeigt schließlich ein Flussdiagramm in Abschnitt 5.8.

Weitere Kenngrößen, Konfiguration und Details der Implementierung werden in Abschnitt 5.9 beschrieben.

5.1. Beschreibung des Verfahrens

Die Aufgabe des gesuchten Verfahrens besteht darin, eine Allokation von VMs zu PMs herzustellen, die eine gewünschte Leistungsaufnahme der Server hervorruft.

In der Praxis entsteht Handlungsbedarf, da in der Regel bereits eine Allokation von VMs vorherrscht und die Leistungsaufnahme, die unter dieser Allokation hervorgerufen wird, nicht den aktuellen Anforderungen entspricht. In diesem Fall wird nun eine Allokation gesucht, die zur aktuellen Anforderung passt.

Wie bereits in Abschnitt 4.4 beschrieben, ist der Aufwand, eine passende Allokation zu finden, sehr hoch, da die Anzahl möglicher Allokationen mit der Anzahl VMs und PMs exponentiell wächst. Neben der reinen Bestimmung einer Allokation, die die geforderte

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Leistungsaufnahme erfüllt, ist ein weiterer Aspekt zu berücksichtigen: Wurde eine Allokation als passend für die angeforderte Leistungsaufnahme ausgemacht, ergibt sich die Fragestellung, ob diese Allokation – ausgehend von der aktuellen Allokation – herstellbar ist. Es ist zu prüfen, ob eine Folge von VM-Permutationen existiert, sodass die ermittelte Allokation in einem iterativen Migrations-Verfahren erreicht werden kann.

Das beschriebene, initiale Provisionieren der Server mit VMs in Analogie zum Rucksackproblem ist ein sogenanntes Offline-Problem, bei dem die vollständigen Eingabedaten bereits vor der Ausführung bekannt sind und die Lösung ohne Berücksichtigung weiterer Faktoren berechnet werden kann. Da im realen Betrieb die VMs bereits auf die Server verteilt sind und eine Allokation vorhanden ist, besteht die Notwendigkeit, eine bestehende Ausgangssituation zu berücksichtigen. In Fortführung der Analogie sind die Rucksäcke bereits gefüllt und es ist nur ein Umpacken von Gegenständen gestattet, kein Auspacken aller Gegenstände.

Auf Basis dieser Voraussetzungen eignet sich die Implementierung eines schrittweisen, heuristischen Verfahrens, um die Komplexität kleinzuhalten und eine angemessene Reaktionszeit zu gewährleisten. Sogenannte Online-Verfahren, bei denen bereits Entscheidungen hinsichtlich des Lösungswegs getroffen werden, bevor alle Daten vollständig vorliegen, bieten entsprechende Möglichkeiten. Aus dieser Arbeitsweise leitet sich auch die Namensgebung des Verfahrens (Online Heuristic Machine Allocation), kurz OHMA, ab.

Konkret bedeutet dies, dass bereits Migrationen durchgeführt werden, bevor ersichtlich ist, wie eine Zielkonstellation zur Erreichung der geforderten Leistungsaufnahme aussehen kann. Weiterhin fordern die Prozesse der Regelleistung eine frühzeitig erkennbare Reaktion auf die angeforderte Leistung, wie in Abschnitt 2.8 beschrieben, der mit dieser Vorgehensweise Genüge getan ist.

Es besteht kein zwingender Anlass zur Steuerung der Migrationen durch eine zentrale Instanz, vielmehr lässt sich die Ausfallsicherheit des Verfahrens durch einen verteilten Ansatz erhöhen, wie in [TKBL13] beschrieben. Um auch den Anforderungen einer Skalierbarkeit zu entsprechen, ist auf jedem physikalischen Server (PM) eine Allocator-Komponente installiert, wie die Abbildung 5.1 zur Systemarchitektur zeigt.

5.1.1. System-Architektur

In Abbildung 5.1 ist die Architektur des Systems als Komponenten-Landkarte abgebildet.

Es erfolgt ein Monitoring der Applikationen und VMs, die auf den einzelnen PMs betrieben werden. Das SLM hält die vereinbarten SLAs vor und kann in Kombination mit den Monitoringdaten den Service-Status ermitteln. Es wird eine Datenbasis erstellt, auf deren Grundlage der Handlungsspielraum im Kontext einer Leistungsanpassung bestimmt werden kann, sodass die SLAs weiterhin eingehalten werden können.

Eine akute Anforderung zur Anpassung der Leistungsaufnahme wird ebenfalls beim SLM platziert. Auf Basis dieser Informationen kann das SLM einen Zielwert (*target*) für die

5.1. Beschreibung des Verfahrens

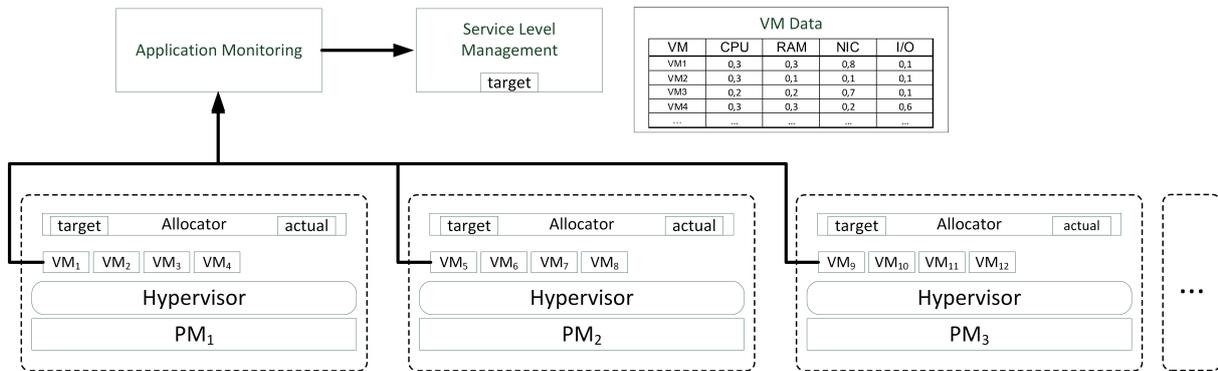


Abbildung 5.1.: Komponenten des Systems: Überwachung der Systemdaten

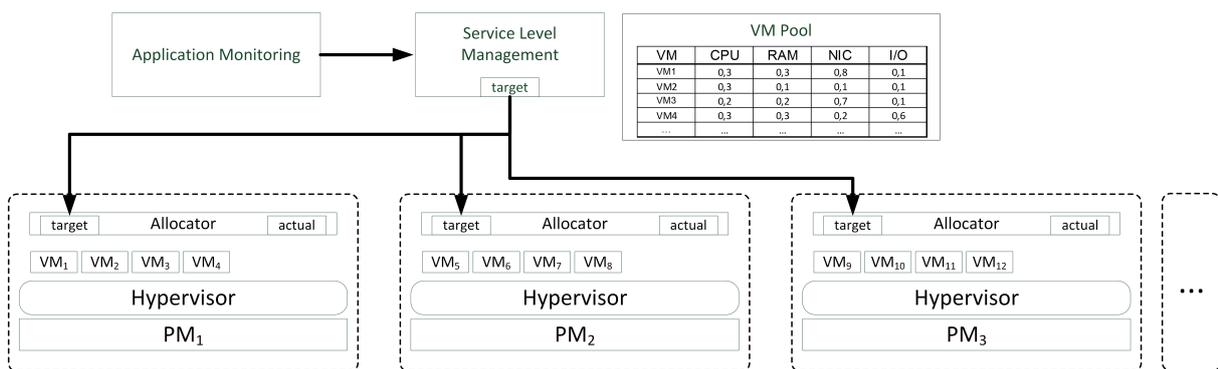


Abbildung 5.2.: Komponenten des Systems: Propagieren der Ziel-Leistungsaufnahme

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Gesamtleistungsaufnahme der Server festlegen und an die Allocator-Komponenten propagieren, wie Abbildung 5.2 zeigt.

Nach der Übertragung der Ziel-Leistungsaufnahme initiieren die jeweiligen Allocator-Komponenten Migrationen, hierzu erfolgt ein Austausch über die Kenndaten der einzelnen VMs, wie in Abbildung 5.3 dargestellt.

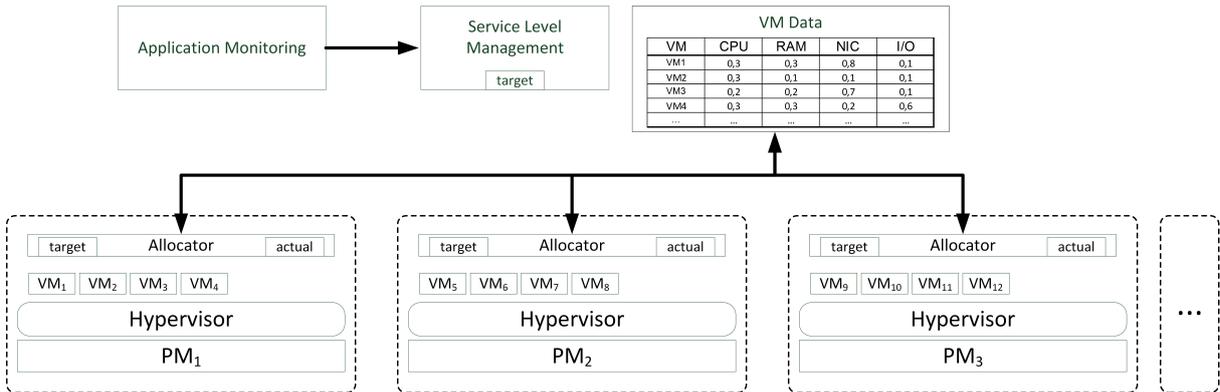


Abbildung 5.3.: Komponenten des Systems: Austausch von Kenndaten der betriebenen VMs

In der Folge werden die zur Verfügung stehenden VMs analysiert und entsprechend der Leistungsvorgabe Migrationen durchgeführt, bis der Zielwert erreicht wird.

Es wird eine gemeinsame Zielvorgabe für die Gesamtleistungsaufnahme der Server an jede PM propagiert, die wiederum autonom agiert und Migrationen zur Erreichung der Vorgabe veranlasst. Durch diese Architektur können einzelne PMs ausfallen, wobei die verbliebenen PMs weiter an der Zielvorgabe arbeiten. Ebenso ist eine Erweiterung des Systems im laufenden Betrieb um weitere PMs und VMs durchführbar. Der verteilte Ansatz bedient ebenfalls die in Abschnitt 4.6.2 benannten nicht-funktionalen Anforderungen.

Ausgehend von einer gegebenen Allokation kann beim Eintreffen einer neuen Anforderung bezüglich der Leistungsaufnahme zunächst ermittelt werden, ob die Leistungsaufnahme abgesenkt oder erhöht werden soll. In beiden Fällen müssen Migrationen von VMs initiiert werden. Der Auswahlprozess und die Kriterien einer entsprechenden VM und Ziel-PM unterscheidet sich in den genannten Fällen jedoch signifikant.

Die PMs werden daher in zwei Gruppen (Gruppe A und Gruppe B) mit alternierender Anordnung eingeteilt, wie in Abbildung 5.4 dargestellt. Eine Gruppe erhält dabei die Kompetenzen für das Absenken der Leistungsaufnahme und die andere Gruppe jene für die Erhöhung der Leistungsaufnahme. Die PMs einer Gruppe arbeiten im Verbund an einem gemeinsamen Ziel und die VMs, die bereits innerhalb derselben Gruppe betrieben werden, sind daher von weiteren Optimierungsschritten ausgeschlossen. Bei einer Konstellation, in der die Servergruppe eine CPU-Auslastung unter 100% aufweist, würde eine VM-Migration innerhalb der Gruppe nur eine Verschiebung der Leistungsaufnahme bewirken. Ist nach einigen Migrationsschritten bereits eine Überprovisionierungssituation vorhanden und die CPU-Auslastung der PMs liegt über 100%, wären Effekte nur möglich,

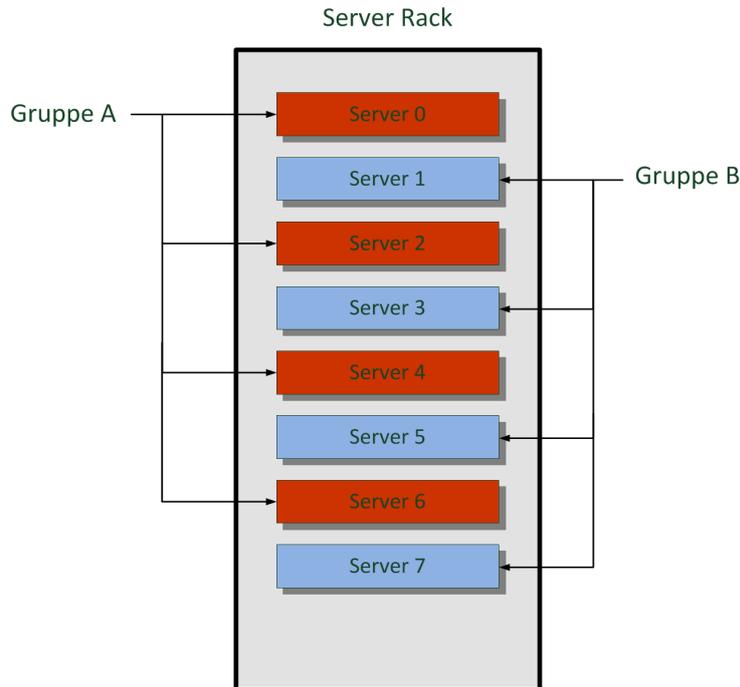


Abbildung 5.4.: Aufteilung der Server in zwei Gruppen

wenn die betreffende VM eine sehr hohe CPU-Auslastung aufweist. Die CPU-Auslastung der VM muss mindestens so hoch sein wie der Überprovisionierungsgrad der abgebenden PM. Eine Migration innerhalb derselben Gruppe würde die Leistungsaufnahme der Server-Gruppe daher nicht relevant verändern.

Dieser Umstand verkleinert zudem bereits die Menge möglicher Kandidaten, da nur VMs, die aktuell auf der antagonistischen Gruppe betrieben werden, für eine zielgerichtete Migration in Betracht kommen. Die Migrationen werden so lange fortgeführt, bis die geforderte Leistungsaufnahme erreicht wurde oder keine weiteren Kandidaten für Migrationen, die zur Zielerreichung beitragen, gefunden werden können.

5.2. Voraussetzungen und Annahmen

Für die Anwendbarkeit des Verfahrens müssen gewisse fachliche Voraussetzungen gegeben sein. Weiterhin gibt es technologische Abhängigkeiten, die in diesem Teil der Arbeit beschrieben werden.

5.2.1. Dynamische Leistungsaufnahme

Als Basisvoraussetzung werden Server benötigt, die über Mechanismen zur Leistungsanpassung verfügen. Insbesondere ältere Server und auch Server aus dem Bereich des High-Performance Computing (HPC) verfügen häufig nicht über DVFS-Technologien, wie in

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Abschnitt 2.3.1 beschrieben. Die Leistungsaufnahme dieser Server ist über das gesamte Leistungsspektrum annähernd konstant. Abbildung 5.5 zeigt die Leistungsdaten (gepunktete Linie) eines Sun-Netra-X4250 Servers mit dem beschriebenen Verhalten.



Abbildung 5.5.: Server mit geringem adaptiven Verhalten bezüglich der Leistungsaufnahme; Grafik unverändert aus [Sta09] übernommen

Moderne Server, die mit DVFS-Technologien ausgestattet sind, können auf eine veränderte Auslastung deutlich besser reagieren und die Leistungsaufnahme adaptieren, wie es in den Benchmarkergebnissen am Beispiel des DellPowerEdge-R830 in Abbildung 5.6 zu sehen ist.

Die Leistungsaufnahme variiert bei diesem Modell in einem Bereich von 86 - 562 Watt. Die durch diese Technologie mögliche Leistungsvarianz wird benötigt, damit sich die durch Umverteilung der VMs veränderte Serverauslastung in der Leistungsaufnahme der Server niederschlägt. Die Adaption der Leistungsaufnahme der Server an ihre Auslastung ist essentielle Grundlage des Verfahrens und stellt gleichzeitig den Rahmen der erzielbaren Varianz in der Leistungsaufnahme dar. Diverse Ansätze, wie der von Laszewski et. al [LWYH09] zur Reduzierung der Leistungsaufnahme von Server-Prozessoren, basieren auf DVFS-Technologie. Auch das in dieser Arbeit vorgestellte Verfahren fußt auf der DVFS-Technologie.

5.2.2. Virtualisierungstechnologie

Die DVFS-Technologie ist Basis für eine Änderung der Server-Leistungsaufnahme, mit der sich die Leistungsaufnahme an die IT-Last adaptiert. Andere Verfahren – wie in Abschnitt 3 beschrieben – verwenden Scheduler in Kombination mit Batch-Jobs, um die IT-Last

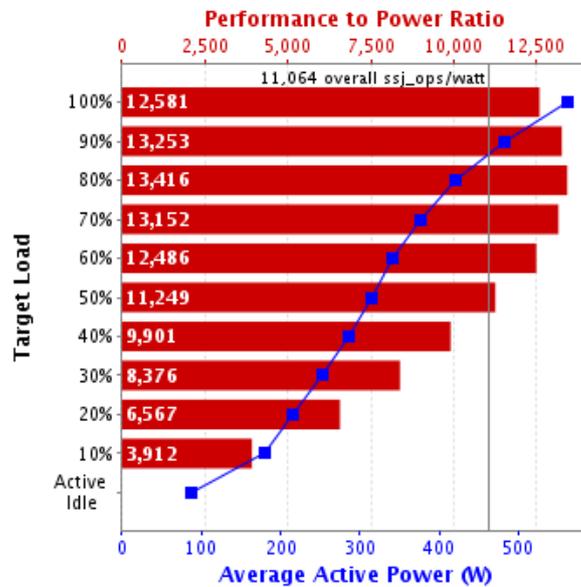


Abbildung 5.6.: Server mit adaptivem Verhalten bezüglich der Leistungsaufnahme; Grafik unverändert aus [Sta16] übernommen

und in der Folge die Leistungsaufnahme der Server zu beeinflussen. Bei der Steuerung der IT-Last mit Hilfe eines Schedulers sind fachliche Kenntnisse über die Applikationen erforderlich, die beispielsweise im Umfeld des IaaS nicht vorhanden sind, wie in Abschnitt 1.2 beschrieben.

Um die Entkopplung von einer fachlichen Job-Steuerung zu erreichen, sollen hier mit Hilfe der Server-Virtualisierung ganze VMs mit ihren Applikationen verschoben werden. Die Virtualisierungstechnologie dient in diesem Zusammenhang als Vehikel zur Migration von IT-Last.

Um auch die Anforderung eines unterbrechungsfreien Betriebs zu gewährleisten, sind Virtualisierungstechnologien nötig, die ebenfalls die Möglichkeit zur Live-Migration bieten, wie in Abschnitt 2.4.2 beschrieben. Weiterhin muss der Hypervisor eine entsprechende API zur Verfügung stellen. Es wird die Möglichkeit zum Abfragen von Daten und Betriebszuständen von VMs benötigt, um mit diesen Informationen die weiteren Migrationsschritte zu planen. Ebenfalls muss das Initiieren von Migrationen über eine API ermöglicht werden.

5.2.3. Weitere Randbedingungen

Weitere Abhängigkeiten ergeben sich durch die Details der Implementierung:

- Wie beschrieben, basiert das Verfahren auf zwei antagonistischen Gruppen. Es muss daher von jedem PM-Gruppen-Typ mindestens ein Server vorhanden sein. Das Verfahren ist erst ab zwei physikalisch vorhandenen Servern einsetzbar.

5. Modellierung des Verfahrens: *Online Heuristic Machine Allocation (OHMA)*

- In der Initialisierungsphase des Systems werden die VMs auf die PMs verteilt. Es wird vorausgesetzt, dass alle VMs auf die PMs zu verteilen sind und die PMs die Ressourcenanforderungen abdecken, sodass nach der Verteilung der VMs keine Überlastung der PMs erreicht wird. Die Ressourcen-Anforderungen der VMs dürfen die PM-Kapazitäten nicht übersteigen.
- Um eine maximal mögliche Leistungsreduzierung des Systems erreichen zu können, muss sicher gestellt sein, dass alle VMs auf die entsprechende PM-Gruppe migriert werden können. Dazu müssen die Server der Gruppe mit entsprechender RAM-Größe ausgestattet sein, andernfalls verringert sich das Potential zur Leistungsreduzierung entsprechend. Der Prozess zur Reduzierung der Leistungsaufnahme endet mit dem Ausschöpfen der maximalen RAM-Kapazität der Servergruppe.

5.2.4. Fachliche Voraussetzungen

Neben den technischen Voraussetzungen gibt es weitere Anforderungen, die die betriebenen Applikationen betreffen. Ein Großteil der in Abschnitt 3 vorgestellten Verfahren optimiert die Serverauslastung unter der vorherrschenden IT-Last. Unter IT-Last wird hier die von den KVMs und hierin betriebenen Applikationen hervorgerufene Serverauslastung verstanden. Bekannte Verfahren zur Adaption der Leistungsaufnahme, die die IT-Last verändern, basieren auf der Verarbeitung von Batch-Jobs, bei denen es keine direkten User-Interaktionen gibt. Die User erwarten von Applikationen ein schnelles Antwortverhalten bei Interaktionen, somit scheiden Prozesse und Applikationskomponenten für dieses Verfahren aus, die derartige Applikationsanteile abbilden. Ebenso wenig geeignet sind die Kontent-Anteile von Voice- oder Streaming-Diensten. Ideal geeignet sind Applikationen, die einen hohen Berechnungsaufwand aufweisen und in Intervallen von mehreren Stunden in der Leistung reduziert werden können. Beispiele hierfür sind Workflowsysteme, die eine asynchrone Taskverarbeitung durchführen, oder auch Transformationskomponenten von Data Warehouse Systemen ebenso wie die Prozesse des Data Mining.

Bei einer monolithischen Applikation muss die gesamte Applikation mit den höchsten Anforderungen einer Teil-Komponente betrieben werden und stellt somit wenig Potential für adaptive Verfahren dar. Aktuelle Trends gehen in Richtung agiler Softwareentwicklung [Wol18, BHJ⁺18], bei der komplexe Applikationen aus einer Vielzahl von sogenannten Microservices aufgebaut werden. Jeder Microservice bildet dabei nur eine kleine, überschaubare Aufgabe ab und ist von anderen Diensten oder Microservices weitgehend unabhängig. Die Microservices können auch in unterschiedlichen Technologien implementiert sein. Dieser Trend – weg von einer monolithischen Architektur – erlaubt es, Komponenten mit geringeren Anforderungen zu separieren und in leistungsadaptive Verfahren zu integrieren. Es wird daher angenommen, dass in den VMs Applikationen betrieben werden, die zu der beschriebenen Anforderungskategorie gehören.

5.3. Modellierung des Verfahrens in einer Uppaal-Umgebung

Zur Simulation von Rechenzentren sind bereits eine ganze Reihe von Werkzeugen verfügbar. Im Bereich der virtualisierten Rechenzentren wird die Möglichkeit geboten, Serverkontingente zu planen und die Ergebnisse bezüglich Laufzeiten und Leistungsaufnahme zu simulieren. Insbesondere das Toolkit Cloudsim, wie in [CRB⁺11] beschrieben, bietet die Möglichkeiten VMs und Server in einem IaaS-Umfeld abzubilden und zu simulieren. Das Framework OpenStack Neat [BB15] bietet Simulationen von VMs auf Basis eines definierten Workloads, wobei verschiedene Allokationsstrategien evaluiert werden können. Der Kern des hier vorgestellten Verfahrens beruht hingegen auf der Überprovisionierung von Serverressourcen, Permutationen von VMs, sowie eines kontinuierlichen Applikationsbetriebes. Die Abbildung des hier entwickelten Verfahrens ist daher mit diesen Frameworks nicht möglich. Um die spezifischen Aspekte in das Toolkit Cloudsim zu integrieren, wären umfangreiche Änderungen und Erweiterungen nötig.

In dieser Arbeit steht das Lösen eines Scheduling-Problems im Vordergrund, bei dem die Permutation von VMs und deren Randeffekte abgebildet werden müssen. Das Prozessmodell sieht ein Initiieren von Maßnahmen zum Erhöhen oder Absenken der Leistungsaufnahme vor; die konkret erreichten Effekte werden zur Laufzeit ermittelt. Insbesondere sind die Abbildung des Systemzustandes und die zeitlichen Abhängigkeiten von Bedeutung. Aus diesem Grund wurde eine Implementierung in dem Simulationsframework Uppaal gewählt, das für die Abbildung von Scheduling-Problemen geeignet ist [DILS09]. Hier lassen sich zeitkritische Bedingungen und Systemzustände abbilden sowie Systemeigenschaften nachweisen und simulieren.

5.3.1. Simulationsframework Uppaal

Zur Analyse und Lösung komplexer Probleme werden oftmals endliche Automaten verwendet, denn sie bieten eine einfache Struktur und übersichtliche Darstellung. Sie werden zur Implementierung von Netzwerkprotokollen, im Compilerbau und auch zur Modellierung von Prozessen und Abläufen im Workflow-Management eingesetzt. Mit der Verwendung von Automaten-Modellen kann zudem der Anforderung einer transparenten Darstellung der Verfahrensweise entsprochen werden.

Zur Simulation von endlichen Automaten existieren eine ganze Reihe an Frameworks wie beispielsweise *Automaton-Simulator* [Sci], *JFLAP* [JFL20] oder *FSM simulator* [FSM]. Für eine Abbildung des vorgestellten Verfahrens werden Datenstrukturen für die Modellierung der VM- und PM-Profile benötigt. Mit Hilfe von Funktionen müssen die Auswahl der zu migrierenden VMs unterstützt, sowie die Berechnung der Leistungsdaten des Gesamtsystems berechnet werden. Weiterhin wird die Abbildung von zeitlichen Bedingungen mit Hilfe eines Uhren-Konstrukts benötigt.

Ein Werkzeug, das diese Funktionalitäten bietet, ist das Simulationsframework Uppaal mit grafischem Editor. Es wird an den Universitäten Uppsala und Aalborg entwickelt

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

und bietet die Möglichkeit einer Implementierung von Realzeit-Systemen mit sogenannten Timed-Automata [LPY97]. Eine Synchronisation der Automaten über Kommunikationskanäle wird geboten, ebenso die Implementierung von lokalen, globalen Variablen und Funktionen in einer C-ähnlichen Programmiersprache. Darüber hinaus wird die Abbildung von Realzeitbedingungen mit Hilfe von Uhren-Variablen ermöglicht. Diese Funktionen werden von anderen genannten Frameworks nicht unterstützt.

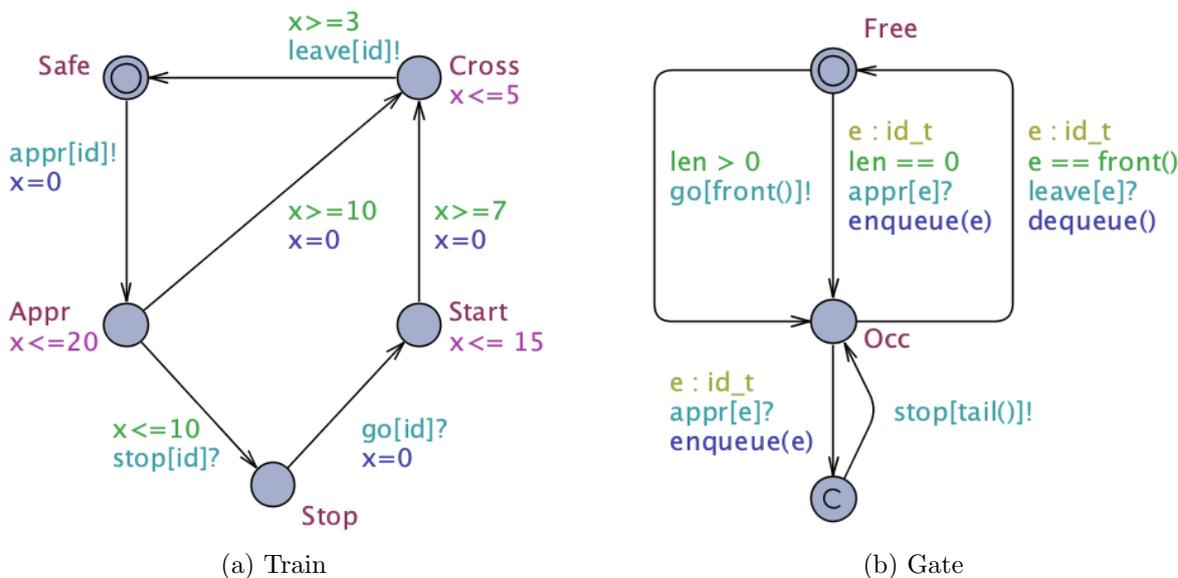


Abbildung 5.9.: Beispiel für die Modellierung eines Bahnübergangs; Screenshot eines mit Uppaal ausgelieferten Implementierungsbeispiels

In Abbildung 5.9 ist ein Beispiel für die Modellierung eines Bahnübergangs dargestellt. Es ist jeweils ein Template für einen Zug und einen Bahnübergang modelliert worden. Für die Abbildung der zeitlichen Bedingungen wird hier die Uhrenvariable x verwendet. Der Startzustand ist durch einen Doppel-Kreis gekennzeichnet.

Nach der Modellierung der Templates kann eine Simulation des erstellten Systems durchgeführt werden. Abbildung 5.10 zeigt das Hauptfenster von Uppaal im Simulations-Modus. Im rechten Teil sind die Instanzen der Automaten abgebildet, in diesem Fall sind es zwei Züge mit den IDs 0 und 1 sowie ein Bahnübergang. Während einer Simulation werden hier die aktuellen Zustände der Automaten-Instanzen und schaltbare Transitionen rot markiert.

Im mittleren Teil können Parameter wie Variablen oder Bedingungen angezeigt werden. Das Feld „Enabled Transitions“ im linken Teil zeigt die Transitionen, die aktuell schalten können, darunter sind die Zustände und Transitionen dargestellt, die in der Simulation bereits durchlaufen wurden. Für eine spätere Verwendung und zur Dokumentation lassen sich Tracefiles aufzeichnen, die erneut geladen werden können, beispielsweise um einen bestimmten Simulationsstand wieder herzustellen.

Um die Modellierung in Uppaal möglichst schlank zu halten, werden in dieser Arbeit nur zustandsbehaftete, zeit- oder effektrelevante Mechanismen abgebildet, die essentiell für das

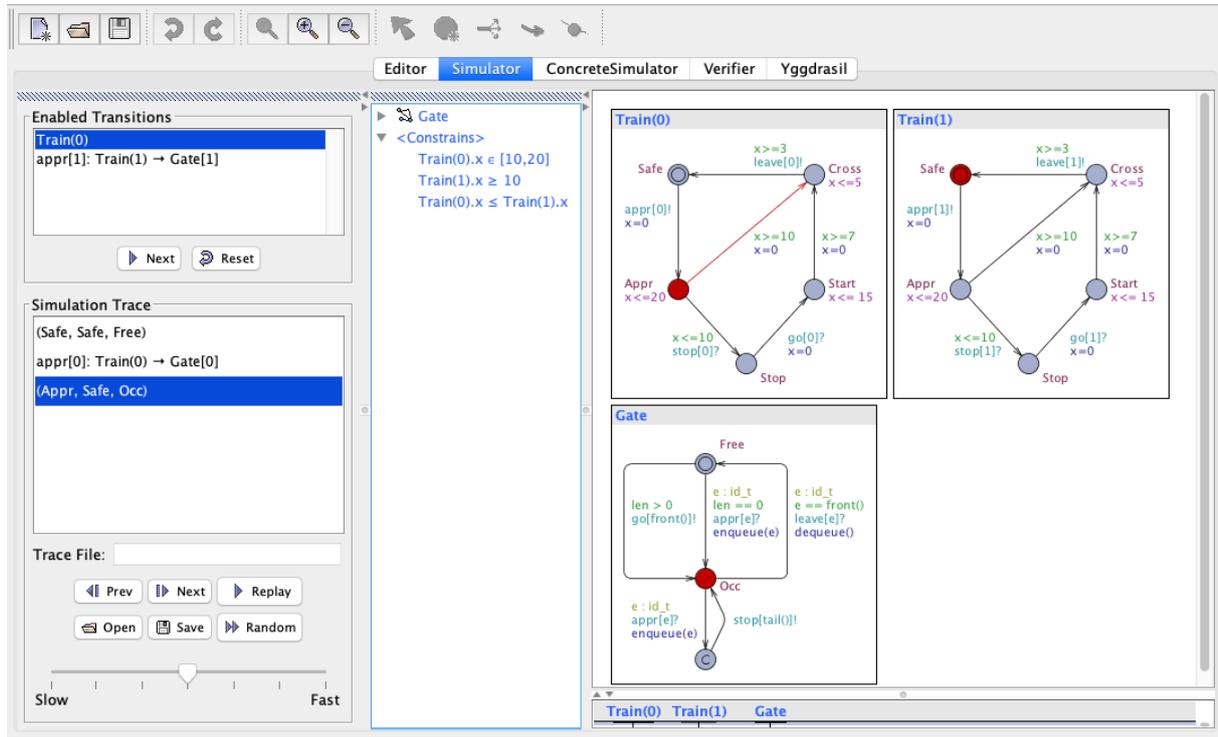


Abbildung 5.10.: Hauptfenster des Simulationssystems Uppaal

Verhalten oder die Ergebnisfindung sind. Nur so lassen sich Systeme geringer Komplexität erzeugen, deren Eigenschaften analysiert werden können. Insbesondere sind mehrdimensionale oder multi-kriterielle Lösungen problematisch, da sich hier exponentiell wachsende Lösungs- und mögliche Zustandsräume ergeben, deren Evaluation sehr ressourcenintensiv ist.

Mit der Modellierung von Automaten in dem Simulationsframework Uppaal kann eine Transparenz über das Verfahren und die Prozessschritte erzeugt werden, wie in Forschungsfrage Nr. 4 gefordert.

5.4. Modell zur Abbildung einer VM

Der Ressourcenbedarf einer VM wird in der Simulation als Vektor \vec{r}_{VM} repräsentiert, hier als 4-Tupel der Ressourcen-Auslastung von CPU, RAM, HDD und NIC. Für eine Vergleichbarkeit des Ressourcenbedarfs einer VM müssen die Bedarfswerte auf einem gemeinsamen Basiswert beruhen, der für alle VMs identisch ist. Hierzu erfolgt eine Normierung des Ressourcenbedarfs. Der Ressourcenbedarf, der einer VM zugeschrieben wird, bezieht sich auf ein Referenzsystem, mit dem der Ressourcenbedarf ermittelt wird. Bei einer homogenen Serverlandschaft, in der nur ein Servertyp zum Einsatz kommt, kann eben dieser Servertyp als Referenzserver dienen. Die Werte des Ressourcenbedarfs werden als

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

prozentuale Auslastung der jeweiligen Ressource ermittelt und vereinen somit den Bedarf von VM und der betriebenen Applikationen.

In heterogenen Rechenzentren, in denen unterschiedliche Server betrieben werden, muss ein Gewichtungsfaktor bestimmt werden, mit dem sich die hervorgerufene Auslastung eines Servers durch eine betriebene VM abbilden lässt.

Neben einer Ressourcen-Darstellung muss das Modell das charakteristische Verhalten einer VM abbilden, welches im Rahmen dieses Anwendungsfalls von Relevanz ist. Hierzu gehören explizit die Prozesse zur Migration einer VM. Für den in dieser Arbeit implementierten Ansatz sind dies folgende Zustände:

- ausgeschaltet (Startzustand)
- initialisiert (initialized)
- in Betrieb (operate)
- für eine Migration reserviert (reserved)
- VM in Migration begriffen: Migrationsprozess wurde angestoßen, RAM-Inhalt wird kopiert (wait_RAM)
- Migration abschließen (go_migrate)

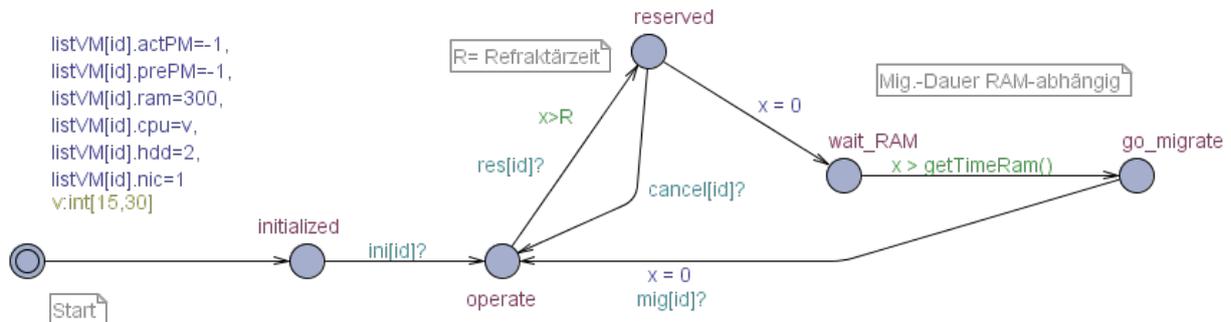


Abbildung 5.11.: Automaten-Template einer VM; Original-Ansicht in dem grafischen Editor des Uppaal-Frameworks

5.4.1. Verhalten einer VM und Ablauf einer Migration

In Abbildung 5.11 ist das Template eines Automaten zur Simulation einer VM dargestellt. Beim Start einer Simulation werden Instanzen dieses Templates erzeugt und mit einer individuellen VM-ID versehen. Entsprechend gibt es zu jeder simulierten VM eine Instanz in der Simulation mit zugehöriger VM-ID.

Ausgehend vom Startzustand wird die VM zunächst mit ihren Kenngrößen initialisiert und wartet in dem Zustand *initialized* auf die Inbetriebnahme durch eine PM. Auf dem Kanal *ini* wartet die VM auf eine Anfrage mit der eigenen ID. Meldet eine PM über diesen

Kanal die Anfrage zum Betrieb dieser VM an, kann die VM in Betrieb genommen werden und der Automat wechselt in den Zustand *operate*. In diesem Zustand verweilt die VM, sofern keine Migration ansteht.

Eine betriebene VM kann ein passender Kandidat für eine optimierende Migration sein. Eine PM, die diese optimierende Migration initiiert und eine bestimmte VM ausgewählt hat, reserviert diese VM über den Kanal *res* und übermittelt die VM-ID. Wird eine VM über diesen Kanal angefragt, wechselt sie in den Zustand *reserved*. Der Zustandswechsel und der Übergang in den *reserved*-Zustand ist jedoch an eine Bedingung geknüpft. Um eine allzu häufige Migration von VMs zu verhindern und eine Ruhephase zu implementieren, in der die VM betrieben werden kann, ohne an Migrationsprozessen teilzunehmen, wurde ein Zeitintervall definiert, in dem kein Übergang in den Zustand *reserved* möglich ist. Das Zeitintervall beginnt nach der letzten Migration durch das Zurücksetzen der Uhr x . Erst nachdem die Zeitdauer R verstrichen ist, wird der Zustandswechsel möglich.

Die Einhaltung des Ruheintervalls wird bewusst nicht vor der Reservierungsanfrage geprüft. Zwischen der Reservierung und vor einer eventuellen Migration der VM kann hinreichend Zeit vergehen, sodass eine Migration dennoch möglich ist. Die PMs stellen aus den möglichen Kandidaten für eine Migration zunächst eine Vorauswahl zusammen und erst in einem zweiten Schritt fällt die Entscheidung für eine bestimmte VM. In der Zeitspanne von der Reservierung zur Vorauswahl bis zur tatsächlichen Entscheidung für eine VM kann das Zeitintervall bereits verstrichen sein und die VM steht für eine Migration zur Verfügung.

Ist die Auswahl der PM auf eine andere VM gefallen, wird die Reservierung über den Kanal *cancel* abgebrochen und die VM wechselt vom Zustand *reserved* zurück in den Zustand *operate*. Da keine Migration stattgefunden hat und sich der letzte Migrationszeitpunkt nicht verändert hat, wird die Uhr x auch nicht zurückgesetzt.

Ist die Auswahl auf eine bestimmte VM gefallen, wechselt diese VM in den Zustand *wait_RAM*. Mit diesem Zustand wird eine Phase der Online-Migration von VMs modelliert, wie in Abschnitt 2.4.2 beschrieben. Für eine Migration muss der Hauptspeicher von der Quell-PM auf die Ziel-PM übertragen werden, die Dauer dieses Vorgangs hängt von der Hauptspeichergroße und dessen Änderungsrate ab. Daher wird beim Wechsel in diesen Zustand die Uhr x zurückgesetzt und so lange in dem Zustand *wait_ram* verweilt, bis die Zeitspanne zur Übertragung des Arbeitsspeichers verstrichen ist, dabei wird dieselbe Uhr x verwendet, die zuvor für den letzten Migrationszeitpunkt zuständig war. An dieser Stelle kann dieselbe Uhr x benutzt werden, da der letzte Migrationszeitpunkt nun keine Rolle mehr spielt. Die Modellierung der Migrationsdauer beruht dabei auf den Ergebnissen der Untersuchungen von Semenov et. al [SVT11] und Verma et. al [VAN08], wie in 2.4.3 beschrieben. In der Folge wird in den Zustand *go_migrate* gewechselt, der den Umschaltzeitpunkt und die Inbetriebnahme der VM auf der Ziel-PM repräsentiert. Beim Übergang in den Zustand *operate* wird die Uhr x zurückgesetzt und dient damit wieder zur Prüfung der Ruhezeiten.

5.5. Modell zur Abbildung einer PM

Wie bei der VM, wird auch der Auslastungszustand der PM in der Simulation als Ressourcen-Vektor \vec{r}_{PM} repräsentiert, hier als 4-Tupel von CPU, RAM, HDD und NIC. Für eine Vergleichbarkeit des Ressourcenangebots einer PM müssen die Ressourcenwerte auf einem gemeinsamen Basiswert beruhen, der für alle PMs gilt. Hierzu erfolgt eine Normierung des Ressourcenangebots. Bei einer homogenen Serverlandschaft, in der nur ein Servertyp zum Einsatz kommt, kann eben dieser Servertyp als Referenzserver dienen. Dies ist der Regelfall. In heterogenen Rechenzentren, in denen unterschiedliche Server betrieben werden, muss ein Gewichtungsfaktor bestimmt werden, mit dem sich die Auslastung eines Servers auf eine entsprechende Auslastung des Referenzservers abbilden lässt.

In Abbildung 5.12 ist das Uppaal-Template des Automaten zur Simulation einer PM dargestellt. Für eine übersichtlichere Darstellung wurde der Automat in eine schematische Grafik überführt, Abbildung 5.13 zeigt diese schematische Darstellung. Die Transitionen, die ein direktes Erreichen des Startzustands erlauben, sobald die geforderte Leistungsaufnahme erreicht wurde, sind nicht enthalten.

Der Automat lässt sich grob in drei funktionale Bereiche einteilen:

1. Es werden initiale Prozesse zur Verteilung der VMs bei Systemstart durchgeführt und anschließend auf eine Anforderung zur Änderung der Leistungsaufnahme gewartet.
2. Trifft eine Anforderung zur Erhöhung der Leistungsaufnahme ein, wird der linke Flügel des Automaten durchlaufen, eine VM zur Bedienung dieser Anforderung gewählt und migriert.
3. Bei einer Anforderung zum Absenken der Leistungsaufnahme wird der rechte Teil durchlaufen. Die Kriterien zur Auswahl der VM unterscheiden sich maßgeblich von denen zum Erhöhen der Leistungsaufnahme, wie im Folgenden konkreter erläutert wird.

Wie ersichtlich ist, deckt das Template den Aufgabenbereich beider Server-Gruppen ab. Auch die simulierte Instanz der PMs beinhaltet beide Funktionalitäten. Das Modell erlaubt es damit auch zur Laufzeit noch eine Festlegung der Gruppenzugehörigkeit durchzuführen oder diese zu ändern. Mit dieser technischen Möglichkeit soll die Anforderung bezüglich Flexibilität und Erweiterbarkeit aus Abschnitt 4.6.2 erfüllt werden.

5.5.1. Auslastung einer PM

Um den Zustand einer PM bewerten zu können, benötigt der Algorithmus Zugriff auf Daten bezüglich der Auslastung einer PM. Diese Daten lassen sich für gewöhnlich über die API des Hypervisors abfragen. Im Modell muss der Zusammenhang zwischen Anzahl betriebener VMs und der Auslastung einer PM nachgebildet werden, um entsprechende Effekte durch VM-Migrationen erzeugen zu können. Die Erstellung eines präzisen Modells

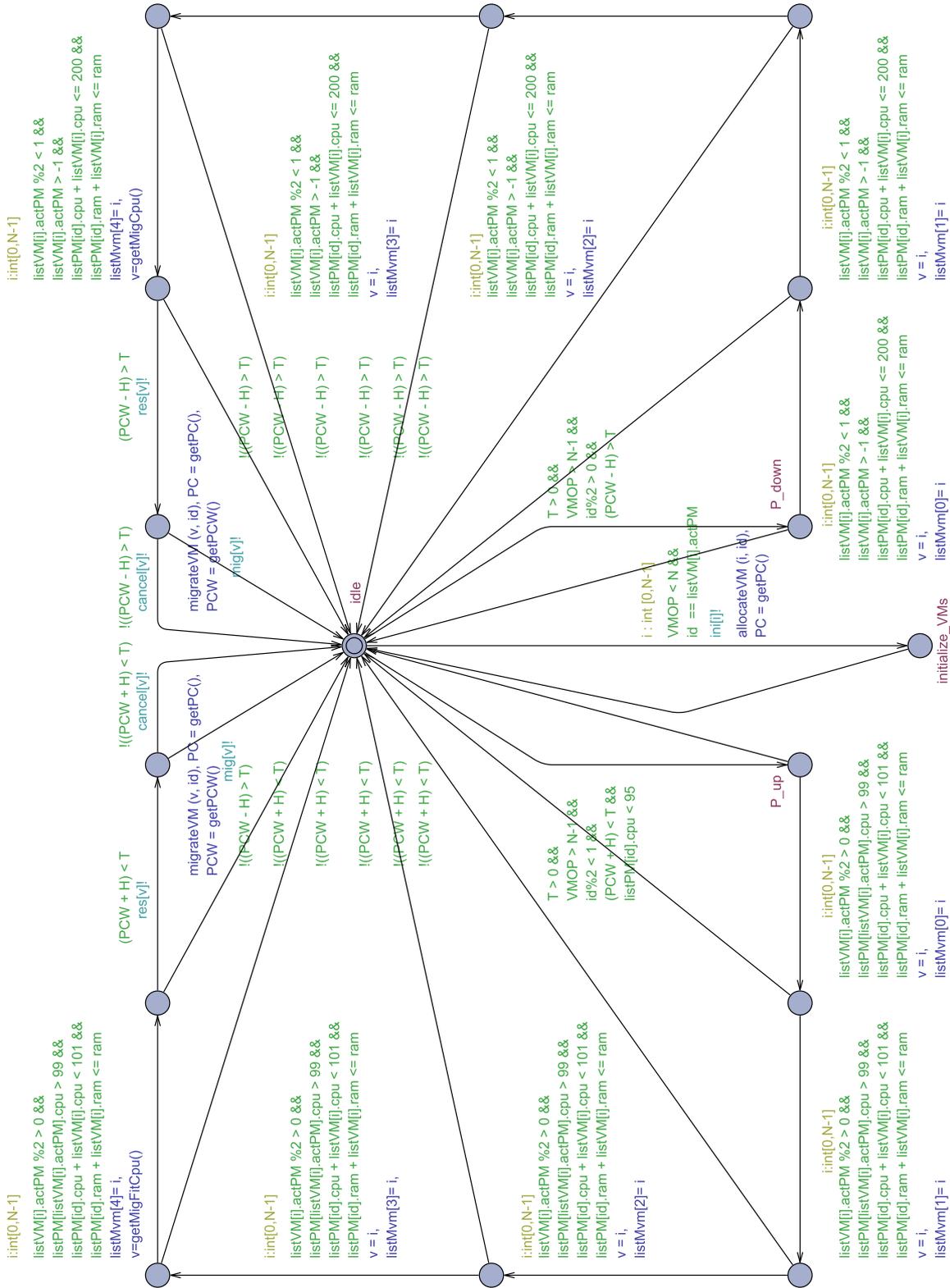


Abbildung 5.12.: Uppaal-Template einer PM

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

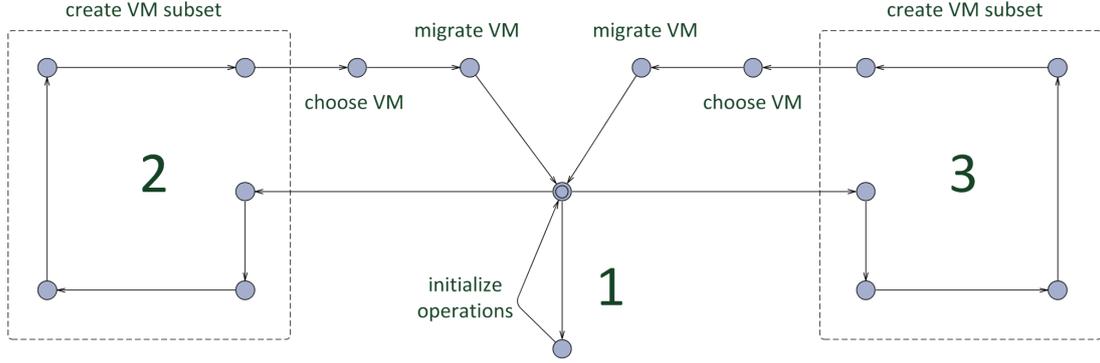


Abbildung 5.13.: Schematische Darstellung des Automaten-Templates zur Simulation einer PM

ist dabei nicht von Bedeutung, da diese Daten im realen Szenario über entsprechende APIs bezogen werden können. Vielmehr steht das Prinzip und der Zusammenhang von Serverauslastung und Leistungsaufnahme im Vordergrund, damit die durch eine Migration erzielten Effekte abgebildet werden können.

Um die Auslastung der Ressourcen einer PM zu taxieren, werden die Ressourcen-Vektoren \vec{r}_{VM} sämtlicher auf der PM betriebenen n VMs summiert. Somit ergibt sich die Gesamtauslastung einer PM \vec{r}_{PM} durch:

$$\vec{r}_{PM} = \begin{pmatrix} \vec{r}_{PM_{CPU}} \\ \vec{r}_{PM_{RAM}} \\ \vec{r}_{PM_{HDD}} \\ \vec{r}_{PM_{NIC}} \end{pmatrix} = \begin{pmatrix} \vec{r}_{VM1_{CPU}} \\ \vec{r}_{VM1_{RAM}} \\ \vec{r}_{VM1_{HDD}} \\ \vec{r}_{VM1_{NIC}} \end{pmatrix} + \begin{pmatrix} \vec{r}_{VM2_{CPU}} \\ \vec{r}_{VM2_{RAM}} \\ \vec{r}_{VM2_{HDD}} \\ \vec{r}_{VM2_{NIC}} \end{pmatrix} + \dots + \begin{pmatrix} \vec{r}_{VMn_{CPU}} \\ \vec{r}_{VMn_{RAM}} \\ \vec{r}_{VMn_{HDD}} \\ \vec{r}_{VMn_{NIC}} \end{pmatrix} \quad (5.1)$$

Beim Summieren der einzelnen Ressourcen können Werte über 100% erreicht werden, in diesem Fall ist eine Ressource überprovisioniert. Die Leistungsaufnahme des Servers kann die 100%-Marke jedoch nicht überschreiten. Wurde beim Migrieren die hundertprozentige Auslastung überschritten, steigt die Leistungsaufnahme des Servers nicht weiter an. Die Leistungsaufnahme des abgebenden Servers sinkt jedoch in Kombination mit der DVFS-Technologie und es wurde eine Leistungsreduzierung herbeigeführt. Bei der Betrachtung der Performance von Applikationen muss diese Überprovisionierung jedoch Berücksichtigung finden.

Experimente [BR17] haben gezeigt, dass es bei der Überprovisionierung zu proportionalen Effekten bei der Anforderung und Auslastung der übrigen Ressourcen kommt. Für die Ermittlung der Gesamtpformance der Applikationen können auf dieser Basis Modelle für die Bewertung der Applikations-SLAs erstellt werden. Modelle für Applikationsperformance werden in dieser Arbeit nicht weiter betrachtet, da erst die Performance in Kombination mit einem vereinbarten SLA die Ableitung von gegebenenfalls nötigen Maßnahmen ermöglicht. Die Bewertung der Performance anhand von Monitoringdaten und Überwachung der SLAs obliegt dem SLM, wie in Abschnitt 2.5 beschrieben.

Besondere Berücksichtigung findet die Sicherstellung von SLAs durch die Implementierung eines Urgent-Flags. Neben der generellen Beeinflussung der Leistungsaufnahme durch das SLM und Beachtung der SLAs kann das SLM bei VMs individuell ein Urgent-Flag setzen, falls eine SLA-Verletzung droht. Ist das Urgent-Flag gesetzt, bevor eine Anforderung zum Absenken der Leistungsaufnahme eintrifft, wird die betreffende VM nicht zur Erzeugung einer Überprovisionierung migriert. Eine Verknappung der Ressourcen und damit einhergehendes Verringern der Applikationsperformance wird für diese VMs damit umgangen.

5.5.2. Verhalten einer PM

Generell werden Migrationen nur von der aktiven PM-Gruppe initiiert. Dabei beginnt die Allocator-Komponente einer PM mit der Auswahl einer VM und führt gegebenenfalls eine Migration auch nur in Richtung der PM selbst durch. Dieses Vorgehen kann als „Pull“-Verfahren bezeichnet werden und stellt die gezielte Steuerung der Effekte sicher.

Im Modell der PM werden die Abläufe modelliert, die zum Algorithmus der VM-Auswahl nötig sind und zu den relevanten Aufgaben eines Hypervisors gehören. Abbildung 5.12 zeigt das in Uppaal implementierte Template eines Automaten zur Instanziierung der PMs.

Ausgehend vom Start-Zustand werden bei der Initialisierung des Systems zunächst die VMs auf die PMs verteilt. Nach der Initialisierung und Inbetriebnahme wird im Anschluss der Start-Zustand wieder erreicht und die PM wartet in der Folge auf das Feststellen erneuten Handlungsbedarfs in Form einer Anforderung zum Ändern der Leistungsaufnahme.

Muss die Leistungsaufnahme erhöht werden, wird der in Abbildung 5.12 und 5.13 dargestellte linke Flügel des Automaten durchlaufen. Hierbei wird zunächst geprüft, ob die PM Kapazitäten zur Aufnahme weiterer VMs zur Verfügung stellen kann, andernfalls wird der Start-Zustand nicht verlassen. Ist der Server bereits voll ausgelastet, kann durch eine Aufnahme zusätzlicher VMs die Leistungsaufnahme nicht weiter erhöht werden. Die Aufgabe muss in dem Fall von anderen PMs derselben Gruppe übernommen werden.

Wurde der linke Flügel betreten, wird die Gesamtmenge der im System vorhandenen VMs auf ihre Eignung zum Betrieb auf der PM geprüft. Dabei werden zunächst die VMs ausgeschlossen, die schon auf einer PM betrieben werden, die zur eigenen PM-Gruppe gehört. Es wird eine Liste von entsprechenden VMs aufgebaut und schließlich eine passende VM migriert. Wie die genaue Auswahl der VMs zum Erhöhen und Absenken der Leistungsaufnahme erfolgt, wird in Abschnitt 5.7 detailliert beschrieben. Im Anschluss wird der Start-Zustand wieder erreicht und weiterer Migrationsbedarf geprüft.

Soll eine Leistungsreduzierung erfolgen, wird der entsprechende, rechte Flügel durchlaufen. Auch hier wird zunächst geprüft, ob die Aufnahme weiterer VMs möglich ist. Es wird eine Teilmenge von VMs gebildet und am Ende eine geeignete VM zur Migration ausgewählt.

Das Verfahren terminiert, sobald die vorgegebene Zielgröße erreicht wurde oder keine weiteren VM-Migrationen durchgeführt werden können.

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Nach einer durchgeführten Migration müssen die Auslastungsdaten der abgebenden und die der aufnehmenden PM neu berechnet werden. Die Neuberechnung erfolgt nach Abschluss der Migration, bevor die PM wieder in den Start-Zustand wechselt.

5.5.3. Leistungsaufnahme der PM

Zur Berechnung der aktuellen Leistungsdaten finden in der Simulation einfache Modelle Anwendung. In diversen Untersuchungen wurde gezeigt, dass die CPU-Auslastung maßgeblicher Faktor zur Bestimmung der Leistungsaufnahme eines Servers ist [SPV09, FWB07, MVGN16]. Es existiert eine lineare Abhängigkeit zwischen Leistungsaufnahme und der Server- bzw. CPU-Auslastung [BB10, BAB12, BB12, FWB07, MLL12].

Zur Berechnung der Leistungsaufnahme wird daher in [MLL12] folgende Formel vorgeschlagen:

$$P_j = P_{Idle} + (P_{busy} * U_j) \quad (5.2)$$

Wobei P_j die Leistungsaufnahme und U_j die kontinuierliche Größe der CPU-Auslastung (Utilization) der PM j sind. Effekte durch die nichtlineare Funktionsweise des DVFS werden nicht berücksichtigt.

In der hier vorgestellten Simulation sollen die Leistungsdaten realer Server dargestellt werden. Zu diesem Zweck wurden die Daten des Servers Fujitsu PRIMERGY TX2560 M1 [Sta15] verwendet und so das Verhalten dieses Servers nachgebildet. Die Standard Performance Evaluation Corporation (SPEC) ist eine gemeinnützige Organisation und bietet u.a. den Zugang zu einer Datenbank mit Benchmark-Ergebnissen einer ganzen Reihe von Servern an. Die Ergebnisse werden sowohl grafisch als auch tabellarisch zur Verfügung gestellt.

Server-Auslastung [%]	Leistungsaufnahme [W]
100	264
90	240
80	214
70	185
60	161
50	144
40	129
30	114
20	99,4
10	84,3
Active Idle	40
Off	0

Tabelle 5.1.: Leistungsdaten eines Fujitsu PRIMERGY TX2560 M1 Servers [Sta15]

In Tabelle 5.1 sind die von SPEC bezogenen Daten des Fujitsu PRIMERGY TX2560 M1 dargestellt. Die jeweiligen Werte liegen in einem Raster von 10%-Schritten vor und decken den gesamten Leistungsbereich ab.

Zur Ermittlung der Leistungsaufnahme einer PM in der Simulation wird eine Abbildung der Server-Auslastung auf die jeweilige Leistungsaufnahme durchgeführt und so der entsprechende Wert ermittelt. Für eine genauere Abbildung scheint zunächst die Interpolation der verfügbaren Daten sinnvoll zu sein. Allerdings fußen die erzielbaren Effekte auf der Implementierung der DVFS-Technologie, die nur eine beschränkte Anzahl von Zuständen annehmen kann, wie in [MLL12] und [Bel13] beschrieben. Durch die Interpolation würde folglich eine Granularität dargestellt, die real nicht gegeben ist. Wie auch in [Bel13] umgesetzt, wird daher eine 10-prozentige Granularität verwendet, die ebenfalls zur Datenbasis der von SPEC beziehbaren Werte passend ist.

Effekte auf die Server-Leistungsaufnahme durch VM-Migrationen

Durch die Migration von VMs werden diverse Effekte ausgelöst, zu denen bereits umfangreiche Untersuchungen durchgeführt wurden. Der Fokus liegt bei diesen Untersuchungen hauptsächlich im Bereich der Performance und den Auswirkungen auf SLAs [VBVB09]. Nur wenige wissenschaftliche Arbeiten haben sich mit den Auswirkungen auf die Leistungsaufnahme während einer Live-Migration beschäftigt [VAN08, SD13, HGWQ11].

Wie von Strunk et. al [SD13] beschrieben, wurden Zunahmen in der Leistungsaufnahme der Server während Migrationen von VMs beobachtet. Im konkreten Beispiel lagen sie bei nur wenigen Watt, wie in Abschnitt 2.4.3 dargestellt. Diese Effekte werden in der Modellierung des hier vorgestellten Verfahrens daher nicht abgebildet.

Der beschriebene Effekt fällt sehr gering aus und es ist fraglich, ob und in welchem Umfang die beobachteten Effekte der Migration einer VM zugeschrieben werden können. Bei den Untersuchungen wurden die Leistungsaufnahmen der Server im Idle-Status zugrunde gelegt und mit der Leistungsaufnahme während einer Migration verglichen. Es ist zu vermuten, dass Komponenten des Servers im Idle-Status inaktiv oder in der Leistungsaufnahme reduziert sind, wie es beispielsweise bei der CPU mittels DVFS möglich wäre. Es kann folglich nicht pauschal von einer zusätzlichen Leistungsaufnahme der Server in dem beschriebenen Umfang ausgegangen werden. Die Effekte einer möglichen Erhöhung der Leistungsaufnahme, die durch den Migrationsprozess bedingt wäre, werden daher in dieser Implementierung nicht berücksichtigt. Da die Migrationsphasen der VMs im Modell abgebildet sind, wäre eine gleichzeitige Erhöhung der Leistungsaufnahme gegebenenfalls in der Modellierung darstellbar.

5.5.4. Migrationszeiten

Das vorgestellte Verfahren initiiert Migrationen von VMs, die eine gewisse Zeit in Anspruch nehmen. Ziel des Verfahrens ist es, in möglich kurzer Zeit ein Absenken oder Erhöhen der Leistungsaufnahme zu ermöglichen. Die Zeit-Komponente spielt daher eine wesentliche Rolle. Die Modellierung der Migrationszeiten stützt sich hier auf Untersuchungen der TU Dresden [SD13, RDSM15], wo VMs unterschiedlicher Größe in diversen Szenarien migriert wurden. Es konnte insbesondere beobachtet werden, dass sich bei einer

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Verdopplung der RAM-Größe auch eine Verdopplung der Migrationszeiten ergibt. Für die Ableitung von Zwischenwerten wird in der Simulation daher eine lineare Interpolation verwendet. Die in der Uppaal-Simulation verwendeten Migrationszeiten errechnen sich aus der RAM-Größe der jeweiligen VM und den von Strunk et. al [SD13] gefundenen linearen Effekten zur Migrationsdauer.

5.6. SLM-Komponente

Die Leistungsaufnahme eines Rechenzentrums setzt sich aus mehreren Komponenten zusammen, wie in Abschnitt 2 beschrieben. Die von den Servern ausgehende Abwärme muss über die Klimatisierung abtransportiert werden. Der hierfür nötige Energieaufwand ist wiederum stark von der Außentemperatur und der verwendeten Kühltechnik abhängig. Hier kommt es unter anderem auch zu Latenzen hinsichtlich der Leistungsaufnahme. Während sich eine geänderte Prozessorauslastung direkt auf die momentane Leistungsaufnahme des Servers auswirkt, kann sich dieser Effekt bei der Leistungsaufnahme der Kühltechnik verzögern. Der systematische Zusammenhang muss im konkreten Einzelfall implementiert werden und wird hier nicht weiter betrachtet. Die reduzierte Kühlleistung kann jedoch zeitverzögert in die Modellierung der Server-Leistungsaufnahme einfließen und so berücksichtigt werden.

In einem realen Szenario werden in der SLM-Komponente die Informationen aus dem aktuellen Service-Status, der Leistungsaufnahme und der SLA-Parameter aggregiert und es kann ein Wert für eine ökonomisch sinnvolle Leistungsaufnahme festgesetzt und als Anforderung propagiert werden. Die fachlichen Abhängigkeiten und Details der SLAs sind für das Erzeugen einer gewünschten Leistungsaufnahme irrelevant und stehen in keinem zu berücksichtigenden Zusammenhang. Die implementierte Dummy-SLM-Komponente generiert daher losgelöste Zielvorgaben oder kann eine vorher festgelegte Zielwertfolge vorgeben.

In der Simulation berechnet die SLM-Komponente die aktuellen Kenndaten des Gesamtsystems und gibt neue Werte für die Ziel-Leistungsaufnahme vor. Das SLM umfasst im realen Szenario alle relevanten Services des Rechenzentrums und neben der Leistungsaufnahme als primäre Zielgröße sind auch diverse weitere Anwendungsfälle denkbar, die eine Leistungsanpassung sinnvoll erscheinen lassen. Wartungsarbeiten an Komponenten des Rechenzentrums wie auch unerwartete Ausfälle von Komponenten kommen hier in Betracht. Fällt beispielsweise ein Klimagerät aus, kann die Leistung der Server entsprechend angepasst werden, um teurere Totalausfälle zu vermeiden. Auch diese Aspekte wären über das SLM-System abbildbar, werden in dieser Arbeit jedoch nicht weiter verfolgt.

5.6.1. Test-Komponente

Zur Laufzeit können sich die Ressourcenanforderungen von Applikationen ändern und so das balancierte Gleichgewicht zwischen den antagonistischen Gruppen stören. Hier

kann es zu SLA-relevanten Ereignissen kommen. Um solche Fälle abzubilden, wurde eine Komponente implementiert, die eine definierte Dynamik des Ressourcenbedarfs erzeugt. Die implementierte Test-Komponente simuliert eine Änderung des Ressourcenbedarfs der VMs, mit der geprüft werden kann, ob der Algorithmus angemessen auf solche Fälle reagiert. So können Schwellenwerte einer Implementierung getestet und die Systemstabilität gegenüber Störeinflüssen geprüft werden.

Die Test-Komponente kann bei einer explizit gewählten oder zufällig bestimmten VM die Ressourcen-Daten entsprechend einer Vorgabe manipulieren. Für das Verfahren selbst hat diese Komponente keine weitere Bedeutung.

5.7. Auswahl einer VM und Migrationsstrategie

Die effiziente Platzierung von VMs auf PMs kann mit dem Behälterproblem – wie in 4.4 beschrieben – verglichen werden. Um die Komplexität der Aufgabe zu reduzieren, sollen die möglichen Varianten bei der VM-Auswahl bereits im Vorfeld reduziert werden.

Das in dieser Arbeit vorgestellte Verfahren implementiert ein Gleichgewicht zwischen zwei Servergruppen. Eine Gruppe ist für die mögliche Leistungserhöhung verantwortlich, die andere für das Absenken. Durch diese Aufgabenverteilung reduzieren sich die für eine Migration in Frage kommenden VMs bereits, wie beschrieben. Alle VMs, die auf einer PM mit derselben Aufgabenstellung betrieben werden, scheiden als mögliche Migrationskandidaten aus.

Eine weitere Reduktion der Auswahlmenge erfolgt über eine Prüfung, ob die potentielle VM einen Ressourcenbedarf hat, der von der anvisierten Ziel-PM gedeckt werden kann, und die verbliebenen Ressourcen der PM für einen Betrieb ausreichen. Im Unterschied zur Lösung des Behälterproblems kann hier nicht von leeren Behältern bzw. Servern ausgegangen werden. Auf der PM befinden sich in der Regel bereits VMs und eine einzige weitere VM kann die Kapazitäten der PM übersteigen.

Bei der Ermittlung einer passenden VM gibt es harte und weiche Kriterien, die Berücksichtigung finden. Der Ressourcenbedarf bezüglich des RAM ist beispielsweise ein hartes Kriterium. Wird von einer VM mehr RAM benötigt als auf der Ziel-PM verfügbar ist, kommt die VM für eine Migration nicht in Betracht. Es käme zu dem beschriebenen Effekt des Auslagerns von Speicherinhalten, der zu vermeiden ist.

Sind alle harten Kriterien erfüllt, gibt es in der Menge dieser Kandidaten dennoch VMs, die besser oder schlechter geeignet sind. Bei der Bewertung dieser weichen Kriterien gilt es, den besten Effekt zu erzielen, während grundsätzlich alle diese Kandidaten technisch passend sind.

Bei der Auswahl einer VM kann ein intelligentes Auswahlverfahren seinen Mehrwert nur voll zur Geltung bringen, wenn die Gesamtmenge der VMs hinreichend unterschiedlich ist und sich durch die gezielte Auswahl eines bestimmten Kandidaten entsprechende Vorteile ergeben. Das Maß der Unterschiedlichkeit wird im Folgenden mit dem aus der Soziologie

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

stammenden Begriff der Diversität bezeichnet. Um eine realitätsnahe Abbildung zu erhalten, stammen die in dieser Arbeit verwendeten VM-Profile aus Profildatensätzen der Industrie, diese Datensätze weisen somit einen realitätsnahen Grad an Diversität auf.

Die PMs müssen nun aus der Gesamtmenge der Kandidaten, die die harten Kriterien erfüllen, eine Auswahl treffen. Jeder PM die Gesamtmenge der Kandidaten zu präsentieren, führt allerdings zu Problemen. Analysiert nun jede PM alle diese Kandidaten mit einem deterministischen Verfahren, würden alle parallel arbeitenden PMs denselben VM-Kandidaten wählen. Die Migration dieser VM könnte nur eine einzige PM durchführen, die Analyse durch die übrigen PMs wäre vergebens und auch die Geschwindigkeit der Zielerreichung wäre herabgesetzt, da nur eine Migration durchgeführt würde.

Um diese Problematik zu umgehen, wählt jede PM randomisiert eine VM aus der Gesamtmenge und prüft diese auf die beschriebenen harten Kriterien. Werden die Kriterien erfüllt, wird diese VM in einen Pool aufgenommen. In diesem Pool werden mehrere potentielle Kandidaten gesammelt. Durch diese Vorgehensweise erhält jede PM einen individuellen Pool an VM-Kandidaten. Die Größe des Pools ist abhängig von der Dimensionierung des jeweiligen Szenarios. Bei einem Szenario von 40 VMs und 8 PMs empfiehlt sich eine Poolgröße von 5 VMs. Somit ergibt sich bei 8 PMs und 5 VMs je Pool die Gesamtmenge von 40 VMs. Weisen die betriebenen VMs eine sehr geringe Diversität auf, kann die Poolgröße reduziert werden, hierfür kann jedoch keine allgemeine Empfehlung abgegeben werden.

Die VM-Kandidaten des Pools erfüllen bereits die harten Kriterien. Mit Hilfe von weichen Kriterien wird die Auswahl weiter eingeschränkt, um genau einen VM-Kandidaten zu erhalten.

Folgende Aspekte werden berücksichtigt:

- Die zu erwartende Leistungsänderung durch die Migration soll möglichst hoch sein, um einen großen Effekt je Migration zu erzielen.
- Die VM nutzt die Ressourcen-Lücke auf dem Ziel-Server optimal aus.
- Die VM soll schnell zu migrieren sein.
- Die VMs sollen möglichst wenig migriert werden.

Um die genannten Punkte abdecken zu können, sollen folgende Leitlinien bei der Migration Berücksichtigung finden:

- Ein Aspekt ist der durch eine Migration erzielbare Effekt. Im Bezug auf die Leistungsaufnahme des Servers lässt sich der größte Effekt mit Migrationen von VMs erreichen, deren CPU-Auslastung besonders hoch ist.
- Auch die auf dem Zielhost zur Verfügung stehenden Ressourcen sind zu betrachten. Ideal ist eine VM, die genau zu den anvisierten Ressourcen-Auslastungen passt. Die Anzahl nötiger Migrationen kann so minimiert werden.
- Die Dauer einer Migration hängt maßgeblich von der Größe des RAMs der VM ab. Dieser muss während der Migration übertragen werden, siehe Abschnitt 2.4.2. Um die Migrationsdauer kurz zu halten, sind VMs mit kleinerem RAM-Bedarf gegenüber jenen mit großem RAM-Bedarf zu bevorzugen.

5.7. Auswahl einer VM und Migrationsstrategie

- Da eine PM jeweils nur eine Migration zur Zeit ausführt und der Umschaltzeitpunkt zu einer Serviceunterbrechung führt, ist mit Migrationen sparsam umzugehen. Die Anzahl der Migrationen, die eine VM bereits vollzogen hat, wird daher zur Kontrolle als zusätzlicher Parameter der VM gespeichert. Bei der Auswahl einer VM wird die Anzahl bereits durchgeführter Migrationen entsprechend berücksichtigt.

Für die konkrete Definition der Auswahl-Kriterien ist entscheidend, ob die Leistungsaufnahme erhöht oder abgesenkt werden soll.

5.7.1. Erhöhen der Leistungsaufnahme

Bei hoher Verfügbarkeit von elektrischer Energie kann die Leistungsaufnahme des Rechenzentrums erhöht werden. Die auf den Servern laufenden Applikationen können mit mehr Ressourcen ausgestattet werden, so erhöht sich die Performance der Applikationen und ebenfalls die Leistungsaufnahme der Server.

Steht eine Migration zur Erhöhung der Leistungsaufnahme an, gelten zunächst die Kriterien – wie beschrieben – die mit möglichst wenig Migrationsschritten den größten Effekt erzielen. Hinzu kommt, dass die aufnehmende PM mindestens so viel Kapazitäten freihaben muss, um die betreffende VM ausführen zu können, ohne dass es zu Einbußen in der Performance kommt. Gerade durch erhöhte Performance soll die gesteigerte Leistungsaufnahme erzielt werden.

Zudem muss auf der abgebenden PM eine Ressourcen-Knappheit herrschen, wie eine CPU-Last, die über 100% liegt. So ist gewährleistet, dass eine Migration zu einer Optimierung der Performance und damit auch zu einer Erhöhung der Leistungsaufnahme führt. Bei zwei völlig identischen VMs bezüglich ihres Ressourcenbedarfs ist eine VM, die aktuell auf einer PM mit 120% Auslastung liegt, besser geeignet als eine, die auf einer PM mit 102% Auslastung liegt. Die über 100% hinausgehende CPU-Auslastung entspricht dem Potential zur Erhöhung der Leistungsaufnahme. Es wird in dieser Konstellation die VM bevorzugt, deren aktuelle PM bei 120% Auslastung liegt.

Weiterhin wird geprüft, wie sich das Auslastungsniveau der PM nach einer Migration darstellen würde. Zu vermeiden sind Migrationen, bei denen sich eine Auslastung der CPU von beispielsweise 120% ergäbe. Die über 100% hinausgehende Auslastung wäre nach der Migration ein verlorenes Potential. Die VM würde dann bereits innerhalb der eigenen PM-Gruppe betrieben und stünde anderen PMs für Migrationen nicht mehr zur Verfügung. Bei der VM-Auswahl wird auch dieser Aspekt berücksichtigt, um kein Potential zur weiteren Erhöhung der Leistungsaufnahme zu verlieren.

Auswahl der VM zum Erhöhen der Leistungsaufnahme

Zur Befüllung des VM-Pools wird randomisiert eine VM der antagonistischen Gruppe gewählt und geprüft, ob die RAM-Größe den freien Speicher der PM nicht übersteigt. Weiterhin wird geprüft, ob die abgebende PM eine CPU-Auslastung über 100% aufweist

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

und die Aufnahme dieser VM die CPU-Auslastung der aufnehmenden PM nicht über 100% ansteigen lässt. Sind diese harten Kriterien erfüllt, wird die VM in den Pool genommen.

Um zur Auswahl genau einer VM zu gelangen, werden die weichen Kriterien angewandt. Hierbei ergibt sich ein Spannungsfeld aus mehreren Kriterien. Eine hohe CPU-Auslastung einer VM verspricht einen großen Effekt auf die Leistungsaufnahme. Eine hohe RAM-Belegung bedeutet eine längere Migrationszeit. Hinzu kommt noch die Bewertung der Anzahl von VM-Migrationen. Naheliegender erscheint die Implementierung eines multi-kriteriellen Problems mit Gewichtungsfaktoren oder die Suche nach einem Pareto-Optimum. Letzteres liefert allerdings eine Menge an VMs, eine automatisierte Entscheidung für eine VM ist mit dieser Technik nicht möglich.

Betrachtet man die Parameter *CPU-Auslastung* und *RAM-Größe*, scheint der Quotient beider Parameter eine valide Messgröße zur Ermittlung des VM-Kandidaten. Der Aufwand einer Migration liegt im Transfer des RAM und somit erscheint eine VM, die eine hohe CPU-Auslastung und geringe RAM-Größe aufweist besonders geeignet. Nachteilig bei einer derartigen Auswahl ist, dass die so gefundene VM eine sehr kleine CPU-Auslastung haben kann. Man hätte hier zwar pro Megabyte migriertem RAM den höchsten Effekt auf die Leistungsaufnahme, es können jedoch insgesamt mehr Migrationen mit dazugehörigem Auswahlprozess erforderlich werden.

Ein Blick auf die Anforderungen offenbart, dass eine deutliche Reaktion auf die geforderte Leistungsänderung erkennbar werden muss (siehe Abschnitt 4.6). Es müssen folglich in den ersten Migrationsschritten die VMs mit der höchsten CPU-Auslastung migriert werden, um diese Anforderung abzudecken. Hinzu kommt die Leistungsaufnahme der Support-Infrastruktur, wie der Klimatisierung, die sich erst verzögert an die geänderte Wärme-Abgabe der Server adaptiert. Andere Faktoren müssen daher eine untergeordnete Rolle spielen. Ein weiterer Punkt, der für die Auswahl einer VM mit möglichst hoher CPU-Auslastung spricht, ist die Annäherung an die Ziel-Leistungsaufnahme und eine optimale Ausnutzung der Ressourcen der PM. Idealerweise kann eine Näherung der Ziel-Leistungsaufnahme zunächst mit großen Schritten (hohe CPU-Auslastung) angegangen werden und die feingranulare Ziel-Erreichung durch VMs mit kleinerer CPU-Auslastung erfolgen. Der Fokus bei der VM-Auswahl liegt damit eindeutig auf der CPU-Auslastung und die Implementierung eines multi-kriteriellen Problems mit Gewichtungsfaktoren ist nicht erforderlich.

Ist der Pool mit VM-Kandidaten vollständig gefüllt, wird er absteigend nach CPU-Auslastung sortiert. Die VMs mit der höchsten CPU-Auslastung haben dabei den besten Migrationseffekt und nutzen die Ressourcenlücke optimal aus. Es werden nun ausschließlich die beiden VMs mit der höchsten CPU-Auslastung bezüglich des RAM verglichen und die VM mit dem kleineren RAM für eine schnelle Migration gewählt. Zeigt diese VM jedoch einen höheren Migrationszähler wird doch auf die VM mit größerem RAM zurückgegriffen. Es erfolgt eine definitive Auswahl aus den beiden VMs mit der höchsten CPU-Auslastung. Mit dieser Vorgehensweise kann sichergestellt werden, dass die CPU-Auslastung die höchste Priorität bei der VM-Auswahl hat, der Migrationszähler und die RAM-Größe aber dennoch Berücksichtigung finden.

Damit ist die VM zur Erhöhung der Leistungsaufnahme gefunden und kann migriert

werden. Die harten Kriterien zur Auswahl einer VM zum Absenken der Leistungsaufnahme unterscheiden sich jedoch in einigen Details.

5.7.2. Absenken der Leistungsaufnahme

Bei niedriger Energie-Verfügbarkeit, soll ein Absenken der Leistungsaufnahme ermöglicht werden. Dies kann über die Konsolidierung von Ressourcen erfolgen, die Applikationen haben somit weniger Ressourcen zur Verfügung, die Leistungsaufnahme der Server sinkt, ebenso wie die Performance der Applikationen.

Grundsätzlich gelten für die Ermittlung einer geeigneten VM zur Erzeugung des beschriebenen Effekts dieselben allgemeinen Kriterien, wie bereits beschrieben. Bei der weiteren Auswahl spielt die momentane CPU-Auslastung der abgebenden PM in diesem Fall jedoch keine Rolle. Um eine überproportionale Auslastung der aufnehmenden PM zu verhindern, wird geprüft, ob die zusätzliche VM die CPU-Auslastung über 200% heben würde. Wäre dies der Fall, wird die VM verworfen. Ziel ist es, die Auslastung der abgebenden PM generell zu reduzieren. Weitere Effekte wie im Fall der Erhöhung der Leistungsaufnahme sind nicht zu berücksichtigen, es findet die in Abschnitt 4.1.1 beschriebene Ressourcen-Überprovisionierung statt. Im Bezug auf die Server-Ressourcen wird einzig geprüft, ob der RAM-Bedarf von der aufnehmenden PM gedeckt werden kann, um den beschriebenen Effizienzverlust bei Überprovisionierung des RAM zu vermeiden. Ausgeklammert werden VMs generell, bei denen das beschriebene Urgent-Flag gesetzt wurde. Darüber hinaus gelten die beschriebenen Leitlinien für eine schnelle und effektive Migration.

Auswahl der VM zum Absenken der Leistungsaufnahme

Zur Befüllung des VM-Pools wird randomisiert eine VM der antagonistischen Gruppe gewählt und geprüft, ob die RAM-Größe den freien Speicher der PM nicht übersteigt, kein Urgent-Flag gesetzt wurde und die CPU-Auslastung der PM mit dieser VM unter 200% bleibt.

Ist der Pool vollständig gefüllt, wird er absteigend nach CPU-Auslastung sortiert. Die VMs mit der höchsten CPU-Auslastung haben dabei den besten Migrationseffekt. Es werden nun ausschließlich die beiden VMs mit der höchsten CPU-Auslastung bezüglich des RAM verglichen und die VM mit dem kleineren RAM für eine schnelle Migration gewählt. Zeigt diese VM jedoch einen höheren Migrationszähler wird doch auf die VM mit größerem RAM zurückgegriffen.

Sind alle VMs einer PM migriert und die PM frei geworden, kann diese PM abgeschaltet werden. Mit dieser Maßnahme lässt sich die Leistungsaufnahme weiter reduzieren. Bei diesem Schritt ist jedoch eine Verzögerung im Falle einer späteren Leistungserhöhung zu berücksichtigen. Soll die Leistungsaufnahme wieder erhöht werden, muss dieser Server zunächst das Betriebssystem hochfahren, bevor er für die Aufnahme von VMs zur Verfügung steht. In der Simulation sind beide Varianten (mit PMs im Status „active Idle“ oder „off“) möglich. In der Praxis kann die Entscheidung über eine Abschaltung der Server

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

über das SLM ermittelt werden. Die Latenzen durch das Hoch- und Herunterfahren der Server müssen in Bezug auf die SLAs der Applikationen und der Reaktionszeiten für eine Änderung der Leistungsaufnahme berücksichtigt werden.

Die in diesem Abschnitt vorgestellte die heuristische Arbeitsweise und die architektonischen Aspekte führen zu einer Reduktion der Komplexität des Verfahrens, wie in Forschungsfrage Nr. 4 gefordert.

5.8. Flussdiagramm zum OHMA-Verfahren

In Abbildung 5.14 ist der Ablauf des Verfahrens in einem Flussdiagramm dargestellt. Der Einstiegspunkt ist die Übermittlung eines neuen Zielwertes und die Prüfung, ob Handlungsbedarf besteht. Bedarf es einer Anpassung der Leistungsaufnahme, werden im Anschluss fallweise die beiden Stränge zur VM-Auswahl durchlaufen und eine VM-Migration vorgenommen.

5.9. Kenngrößen, Dimensionierung und Konfiguration

Bezüglich der Implementierung des Verfahrens ergeben sich gewissen Kenngrößen, deren Bedeutung und Festlegung in diesem Abschnitt kurz beschrieben werden. Weiterhin wird die Frage nach einer geeigneten Dimensionierung und Konfiguration beantwortet.

5.9.1. Leistungsreduzierung

In der Modellierung wurde für die Konfiguration eine 50-prozentige Leistungsreduzierung gewählt. Diese Entscheidung stützt sich auf Beispiele aus der Praxis, wo Applikationen mit hohen Ansprüchen bezüglich der Verfügbarkeit häufig in einem Verbund aus zwei oder mehr Servern betrieben werden. Im Falle eines Hardware- oder Standort-Ausfalls können die Applikationen auf die verbliebenen Server geschwenkt und weiter betrieben werden. Die Performance mag sich dabei allerdings verringern. Die Server sind in der Regel jedoch so ausgestattet, dass zumindest ausreichend RAM zu Verfügung steht, um einen intermittierenden Betrieb zu verhindern. Bei einer zu diesem Szenario passenden Worst-Case-Abschätzung ergäbe sich damit eine Leistungsreduzierung von 50%. Technologisch ist eine größere Leistungsreduzierung möglich und kann über die Konfiguration der zwei Gruppen abgebildet werden.

5.9.2. Aufteilung der Server in Gruppen

Wie bereits beschrieben, werden die Server in zwei Gruppen aufgeteilt. Dies geschieht zum einen, um die Kompetenzen für das Absenken und Erhöhen der Leistungsaufnahme

5.9. Kenngrößen, Dimensionierung und Konfiguration

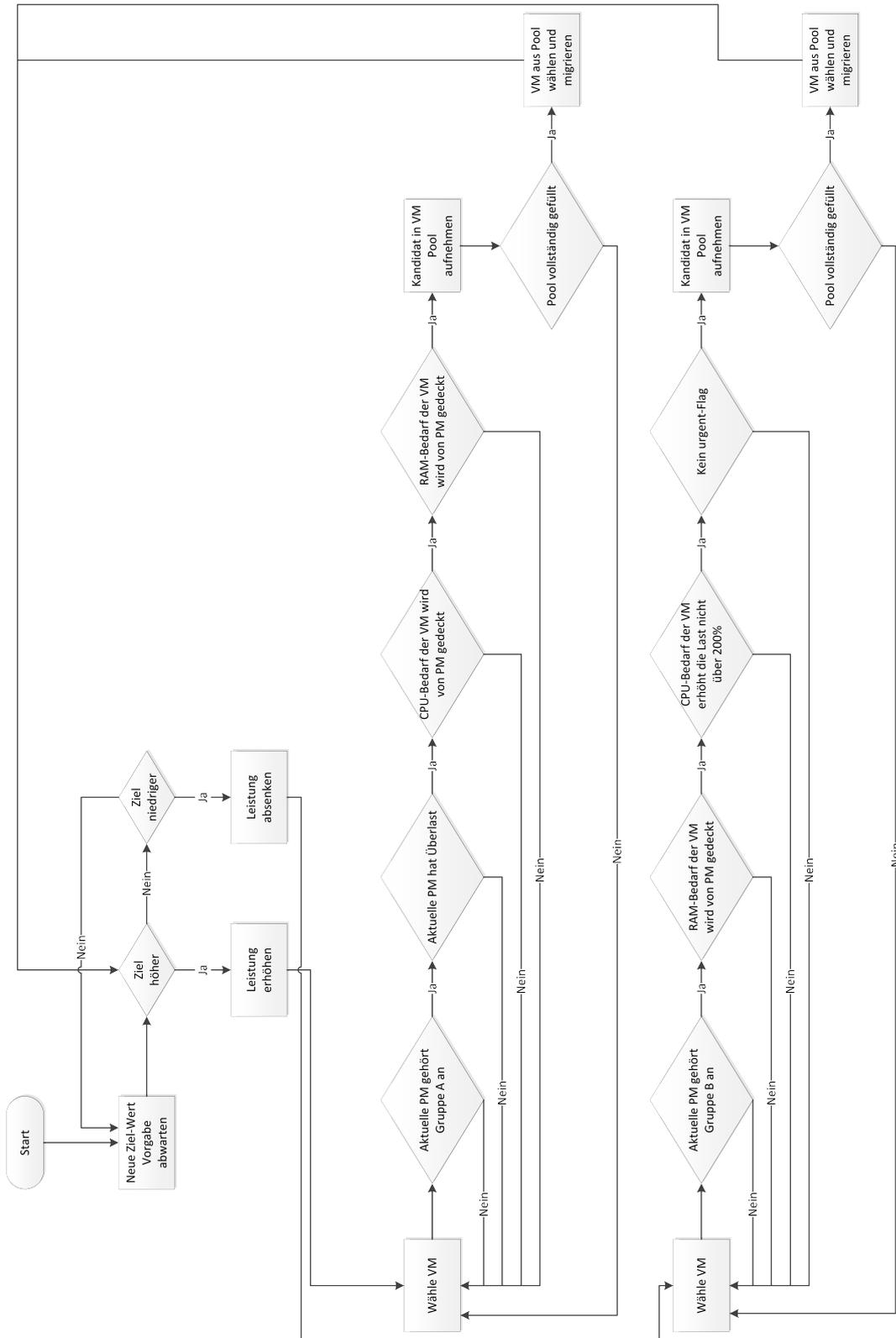


Abbildung 5.14.: Darstellung der Prozessschritte des OHMA-Verfahrens in einem Flussdiagramm

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

festzulegen, und zum anderen, um gegebenenfalls die Hardwareressourcen der Aufgabe entsprechend vorzuhalten. In einem Hochverfügbarkeits-Szenario werden Server oftmals so dimensioniert, dass sie im Störfall weitere VMs aufnehmen können. In dem vorgestellten Verfahren ist der Arbeitsspeicher von Bedeutung und entsprechende Kapazitäten sind von den Servern der Gruppe mit Kompetenz für das Absenken der Leistungsaufnahme vorzuhalten. So ist die Voraussetzung für eine CPU-Überprovisionierung gegeben.

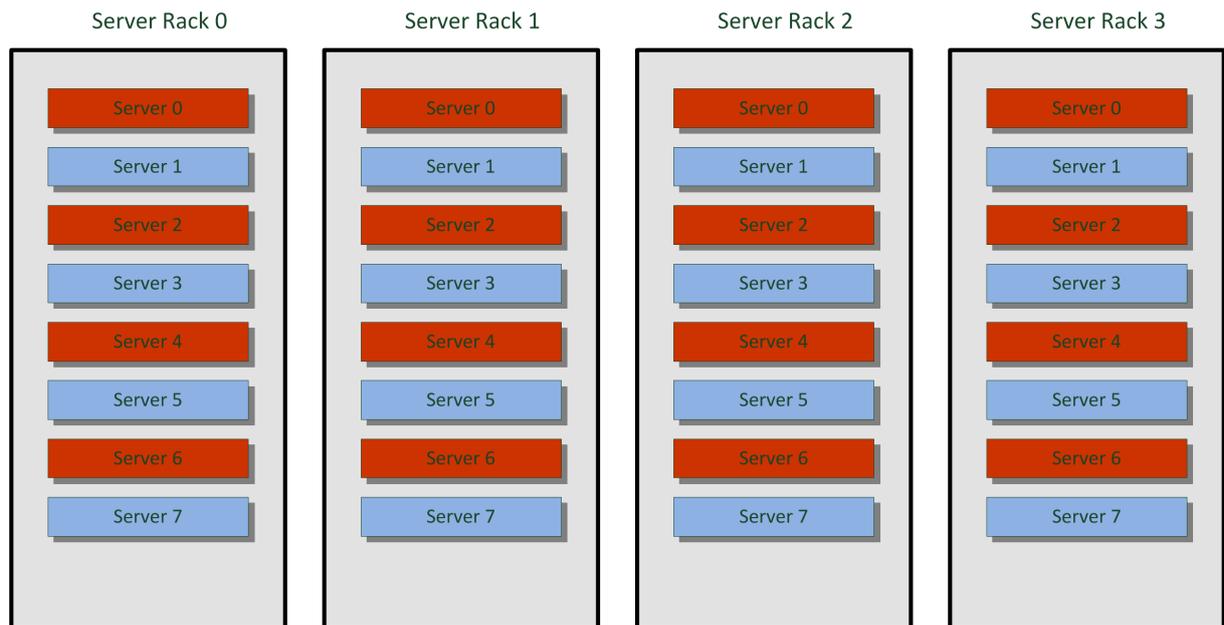


Abbildung 5.15.: Alternierende Aufteilung der Servergruppen

Wie in [TGV08] beschrieben, ist für eine effiziente Klimatisierung eine Gleichverteilung der Wärme und Vermeidung von Hotspots erforderlich. Durch die alternierende Installation der Server im Rack, wie in Abbildung 5.15 dargestellt, wird eine gleichmäßige Verteilung der Abwärme sichergestellt und der Entstehung von Hotspots entgegengewirkt.

Die Zuteilung der Gruppen kann auch unter Berücksichtigung weiterer Gesichtspunkte erfolgen. Lassen sich die Server-Racks bezüglich der Klimatisierung individuell regeln, empfiehlt sich eine Aufteilung, wie in Abbildung 5.16 dargestellt. Hier sind die Server, die derselben Gruppe angehören, in einem Rack zusammen verbaut. Die Klimatisierung kann damit speziell auf die Anforderungen dieser Servergruppe reagieren. Die Klimatisierung kann durch die Aufteilung in Gruppen effizient betrieben werden und stellt einen Baustein bezüglich der Berücksichtigung von Effizienz-Gesichtspunkten dar, wie in Forschungsfrage Nr. 5 gefordert.

Ein weiterer Aspekt für die Implementierung der Gruppen liegt in der Abdeckung von nicht-funktionalen Anforderungen aus Abschnitt 4.6.2 begründet. Durch die Delegation der Leistungsanpassung an eine Gruppe von PMs ist gewährleistet, dass auch bei einem Ausfall einzelner PMs die Anforderung weiterhin bearbeitet und die geforderte Leistungsaufnahme erreicht wird.

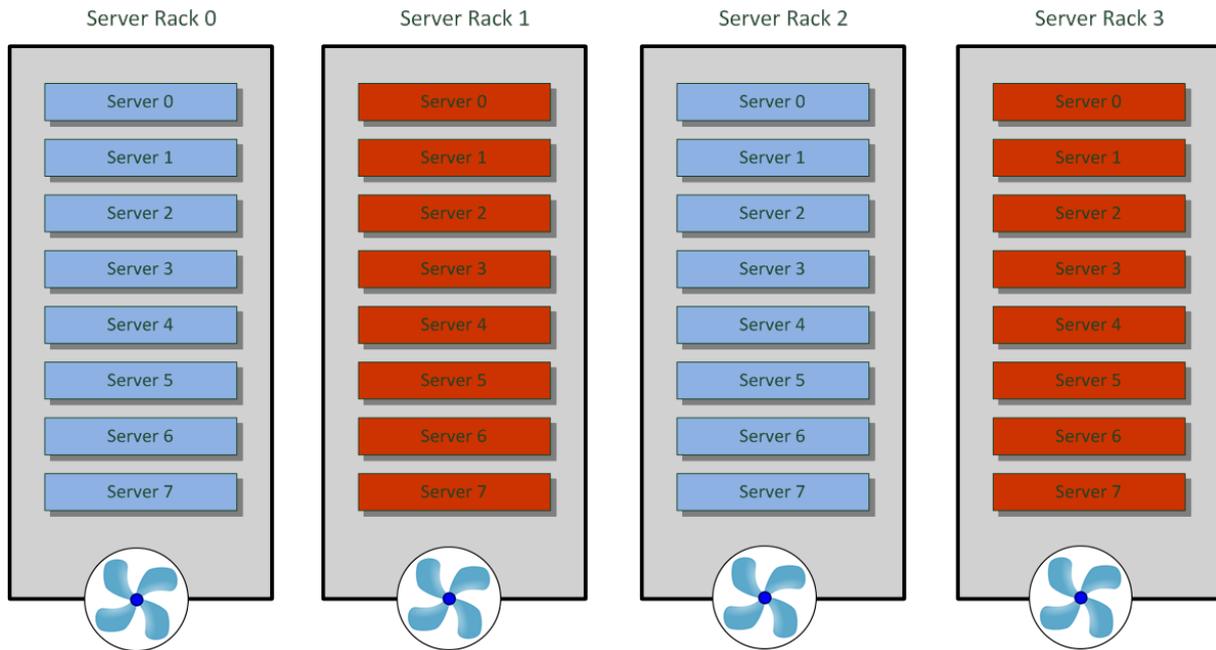


Abbildung 5.16.: Aufteilung der Servergruppen bei individueller Klimatisierung der Racks

5.9.3. Konvergenzverhalten

Nach der Übermittlung einer Anforderung zur Änderung der Leistungsaufnahme wird die Umsetzung sofort verfolgt. Es wird keine vorherige, komplette Evaluation durchgeführt, ob das gestellte Ziel erreichbar ist. Die Anforderung lässt sich in zwei Teilaspekte gliedern: Generell wird ein Erhöhen oder Absenken der Leistungsaufnahme gewünscht und in zweiter Instanz soll eine bestimmte Zielgröße erreicht werden. Es wird davon ausgegangen, dass ein Mehrwert durch das Verfahren bereits mit der Erbringung von Teilergebnissen bezüglich der gewünschten Tendenz erbracht werden kann.

Zum Erreichen der geforderten Leistungsaufnahme werden eine Reihe von VM-Migrationen durchgeführt. Mit jeder einzelnen Migration verringert sich die Diskrepanz zwischen aktueller und geforderter Leistungsaufnahme. Der Abstand zur geforderten Leistungsaufnahme kann dabei beliebig klein werden und die Wahrscheinlichkeit, eine VM zu finden, die schließlich genau zum Erreichen der geforderten Leistungsaufnahme führt, hängt maßgeblich von der Diversität der verfügbaren VMs ab. Weiterhin arbeiten mehrere PMs parallel an der Zielerreichung, sodass es zu typischen Effekten einer Hysterese kommt. Bereits angestoßene Migrationen erzielen ihren Effekt auf die Leistungsaufnahme erst nach Abschluss der Migration, sodass es hier ebenfalls zu Verzögerungen kommt.

Ohne entsprechende Vorkehrungen erreicht das System keinen stabilen Zustand, da es permanent zur Über- oder Untererfüllung der Ziel-Leistungsaufnahme kommt. Um diesen Effekten entgegenzuwirken, wird üblicherweise ein Toleranzbereich um die Ziel-Leistungsaufnahme gebildet. Das Ziel gilt nunmehr als erreicht, sobald sich die Leistungsaufnahme in einem Intervall um die geforderte Leistungsaufnahme bewegt. Der Toleranzwert, der

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

das Intervall um die Ziel-Leistungsaufnahme bildet, wird im Folgenden als Hysteresewert bezeichnet. Um einen geeigneten Hysteresewert zu bestimmen, sind einige Rahmenbedingungen zu berücksichtigen.

Grundsätzlich stellt der Hysteresewert eine erlaubte Abweichung von der gestellten Ziel-Leistungsaufnahme dar und reduziert damit unmittelbar die Genauigkeit des Verfahrens. Der Hysteresewert ist folglich mit Bedacht und möglichst klein zu wählen, um die Genauigkeit des Verfahrens nicht unnötig zu kompromittieren. Bei der konkreten Auswahl eines Wertes sind folgende Faktoren ausschlaggebend:

- *Zusammensetzung der VMs bezüglich Diversität.* Die betriebenen VMs haben Leistungsprofile, über deren Kombination eine bestimmte Leistungsaufnahme erreicht werden soll. Bei einer hohen Diversität der VMs kann sich dem Ergebnis zunächst mit der Migration von VMs mit hoher CPU-Auslastung genähert werden und die feingranulare Näherung an die Zielvorgabe mit VMs geringerer CPU-Auslastung erfolgen. Entsprechend der Beschaffenheit der betriebenen VMs ist diese Näherung bis zu einem gewissen Grad möglich.
- *Leistungsaufnahme der Server und die Implementierung des DVFS.* Die Implementierung des DVFS der PMs ist ebenfalls zu betrachten. Wie beschrieben, wird die Serverauslastung häufig diskret auf die Leistungsaufnahme abgebildet und nicht stetig. Bei einer theoretischen Serverauslastung, die zur Leistungsaufnahme passend wäre, kann die Granularität des DVFS zu grob sein und einen entsprechenden Hysteresewert bedingen.
- *Dimensionierung des Systems.* Die PMs einer Gruppe arbeiten parallel an der Erreichung der Ziel-Leistungsaufnahme. Es kann folglich zu dem Effekt kommen, dass eine bereits initiierte Migration zur weiteren Senkung oder Erhöhung der Leistungsaufnahme führt, obwohl die Ziel-Leistungsaufnahme bereits während der Migration erreicht wurde.

In der Kombination dieser Faktoren muss betrachtet werden, ob eine stetige Abbildung der Leistungsaufnahme möglich ist. Eine zur geforderten Leistungsaufnahme passende Serverauslastung muss durch die Kombination der VMs herstellbar sein und letztlich diese Leistungsaufnahme durch das DVFS abgebildet werden. Kommt es hier zu Abweichungen, muss der Hysteresewert passend gewählt werden. Ohne Hysteresewert würde die Ziel-Leistungsaufnahme permanent über- oder unterschritten und VM-Migrationen zum Absenken bzw. Erhöhen der Leistungsaufnahme initiiert. Der zu wählende Hysteresewert wird maßgeblich durch die genannten Faktoren beeinflusst und kann daher nicht allgemein bestimmt werden. Insbesondere sind die Diversität der betriebenen VMs und die daraus resultierenden Allokationen ausschlaggebend. Um einen geeigneten Hysteresewert zu bestimmen, eignet sich eine heuristische Herangehensweise.

Das Ziel bei der Verwendung des Hysteresewertes ist es, weitere Migrationsprozesse zu unterbinden, sobald sich die Leistungsaufnahme der Ziel-Leistungsaufnahme angenähert hat, und insbesondere Migrationen zu verhindern, die eine gegenläufige Änderung der Leistungsaufnahme bewirken. Hierzu wird die Leistungsaufnahme im Verlauf der Allokationsfolgen betrachtet. Sobald es zu einer gegenläufigen Änderung der Leistungsaufnahme

5.9. Kenngrößen, Dimensionierung und Konfiguration

kommt, ist die Leistungsaufnahme gefunden, die zum Überschreiten der Ziel-Leistungsaufnahme geführt hat, während die Leistungsaufnahme der vorhergehenden Allokation den Zielwert noch nicht erreicht. Die Hysterese ist daher so zu dimensionieren, dass die Leistungsaufnahmen beider Vorgänger-Allokationen in den Korridor der Hysterese fallen.

Zur Bestimmung des Hysteresewertes werden im Vorfeld des realen Betriebes Simulationen durchgeführt. Sind die angestrebten Ziel-Leistungsaufnahmen bekannt, können diese Szenarien zur Bestimmung herangezogen werden. Andernfalls sind Datensätze mit einem Raster von 10%-Schritten bezüglich der Ziel-Leistungsaufnahmen für das Absenken und Erhöhen zu wählen, um auch die Effekte des DVFS abdecken zu können und einen generellen Hysteresewert für das Szenario zu bestimmen.

Für diese Datensätze kann die Allokationsfolge analysiert werden, dies wird durch Pseudocode 1 beschrieben.

Die Hysterese wird ausgehend von einem festen Szenario von VMs und PMs bestimmt. Variabel sind die jeweiligen Ziel-Allokationen und die auf den Servern durch das DVFS erzeugten Leistungsaufnahmen. Um die variablen Effekte weitgehend abzudecken, werden je Szenario mehrere ($N > 1$) Simulationen durchgeführt.

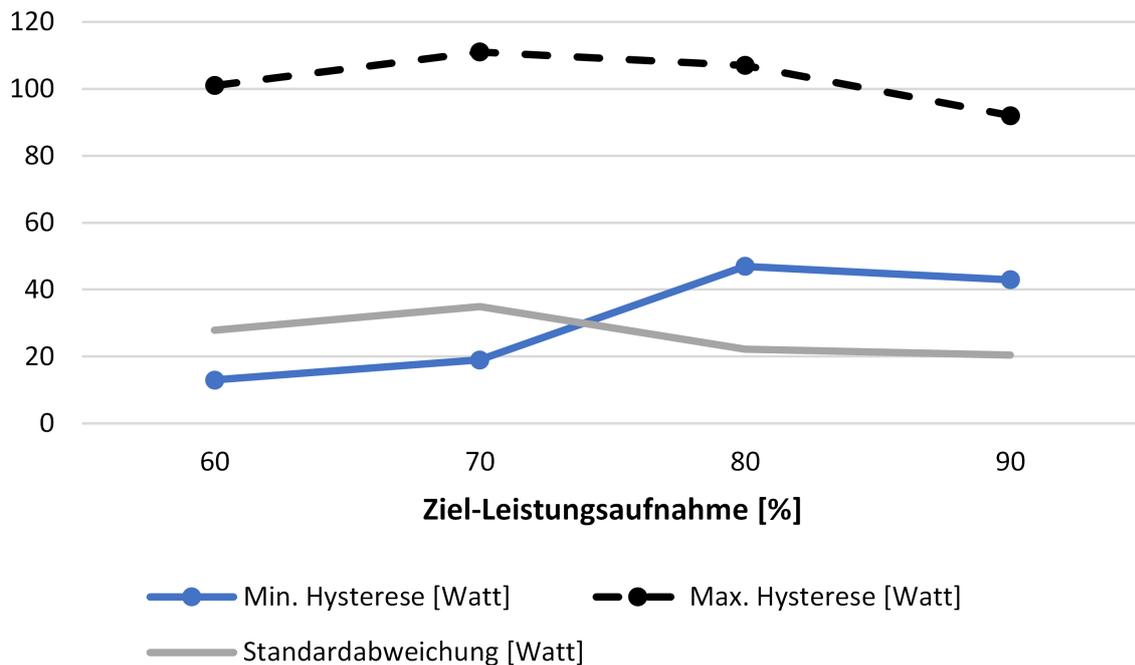


Abbildung 5.17.: Ergebnisse aus der Bestimmung eines Hysteresewertes für ein konkretes Szenario

Aus den so erhaltenen Hysteresewerten ist das Maximum zu wählen, um einem generell verwendbaren Hysteresewert für dieses Szenario zu erhalten. Im Worst Case wird die Hysterese durch diejenige VM mit dem höchsten CPU-Bedarf bestimmt. Die Migration eben dieser VM könnte initiiert sein, während die Ziel-Leistungsaufnahme bereits annähernd

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Pseudocode 1: Heuristischer Algorithmus zur Bestimmung eines Hysteresewertes

Input : N Allokationsfolgen: $AF[N] = \{(\mathcal{A}_1, \dots, \mathcal{A}_j), \dots, (\mathcal{A}_1, \dots, \mathcal{A}_k)\}$,
 Ziel-Leistungsaufnahme: t ,
 Leistungsaufnahme unter Allokation i : $P(\mathcal{A}_i)$

Output: Hysteresewert: h
 Max. Hysteresewert: h_{max}

```

1  $h \leftarrow 0; h_{max} \leftarrow 0;$ 
2 for  $a \leftarrow 1 \leq N$  do
3    $(\mathcal{A}_1, \dots, \mathcal{A}_n) \leftarrow AF[a];$ 
4   if  $t < P(\mathcal{A}_1)$  then
5     for  $i \leftarrow 1 \leq n$  do
6       if  $P(\mathcal{A}_i) \leq P(\mathcal{A}_{i+1})$  then
7          $i ++;$ 
8       else
9          $h_{max} \leftarrow \max(|t - P(\mathcal{A}_{i-1})|, |t - P(\mathcal{A}_{i-2})|, h_{max});$ 
10         $h \leftarrow h + \max(|t - P(\mathcal{A}_{i-1})|, |t - P(\mathcal{A}_{i-2})|);$ 
11      end
12    end
13  else
14    for  $i \leftarrow 1 \leq n$  do
15      if  $P(\mathcal{A}_i) \geq P(\mathcal{A}_{i+1})$  then
16         $i ++;$ 
17      else
18         $h_{max} \leftarrow \max(|t - P(\mathcal{A}_{i-1})|, |t - P(\mathcal{A}_{i-2})|, h_{max});$ 
19         $h \leftarrow h + \max(|t - P(\mathcal{A}_{i-1})|, |t - P(\mathcal{A}_{i-2})|);$ 
20      end
21    end
22  end
23   $a ++;$ 
24 end
25 return  $(h \leftarrow h \div N, h_{max});$ 

```

5.9. Kenngrößen, Dimensionierung und Konfiguration

erreicht wurde. In diesem Fall würde der durch die Hysterese aufgespannte Ziel-Korridor die Leistungsänderung durch diese Migration abdecken. Um die Worst-Case-Fälle zu relativieren, wird der Mittelwert der gefundenen Maximalwerte als Hysteresewert für das untersuchte Szenario verwendet.

Für ein konkretes Szenario mit 40 VMs und 8 PMs wurden Simulationen zur Bestimmung des Hysteresewertes durchgeführt. In Abbildung 5.17 sind die Ergebnisse bei einer Ziel-Leistungsaufnahme von 60%, 70%, 80% und 90% der Gesamtleistungsaufnahme mit jeweils 5 Simulationen dargestellt. Je Ziel-Leistungsaufnahme wurden die gefundenen Minima, Maxima und die Standardabweichung aufgetragen. Die dargestellte Standardabweichung und die gefundenen Minimum-Werte zeigen die Streuung der Ergebnisse in den jeweiligen Szenarien auf. Bei dem Maximal-Wert, der hier von Interesse ist, zeigen sich jedoch nur geringe Abweichungen. Die gefundenen Werte liegen in einem Bereich von 92-111 Watt, der Mittelwert liegt bei 102 Watt. Der für dieses konkrete Szenario bestimmte Hysteresewert liegt damit bei 102 Watt bzw. 5% der maximalen Systemleistung von 2112 Watt.

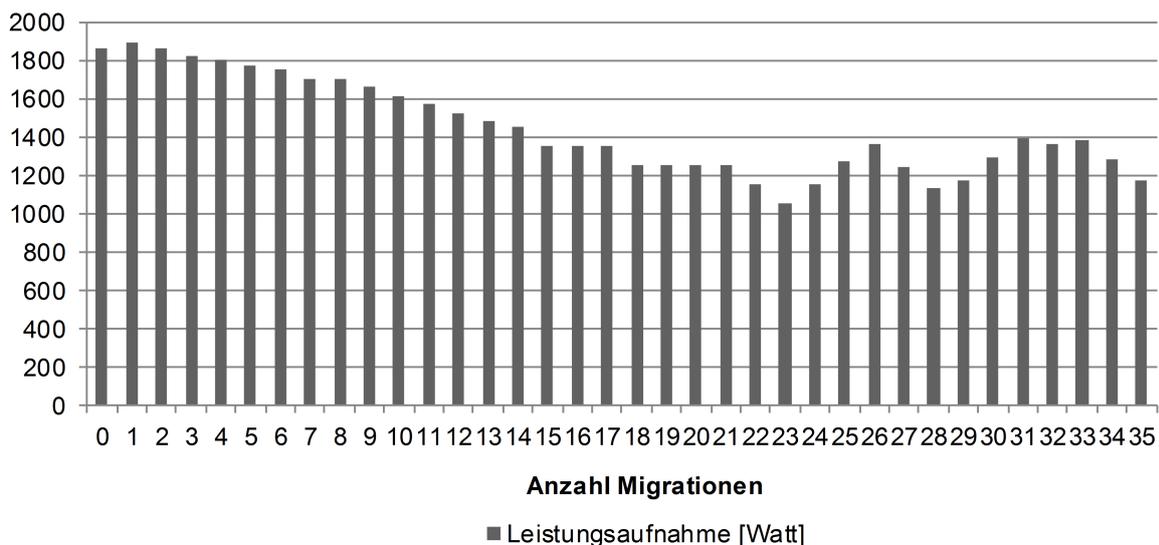


Abbildung 5.18.: Konvergenzverhalten bei zu klein gewählter Hysterese

Zur Demonstration der Funktionsweise und den Auswirkungen des Hysteresewertes wurden Simulationen auf Basis von Daten aus der Industrie mit 40 VMs und 8 PMs durchgeführt. In Abbildung 5.18 ist das Konvergenzverhalten des Systems mit einer Ziel-Leistungsaufnahme von 1200 Watt und einer Hysterese von 50 Watt (entsprechend 2,5% der Gesamtleistung) dargestellt.

Erkennbar ist, dass nach 22 Migrationen eine Näherung auf 1155 Watt erreicht wurde. Bereits initiierte Migrationen senken die Leistungsaufnahme jedoch weiter auf 1056 Watt ab und die Ziel-Leistungsaufnahme unter Berücksichtigung der Hysterese wird unterschritten. In der Folge wird die Leistungsaufnahme wieder angehoben und schließlich die vom System akzeptierte Leistungsaufnahme von 1170 Watt erreicht. Hierfür waren 13 weitere

5. Modellierung des Verfahrens: Online Heuristic Machine Allocation (OHMA)

Migrationen nötig, wobei die erreichte Effektivität nicht im angemessenen Verhältnis zum Aufwand steht.

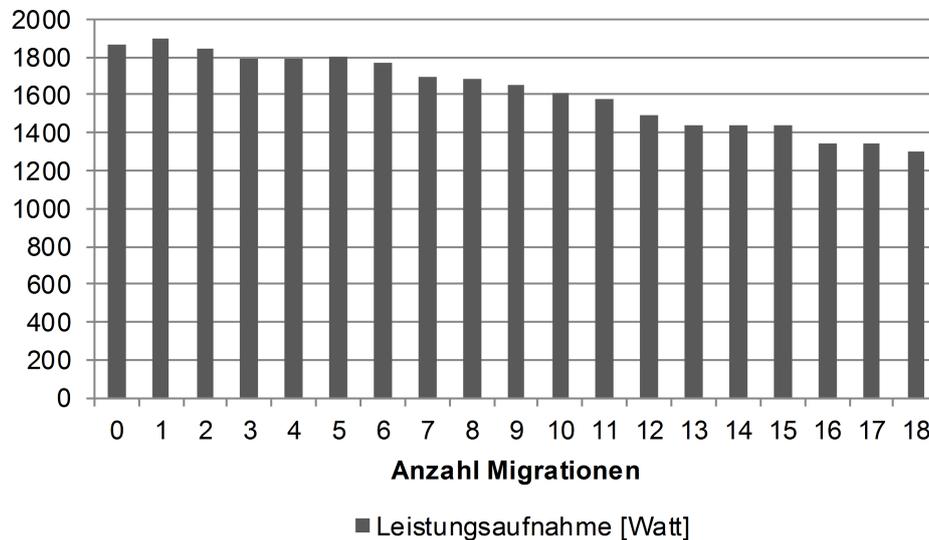


Abbildung 5.19.: Konvergenzverhalten bei passendem Hysteresewert

Mit der beschriebenen Heuristik wurde für dieses Szenario eine Hysterese von 5% ermittelt. Bei einer Hysterese von 5% zeigt sich ein völlig anderes Konvergenzverhalten, wie Abbildung 5.19 darstellt. Nach der Durchführung von 18 Migrationen wird eine Leistungsaufnahme im Bereich der Hysterese erreicht und es werden keine weiteren Migrationen veranlasst. Für den hier verwendeten VM-Datensatz konnte somit ein passender Hysteresewert von 5% der Gesamtleistung ermittelt werden.

In dem konkreten Szenario ergab sich bei der Hysterese von 2,5% eine Leistungsaufnahme von 1170 Watt, 35 Migrationen waren hierzu nötig. Es wurde eine durchschnittliche Leistungsänderung von 19,9 Watt je Migration erbracht. Bei einer gewählten Hysterese von 5% wurden 1299 Watt erreicht und eine Leistungsänderung von 31,5 Watt je Migration. Im Vergleich der beiden Szenarien zeigt das Szenario mit einer nach der Heuristik bestimmten Hysterese eine deutlich bessere Effizienz. Die Auswahl einer passenden Hysterese stellt damit einen wichtigen Konfigurationsparameter dar, er hat direkten Einfluss auf die Dauer bis zum Erreichen eines stabilen Zustands und auch auf die Genauigkeit der Zielerreichung. Die Definition des Hysteresewertes wirkt sich positiv auf die Effizienz des Verfahrens aus, bedingt jedoch Abstriche in der Effektivität.

5.10. Zusammenfassung

In diesem Abschnitt wurde zunächst die allgemeine Funktionsweise des OHMA-Verfahrens dargelegt und die gewählte Systemarchitektur beschrieben. Die Systemarchitektur ist von

diversen Anforderungen geprägt, insbesondere ist der verteilte Ansatz und die Teilung der PMs in zwei Gruppen zu nennen.

In Abschnitt 5.2 wurden die technischen und fachlichen Voraussetzungen benannt, die für die Anwendbarkeit des Verfahrens nötig sind. Der verwendete Server-Typ muss beispielsweise über DVFS-Technologie verfügen und es muss eine Virtualisierungsumgebung zur Verfügung stehen. Das OHMA-Verfahren eignet sich aus fachlicher Sicht für Applikationsanteile, die nicht an direkten User-Interaktionen beteiligt sind und keinen Realzeitanforderungen unterliegen.

Die Modellierung des Verfahrens wurde in dem Simulationsframework Uppaal vorgenommen. Die Entscheidung wurde getroffen, da bestehende Verfahren das intendierte Überprovisionieren von Serverressourcen nicht unterstützen. Im Vordergrund steht beim OHMA-Verfahren das Scheduling von VM-Migrationen. Das Uppaal-Framework wird in diversen weiteren Bereichen der Informatik zum Modellieren von Scheduling-Problemen eingesetzt und ist ein etabliertes Werkzeug.

In Abschnitt 5.3 wurde das Uppaal-Framework und die Verwendung der Automaten-Modelle kurz vorgestellt. In den darauf folgenden Abschnitten wurden die implementierten Automaten-Templates zur Instantiierung des System beschrieben und ihre Funktionsweise erläutert. Der Prozess zur Auswahl einer VM für eine Migration wurde detailliert dargestellt, ebenso die kaskadierenden Schritte zur Reduktion der Komplexität des Problems.

In Abschnitt 5.9 wurden technische Details und Hintergründe erläutert, die die Implementierung beeinflusst haben, so ergibt sich in der Standard-Konfiguration eine Leistungsreduzierung von bis zu 50% und die PMs sind in zwei Gruppen mit klar getrennten Kompetenzen zum Erhöhen und Absenken der Leistungsaufnahme aufgeteilt.

Schließlich wurde das Konvergenzverhalten des Systems betrachtet und die Notwendigkeit eines adäquaten Hysterese-Wertes herausgestellt. Der Einfluss und die Auswirkungen des Hysterese-Wertes wurden diskutiert und die Faktoren benannt, die eine Dimensionierung des Wertes beeinflussen.

6. Potentialanalyse und Evaluation

Die mit Hilfe des OHMA-Verfahrens zu erzielenden Effekte werden in diesem Abschnitt dargestellt. Hierzu werden Ergebnisse aus Simulationen unterschiedlicher Szenarien präsentiert und diskutiert.

Der Kern des Verfahrens besteht in einer Reduzierung der Komplexität für die gestellte Aufgabe. Eine Darstellung der Komplexität des OHMA-Verfahrens erfolgt in Abschnitt 6.2. Mit Simulationen zur Berechnungsdauer werden die Behauptungen bezüglich der Komplexitätsklasse geprüft.

Um die Effizienz und Effektivität des Verfahrens zu beurteilen, werden in Abschnitt 6.3 zunächst Metriken hierfür definiert. Die anschließende Bewertung wird auf Basis der Daten von sechs Szenarien mit jeweils zehn Einzel-Simulationen vorgenommen.

Das heuristische OHMA-Verfahren arbeitet mit erheblich reduzierter Berechnungskomplexität, daher wird mit synthetischen Experimenten in Abschnitt 6.4 die Leistung des Verfahrens mit deren einer bekannten Optimal-Lösung verglichen.

Neben der generellen Funktionalität des Verfahrens wird die Einsetzbarkeit in der Industrie betrachtet. In Abschnitt 6.5 wird das Einsatzpotential im Hinblick auf Fachapplikationen beschrieben und konkrete Beispiele betrachtet. Auf Basis der Rahmenbedingungen werden potenziell geeignete Prozesse beschrieben, wie sie in einem Telekommunikationsunternehmen vertreten sind, gleichzeitig werden die Grenzen des Verfahrens aufgezeigt und wo konkret technische Hindernisse liegen.

Zur Anpassung der Leistungsaufnahme von Rechenzentren existieren bereits Verfahren mit ähnlichem Fokus. In Abschnitt 6.6 werden diese Ansätze und das OHMA-Verfahren bezüglich der Leistungssteuerung und Erfüllung der gestellten Anforderungen verglichen.

6.1. Ergebnisse

Kapitel 5 beschreibt die Modellierung des Verfahrens sowie die Implementierung in dem Simulationsframework Uppaal. Die erzielbaren Ergebnisse des Verfahrens sollen nun anhand eines Beispielszenarios mit 40 VMs und 8 PMs in einer Uppaal-Simulation demonstriert werden. Die verwendeten VM-Profile (s. Anhang Tabelle A.1) sind reale Datensätze, die für diese Zwecke freundlicherweise vom technischen Betriebsteam eines Telekommunikationsdienstleisters zur Verfügung gestellt wurden.

6.1.1. Absenken der Leistungsaufnahme

Ausgehend von einer hundertprozentigen Leistungsaufnahme werden jeweils Leistungsaufnahmen entsprechend 90%, 80%, 70% und 60% gefordert. In Abbildung 6.1 sind die Ergebnisse grafisch dargestellt. Neben der erreichten Leistungsaufnahme und der nötigen Zeitdauer zum Erreichen der Allokation ist auch ein Effizienz-Wert dargestellt.

Das Maß der Effizienz eines Servers wurde hier auf den Wert bei maximaler Auslastung auf 100% normiert. Wie in Abschnitt 2 bereits beschrieben, ist die Effizienz eines Server nicht bei jedem Auslastungsniveau gleich hoch. Die Werte zur Effizienz basieren auf den von [Sta15] zur Verfügung gestellten Daten. Zum Vergleich der Szenarien wird die durchschnittliche Effizienz aller betriebenen Server angegeben.

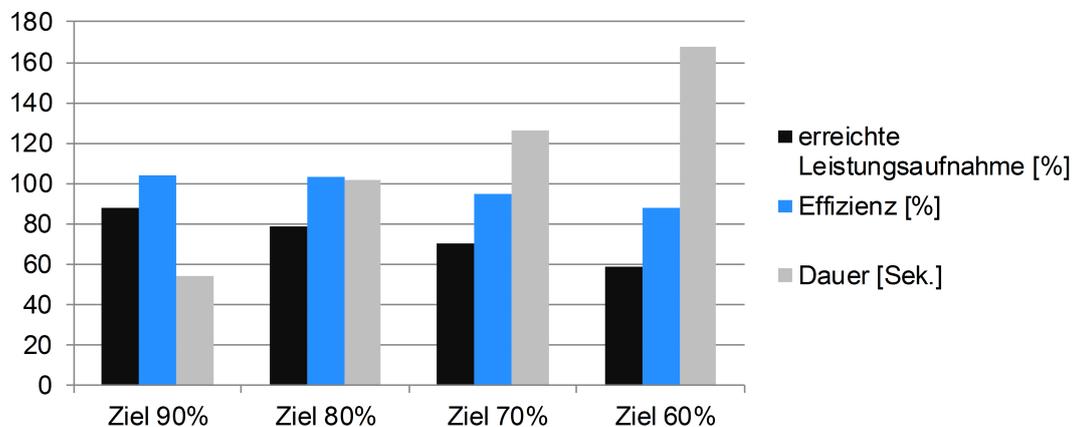


Abbildung 6.1.: Ergebnisse von Simulationen mit unterschiedlichen Zielwert-Vorgaben zum Absenken der Leistungsaufnahme

Die Ergebnisse zeigen, dass die erreichte Leistungsaufnahme in den Szenarien bis auf wenige Watt der Forderung entspricht und die gestellte Aufgabe erfüllt werden konnte. Die Effizienz der Server liegt auf einem hohen Niveau und sinkt auch bei Szenarien mit größeren Leistungsreduzierungen nur leicht ab. Die Migrationsdauer nimmt analog mit dem geforderten Betrag der Leistungsreduzierung zu, da entsprechend mehr VMs zum Erreichen einer höheren Leistungsdifferenz migriert werden müssen. In den Ziel-Szenarien mit 90% und 80% liegt die Effizienz leicht über 100%, dies geht auf den typischen, sigmoidalen Verlauf der Effizienz der Server zurück, wie in Abschnitt 2.6.1 beschrieben.

Besteht die Möglichkeit, nicht genutzte Server abzuschalten, lässt sich die Effizienz weiter erhöhen. Die Abbildung 6.2 zeigt die Ergebnisse der gleichen Simulation mit vollständiger Abschaltung von Servern. Insbesondere in den Szenarien mit einer Absenkung der Leistungsaufnahme auf 70% und 60% zeigt sich der erhöhte Grad an Effizienz. In diesen Szenarien werden mehrere Server durch abgebende Migrationen komplett frei, sodass keine VMs mehr betrieben werden und sie abgeschaltet werden können. Durch die Möglichkeit einer Server-Abschaltung lässt sich die Leistungsaufnahme bis auf 50% reduzieren, da die Hälfte der Server komplett abgeschaltet werden kann. Durch das sukzessive Abschalten der

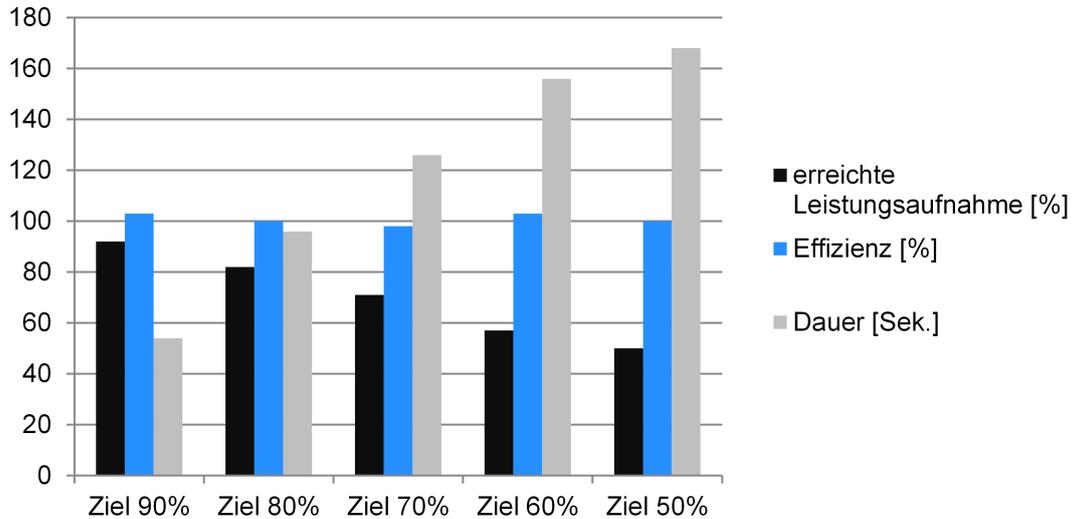


Abbildung 6.2.: Ergebnisse von Simulationen mit unterschiedlichen Zielwert-Vorgaben und Serverabschaltung zum Absenken der Leistungsaufnahme

Server in den jeweiligen Szenarien flacht die durch die Balkengrafik angedeutete Kurve der Zeit-Dauer leicht ab. Der Effekt einer VM-Migration auf die Leistungsaufnahme wird hier durch eine zusätzliche Abschaltung von Servern verstärkt, sodass weniger VM-Migrationen erforderlich werden.

Im Vergleich zu Abbildung 6.1 sind die erhöhten Effizienzwerte erkennbar, bei ansonsten gleichen Ergebnissen bezüglich erreichter Leistungsaufnahme.

Leistungsaufnahme im Verlauf

In Abbildung 6.3 ist der Verlauf der Leistungsaufnahme dargestellt, wie er sich in einem Szenario mit einer Anforderung von 50% Leistungsaufnahme in der Simulation ergeben hat. Es sind die Daten in der Granularität nach jeweils 5 weiteren VM-Migrationen dargestellt.

Es ist deutlich zu erkennen, dass die ersten 5 Migrationen bereits den höchsten Beitrag zum Absenken der Leistungsaufnahme beisteuern. Dies wird dadurch erreicht, dass zunächst die VMs migriert werden, die schnell migrierbar sind und einen hohen Effekt auf die Leistungsaufnahme haben. Bei den darauf folgenden Migrationen stehen nur noch VMs mit geringerem Einfluss zur Verfügung, wie es sich in dem abflachenden Verlauf der Leistungsaufnahme zeigt.

Mit dem beschriebenen Verhalten wird eine Anforderung aus dem Bereich der Regelenergie abgedeckt, wo eine deutlich erkennbare Reaktion in Form einer Leistungsänderung gefordert wird (vgl. Abschnitt 2.8 und Abschnitt 4.6).

6. Potentialanalyse und Evaluation

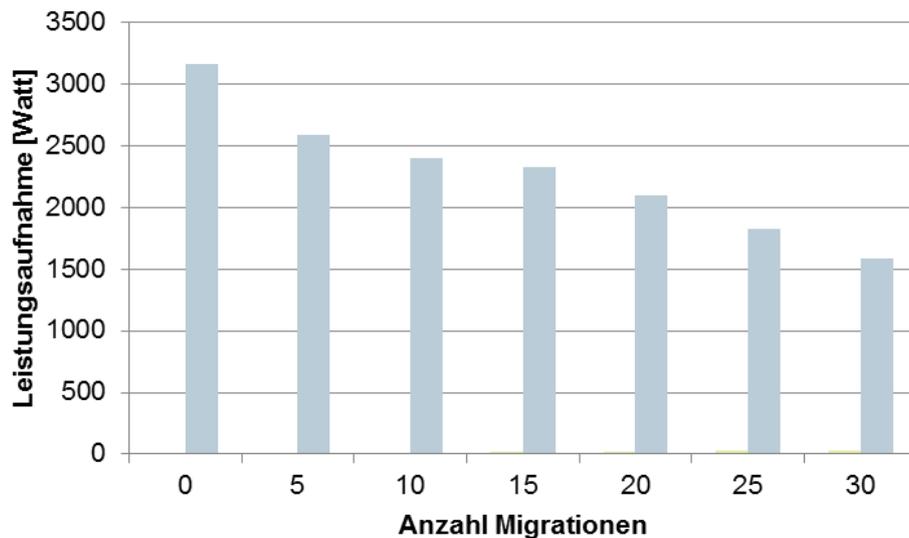


Abbildung 6.3.: Verlauf der Leistungsaufnahme während des Absenkens auf einen Zielwert von 50% der Maximalleistung

6.1.2. Erhöhen der Leistungsaufnahme

Bei einer Erhöhung der Leistungsaufnahme werden dieselben Kriterien bezüglich einer schnellen Migration und hoher Effektivität einer VM-Migration angewandt. Zu dem bekannten Szenario mit 40 VMs und 8PMs wurden ausgehend von einer Anfangskonfiguration mit 50% Leistungsaufnahme Leistungsaufnahmen entsprechend 60%, 70%, 80% und 90% gefordert. Die Ergebnisse sind in Abbildung 6.4 dargestellt. Die erzielbaren Effekte sind vergleichbar mit denen beim Absenken der Leistungsaufnahme.

In diesem Abschnitt wurden diverse Ergebnisse aus verschiedenen Szenarien zum Absenken und Erhöhen der Leistungsaufnahme dargestellt. Die Ergebnisse zeigen, dass die gestellten Anforderungen erfüllt werden konnten. Das OHMA-Verfahren nutzt dabei eine Ressourcen-Überprovisionierung, die mit nur einem Rechenzentrumsstandort realisierbar ist. Damit sind die Forschungsfragen Nr. 1 und 2 beantwortet.

Für eine detaillierte Betrachtung des Rechenaufwands erfolgt im nächsten Abschnitt eine Analyse der Komplexität des Verfahrens.

6.2. Komplexitätsbetrachtung

Wie in Abschnitt 4.4 bereits beschrieben, ist der Aufwand, alle möglichen Allokationen zu betrachten, sehr hoch und insbesondere für Online-Verfahren ungeeignet, die reaktiv agieren müssen. Mit Hilfe der sogenannten O-Notation und der Bestimmung der Komplexität von Algorithmen lässt sich eine relative Aussage über Rechenzeit, den benötigten

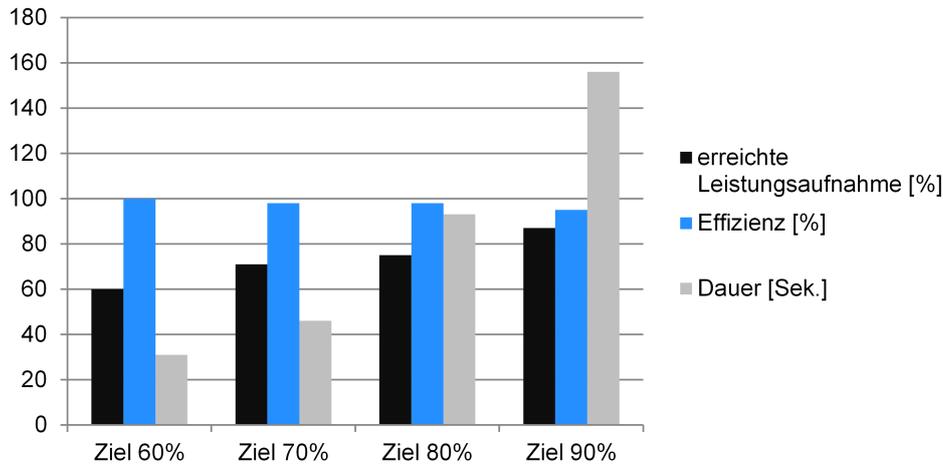


Abbildung 6.4.: Ergebnisse von Simulationen mit unterschiedlichen Zielwert-Vorgaben und Serverabschaltung zur Erhöhung der Leistungsaufnahme

Speicherplatz und die erforderliche Datenübertragung treffen. Es wird ein Vergleich auf einer höheren Ebene möglich, ohne die Betrachtung einer konkreten Implementierung oder Hardwaredetails.

Die Komplexität des absoluten Verfahrens (siehe Abschnitt 4.5.2) bei n VMs und k Servern lässt sich durch

$$\frac{1}{k!} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} j^n \in O(k^n) \quad (6.1)$$

beschreiben, wobei sich eine exponentielle Komplexität zeigt. Bei dem vorgestellten Verfahren wurden diverse Aspekte berücksichtigt, um die Komplexität zu senken und niedrig zu halten. Für die Auswahl einer zu migrierenden VM kommen nur die VMs n' in Betracht, die auf der antagonistischen Server-Gruppe betrieben werden. Aus dieser Menge wird eine Teilmenge von p VMs gebildet, die rein technisch auf der PM betreibbar sind. Im Worst Case werden für die Auswahl der ersten VM alle n' VMs untersucht. In den folgenden Iterationen sind es dann jeweils eine VM weniger, wobei aus diesen p VMs schließlich die VM mit dem größten Effekt zur Zielerreichung gewählt wird. Dieses Prozedere wird maximal mit allen n' VMs vollzogen. Die Komplexität des Verfahrens lässt sich daher mit

$$n' * \left(\sum_{j=1}^p n' - j + p \right) \in O(n) \quad (6.2)$$

beschreiben. Wie erkennbar, ist die Komplexität des in dieser Arbeit entwickelten Verfahrens ($O(n)$) gegenüber der Komplexität einer vollständigen Betrachtung aller Allokationen

6. Potentialanalyse und Evaluation

($O(k^n)$) deutlich reduziert. Das Verfahren kann mit linearer Komplexität beschrieben werden und beantwortet die Forschungsfragen Nr. 3 und 4 bezüglich der Fragestellung nach einem skalierbaren Verfahren mit geringer Komplexität.

6.2.1. Berechnungsdauer

Wie bereits dargestellt, hat das OHMA-Verfahren eine lineare Komplexität. Um nun diese Aussage zu stützen, soll die Berechnungsdauer zur Auswahl einer VM genauer untersucht werden.

In Simulationen von 12 PMs und 60 VMs (verwendete VM-Profile s. Anhang Tabelle A.2) wurden unterschiedliche Leistungsaufnahmen angefordert, die zu einer Streuung von nötigen Migrationen zum Erreichen dieser Leistungsanforderungen geführt haben. Die Messung mit einem Szenario ist hinreichend, da sich mit jeder bereits durchgeführten Migration die Menge potentieller VM-Kandidaten reduziert. Gemessen wurde die Zeitspanne vom Zeitpunkt des Propagierens der Anforderung, bis alle PMs in den Idle-Zustand zurückgekehrt sind. Die modellierten Effekte zur Schutzzeit der VM und die Berücksichtigung von Migrationszeiten (s. Abschnitt 5.4.1) wurden für diese Messungen deaktiviert, um Einflüsse auf die reine Berechnungsdauer so gering wie möglich zu halten. Die Ergebnisse sind in Abbildung 6.5 dargestellt. Werte zu Szenarien von 16 und 17 Migrationen ließen sich im konkreten Beispiel nicht erzeugen, die gewählte Hysterese hat die Migrationsphase entsprechend beendet.

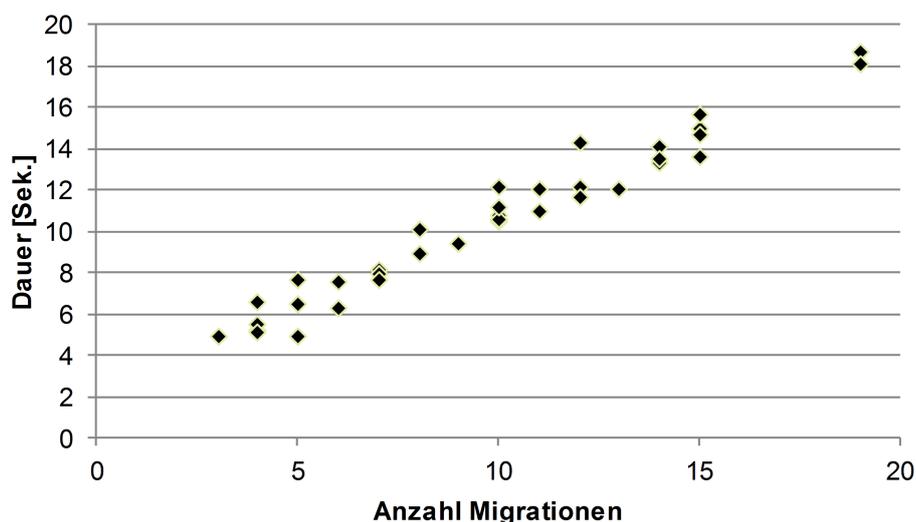


Abbildung 6.5.: Berechnungsdauer des OHMA-Verfahrens zur Auswahl von VMs

Die ermittelten Datenpunkte in der Abbildung 6.5 deuten eine lineare Kurve an, die erwartungskonform mit der beschriebenen Komplexitätsdarstellung ist.

6.3. Bewertungs-Metriken

Um die Leistung des Verfahren bewerten zu können, werden Metriken zur Bewertung der Zielerreichung benötigt. Die Qualität des Ergebnisses kann anhand der Abweichung zwischen der erreichten und der geforderten Leistungsaufnahme erfolgen.

Die Effektivität des Verfahrens lässt sich über den Aufwand bzw. die Anzahl benötigter Migrationen bis zum Erreichen eines stabilen Zielzustands abbilden. Weiterhin ist die Effizienz der durchgeführten Migrationen zu berücksichtigen und die Skalierbarkeit des Verfahrens zu prüfen.

6.3.1. Effizienz und Skalierbarkeit

Zur Betrachtung der Effizienz des Verfahrens ist zunächst eine geeignete Metrik zu wählen, die das Verhältnis von Kosten und Nutzen beschreibt. Die Kosten des Verfahrens werden durch die Anzahl der durchgeführten Migrationen und der Nutzen durch die erreichte Veränderung in der Leistungsaufnahme beschrieben. Die Effizienz wird daher abgebildet durch:

$$\text{Effizienz}_{OHMA} = \frac{\text{erzielte Änderung der Leistungsaufnahme [Watt]}}{\text{durchgeführte Migrationen [Anzahl]}}. \quad (6.3)$$

Die Metrik beschreibt die durchschnittliche Änderung der Leistungsaufnahme in Watt pro Migration.

Zur Analyse der Effizienz und Skalierbarkeit des Verfahrens wurden Simulationen durchgeführt, bei denen ausgehend von 100% Leistungsaufnahme der Server eine Absenkung auf 80 % gefordert wurde. Diese Simulationen wurden in verschiedenen Szenarien (verwendete VM-Profile s. Anhang Tabelle A.3) mit 20 VMs und 4 PMs, 40 VMs und 8 PMs, 60 VMs und 12 PMs, 80 VMs und 16 PMs, 100 VMs und 20 PMs, sowie 160 VMs und 32 PMs wiederholt.

In Abbildung 6.6 sind die Ergebnisse dargestellt, die in jeweils 10 Simulationen je Szenario erzielt wurden. Zu erkennen ist eine breite Streuung der Effizienz bei den Szenarien mit 20 und 40 VMs, die sich mit zunehmender VM-Anzahl reduziert.

In den Szenarien mit kleinerer Dimensionierung werden zur Reduzierung der Leistungsaufnahme nur wenige Migrationen durchgeführt, in den größeren Szenarien sind es entsprechend mehrere Migrationen, so nähern sich die Ergebnisse im großen Szenario einem Durchschnittswert an. Betrachtet man die durchschnittliche Effizienz des Verfahrens in Abbildung 6.7, so ergeben sich je Szenario nur leichte Unterschiede. Im Durchschnitt über alle Szenarien liegt die Effizienz bei etwa 118 Watt pro Migration.

6. Potentialanalyse und Evaluation

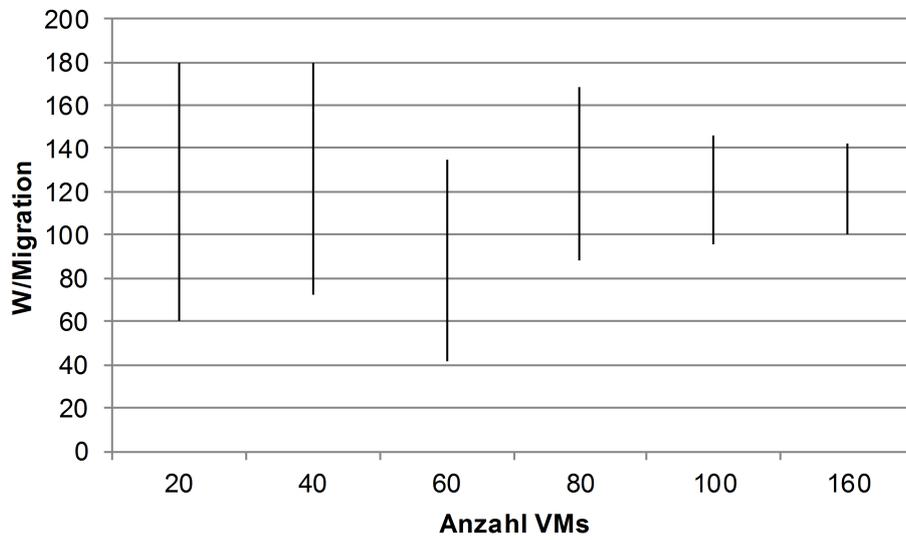


Abbildung 6.6.: Effizienz beim Absenken der Leistungsaufnahme, Ergebnisse aus 10 Simulationen je Szenario

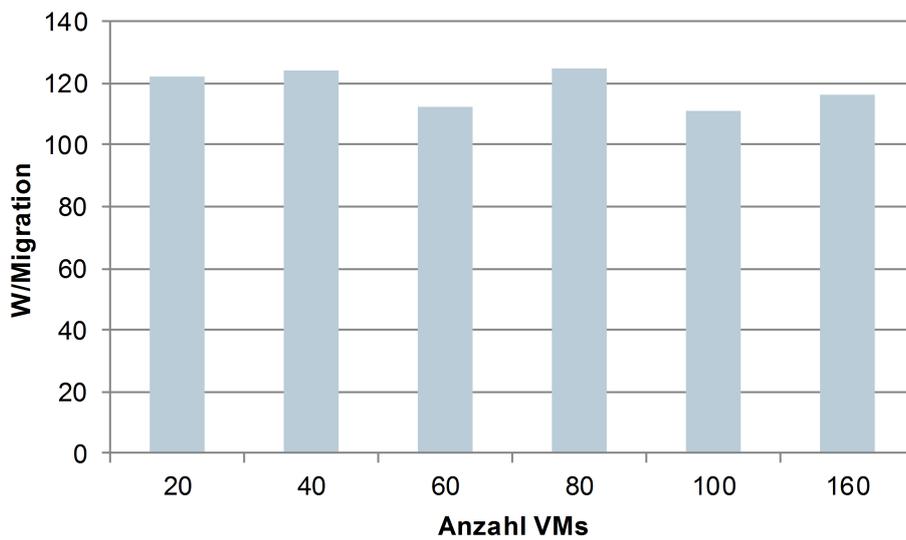


Abbildung 6.7.: Durchschnittliche Effizienz der Migrationen beim Absenken der Leistungsaufnahme

Effizienz beim Erhöhen der Leistungsaufnahme

Analog zum Fall des Absenkens der Leistungsaufnahme wurden für das Erhöhen der Leistungsaufnahme Simulationen der bekannten Szenarien mit 20 VMs und 4 PMs, 40 VMs und 8 PMs, 60 VMs und 12 PMs, 80 VMs und 16 PMs, 100 VMs und 20 PMs, sowie 160 VMs und 32 PMs durchgeführt. Gefordert wurde ebenfalls eine zwanzigprozentige Änderung der Leistungsaufnahme von 50% auf 70%.

In Abbildung 6.8 sind die Ergebnisse aus den Simulationen zur Erhöhung der Leistungsaufnahme dargestellt. Es zeigt sich in allen Szenarien eine Streuung der Effizienz, während das Niveau in den kleineren Szenarien insgesamt höher liegt.

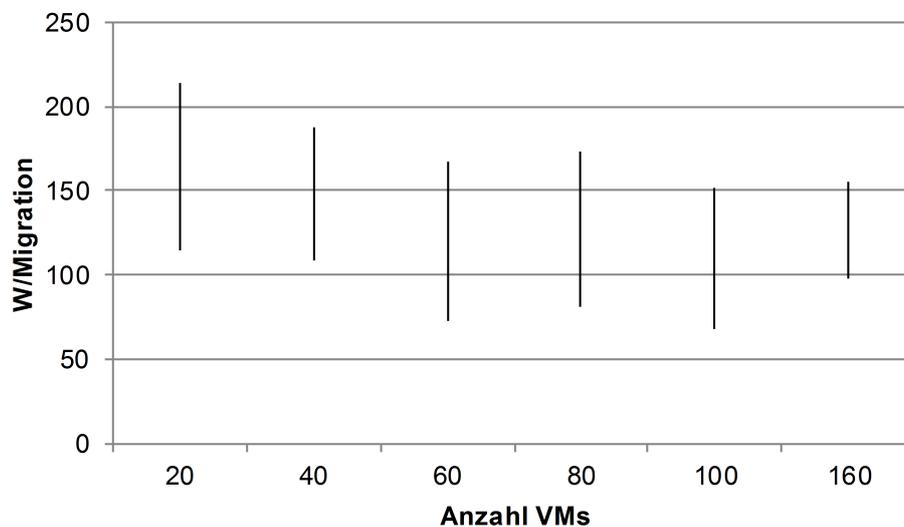


Abbildung 6.8.: Effizienz beim Erhöhen der Leistungsaufnahme

Betrachtet man nun die durchschnittliche Effizienz des Verfahrens in Abbildung 6.9, so zeigt sich in den größeren Szenarien ein Migrationseffekt von etwa 126 Watt pro Migration. In dem kleinen Szenario mit 20 VMs ergibt sich ein höherer Effizienzwert, da hier teilweise nur wenige Migrationen durchgeführt werden müssen und die VMs mit den besten Effekten für eine Migration gewählt werden können. In den größeren Szenarien müssen größere Anzahlen von Migrationen durchgeführt werden. Bei den ersten Migrationsschritten stehen auch hier die VMs mit hohem Potential zur Verfügung, bei Folgemigrationen muss jedoch auch auf Kandidaten mit geringerem Potential zur Änderung der Leistungsaufnahme zurückgegriffen werden. Dies erklärt das Abflachen der Effizienz in den größeren Szenarien.

Mit dem Anstieg der Größe eines Szenarios ändert sich die Effizienz des Verfahrens nicht signifikant. Ab einer Größe von 60 VMs und 12 PMs wird ein annähernd gleiches Effizienzniveau erreicht. Bei größeren Szenarien ist kein Rückgang der Effizienz zu beobachten, was gleichsam zeigt, dass mit zunehmender Größe der Szenarien keine negativen Effekte auf die Effizienz des Verfahren zu erwarten sind.

6. Potentialanalyse und Evaluation

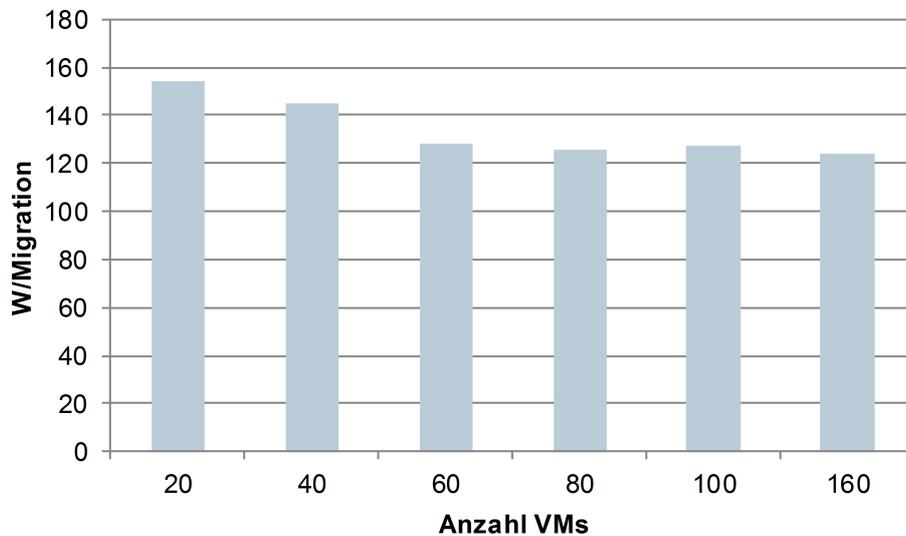


Abbildung 6.9.: Durchschnittliche Effizienz der Migrationen beim Erhöhen der Leistungsaufnahme

6.3.2. Effektivität

Zur Bewertung der Effektivität des Verfahrens soll der Grad der Zielerreichung herangezogen werden. Um Zielwerte und Zielerreichung unterschiedlicher Szenarien miteinander vergleichen zu können, wird ein prozentualer Wert ermittelt, um so die Vergleichbarkeit zu gewährleisten. Die geforderte Änderung der Leistungsaufnahme wird jeweils mit 100% angesetzt und stellt den Referenzwert dar. Die letztlich durch eine neue Allokation der VMs erreichte Änderung der Leistungsaufnahme kann von dem gestellten Zielwert abweichen und reduziert den Grad der Zielerreichung entsprechend. Die Effektivität wird somit definiert durch:

$$\text{Effektivität}_{OHMA} = \frac{100\%}{\text{geforderte Änderung [Watt]}} * \text{erzielte Änderung [Watt]}. \quad (6.4)$$

Für die Bewertung der Effektivität werden die Ergebnisse derselben Simulationen verwendet wie bei der Betrachtung der Effizienz. Wie in Abbildung 6.10 zu erkennen, ist die Effektivität nicht in jedem Szenario gleich hoch. Dies liegt zum einen an der gewählten Hysterese, die eine Terminierung der Migration gestattet, sobald die erreichte Leistungsaufnahme im Korridor der Hysterese liegt. Die Notwendigkeit der Implementierung eines Hysteresewertes ist bereits in Abschnitt 5.9.3 dargelegt worden.

Zum anderen ist die erreichbare Effektivität von den VMs abhängig, die für eine Migration zur Verfügung stehen. Durch die randomisierte Auswahl einer Teilmenge von VMs (vgl. Abschnitt 5.7) sind jeweils pro Migrationslauf potentiell andere VMs verfügbar. Generell

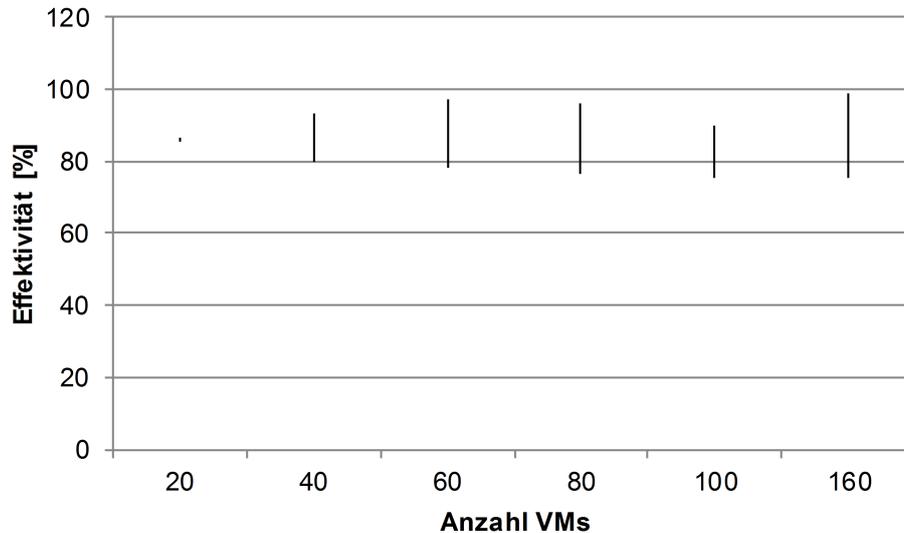


Abbildung 6.10.: Effektivität beim Absenken der Leistungsaufnahme

hat die Ausprägung der VMs Einfluss auf die Effektivität, je nach Ressourcenprofil sind VMs besser oder schlechter geeignet. VMs mit kleiner CPU-Auslastung sind für eine feingranulare Annäherung der Zielvorgaben ideal, so lässt sich ein hoher Effektivitäts-Wert erreichen. Es sind allerdings mehr Migrationen erforderlich als bei VMs mit höherer CPU-Auslastung.

In dem Szenario mit 20 VMs ist kaum eine Streuung der Ergebnisse erkennbar und die Effektivität liegt bei 86%. In diesem kleinen Szenario sind nur wenige Migrationen nötig, um das Leistungsziel zu erreichen. Hier wurde jeweils auf dieselben VMs mit hoher CPU-Auslastung für die Migrationen zurückgegriffen.

Wie auch in den vorangestellten Untersuchungen ist erkennbar, dass die Effektivität in den Szenarien mit mehr als 40 VMs ein höheres Maß erreichen kann. Allerdings sind die Ergebnisse aufgrund des Hysterese-Wertes ebenfalls breiter gestreut und es wird der Ziel-Korridor der Hysterese ausgenutzt.

Das implementierte OHMA-Verfahren kann in Abhängigkeit von den betriebenen VMs und einer passend gewählten Hysterese in Szenarien ab 40 VMs und 8 PMs Effektivitäts-Werte von 93% und mehr erreichen, wie die Ergebnisse in Abbildung 6.10 zeigen.

Effektivität beim Erhöhen der Leistungsaufnahme

Die Abbildung 6.11 zeigt die Effektivität des Verfahrens beim Erhöhen der Leistungsaufnahme in den bekannten Szenarien. Auch hier ist eine Streuung der Effektivitätswerte erkennbar, die mit zunehmender Größe des Szenarios abnimmt. Insbesondere in dem Szenario mit 20 VMs werden nur wenige Migrationen durchgeführt und die Auswahl der VMs

6. Potentialanalyse und Evaluation

ist beim Erhöhen der Leistungsaufnahme weniger restriktiv, wie in Abschnitt 5.7 beschrieben. Es kommt folglich für die ersten Migrationen eine Auswahl an VMs mit großer Diversität zusammen. Die Migration dieser VMs mit vornehmlich hoher CPU-Auslastung führt zu einer groben Zielerreichung mit entsprechender Nutzung des Hysteresebereichs. In den größeren Szenarien stehen nach mehreren bereits erfolgten Migrationen lediglich VMs mit kleinerer CPU-Auslastung zur Verfügung. So erklärt sich eine feingranulare Annäherung an die Zielvorgabe und dadurch eine geringere Streuung der Effektivitätswerte.

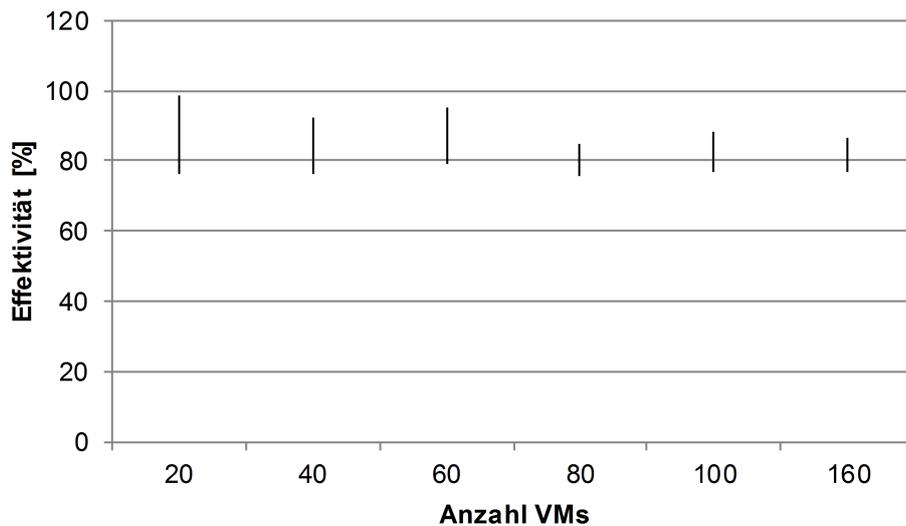


Abbildung 6.11.: Effektivität beim Erhöhen der Leistungsaufnahme

6.4. Synthetische Experimente

Das in dieser Arbeit vorgestellte Verfahren reduziert die Komplexität der ursprünglichen Aufgabenstellung erheblich und basiert auf einer Heuristik. Wie bei den meisten heuristischen Verfahren entsprechen auch hier die erzeugten Lösungen in ihrer Qualität nicht den Lösungen, die mit dem absoluten Verfahren zu finden sind, und weichen vom exakten Optimum ab. In diesem Abschnitt wird nun die Qualität der mit Hilfe des Verfahrens erreichten Lösung mit dem möglichen Optimum verglichen.

Zur Untersuchung des Sachverhalts wird eine Ausgangsallokation hergestellt und im Anschluss eine definierte Leistungsaufnahme gefordert, zu der die optimale Lösung bekannt ist. Die durch das Verfahren gelieferte Allokation und damit erreichte Leistungsaufnahme wird mit der optimalen Allokation für diese Anforderung verglichen.

In dem bekannten Szenario, bestehend aus 60 VMs und 12 PMs, werden die VMs in einem ersten Schritt auf die PMs verteilt, um die Ausgangsallokation herzustellen. Die Ausgangsallokation erzeugt auf den PMs eine Gesamtleistungsaufnahme von 3168 Watt. Gefordert

	Start [W]	Ziel [W]	erreicht [W]	Migrationen[#]
Experiment 1	3168	2549	2611	7
Opt. Lösung	3168	2549	2549	5

Tabelle 6.1.: Ergebnisse eines synthetischen Szenarios und bekannter Optimal-Lösung

wird eine Leistungsaufnahme von 2549 Watt, zu der die optimale Lösung bekannt ist. Die Ergebnisse des Test-Szenarios und der Optimal-Lösung sind in Tabelle 6.1 dargestellt.

Das Verfahren hat in dem konkreten Fall zu der gestellten Anforderung eine Leistungsaufnahme von 2611 Watt erreichen können. Zur Herstellung dieser Allokation waren 7 Migrationen notwendig. Die Leistungsaufnahme weicht jedoch um 62 Watt von der gestellten Anforderung ab. Dies liegt im Verfahren selbst begründet und wurde bereits in Kapitel 5 dargestellt und steht im Zusammenhang mit der implementierten Hysterese. Die Herstellung der optimalen Allokation benötigt nur 5 Migrationen und liefert exakt die geforderte Leistungsaufnahme.

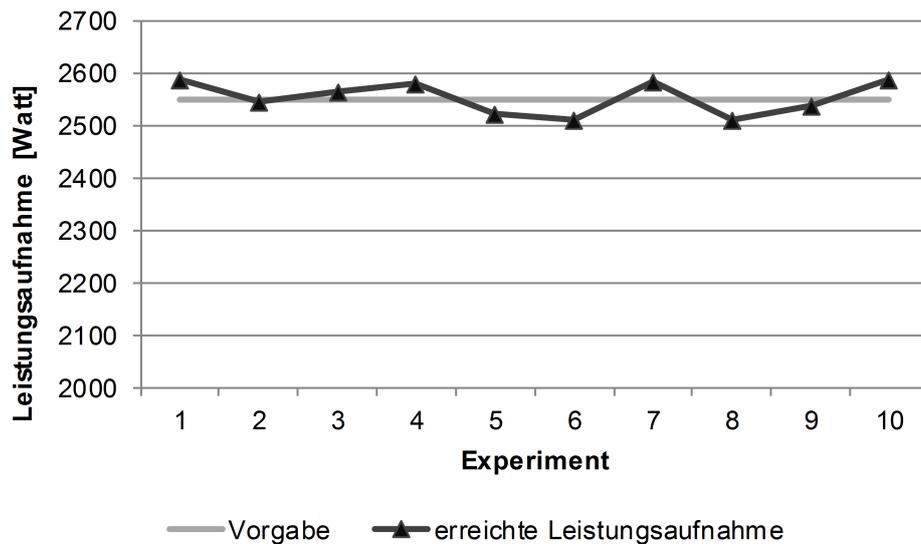


Abbildung 6.12.: Leistungsaufnahme mit Hysterese von 1,25%

In Abbildung 6.12 sind die gelieferten Ergebnisse von 10 Experimenten und die Zielvorgabe (grauer Balken), die mit der optimalen Lösung erreicht werden kann, dargestellt. Bei der gewählten Hysterese von 1,25% kam es nur zu geringen Abweichungen von der Zielvorgabe.

In Abbildung 6.13 sind die benötigten Migrationen zum Erreichen der jeweiligen Allokationen dargestellt. Es ist erkennbar, dass die enge Zielvorgabe teilweise bis zu 16 Migrationen erfordert. Im Durchschnitt liefert das Verfahren eine Leistungsaufnahme von 2553,3 Watt und benötigt 7,1 Migrationen.

Zum weiteren Vergleich wurden die Experimente mit einer Hysterese von 3% wiederholt. Die Ergebnisse sind in Abbildung 6.14 dargestellt und zeigen leicht erhöhte Abweichungen,

6. Potentialanalyse und Evaluation

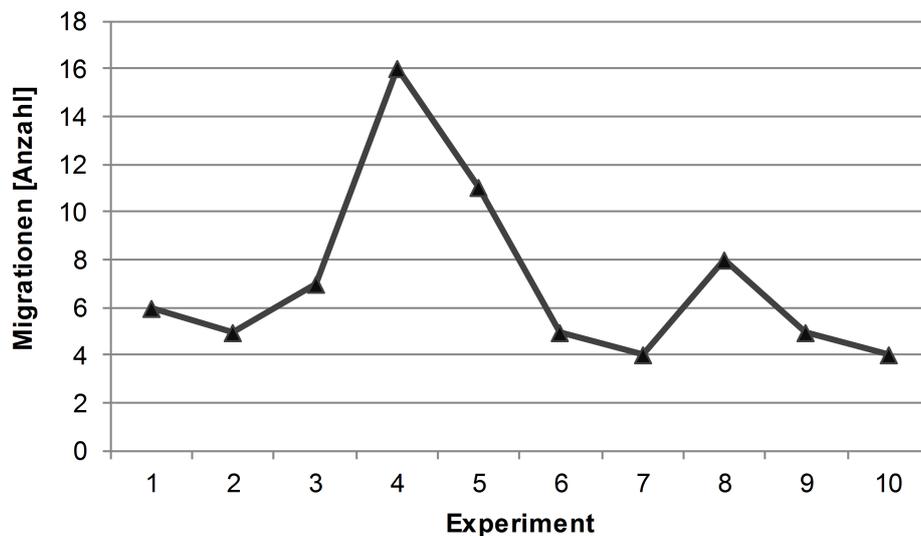


Abbildung 6.13.: Anzahl benötigter Migrationen mit Hysterese von 1,25%

korrespondierend mit der Hysterese von 3%. Betrachtet man nun hierzu die erforderlichen Migrationen in Abbildung 6.15, so zeigt sich eine deutlich verringerte Anzahl erforderlicher Migrationen. Durchschnittlich ergibt sich eine Leistungsaufnahme von 2583 Watt, wobei 4,5 Migrationen erforderlich sind.

	Start [W]	Ziel [W]	erreicht [W]	Migrationen[#]
ØHysterese 1,25%	3168	2549	2553,3	7,1
ØHysterese 3%	3168	2549	2583	4,5
Opt. Lösung	3168	2549	2549	5

Tabelle 6.2.: Zusammenfassung der Ergebnisse aus synthetischen Experimenten

In Tabelle 6.2 sind die Ergebnisse noch einmal zusammengefasst. Im direkten Vergleich der Lösungen ist nur ein marginaler Unterschied erkennbar. Die mit dem vorgestellten Verfahren und einer Hysterese von 3% erreichte Lösung zeigt im Verhältnis zu den benötigten Rechenoperationen ein akzeptables Ergebnis. Die durchschnittliche Anzahl der Migrationen lag noch unter der Anzahl der optimalen Lösung.

6.5. Einsetzbarkeit des Verfahren im Kontext von Fachapplikationen

Applikationen haben bezüglich ihres Ressourcenbedarfs ganz unterschiedliche Anforderungen, wie Boroff et. al in [BKB07] beschrieben haben. Je nach Applikationstyp und

6.5. Einsetzbarkeit des Verfahren im Kontext von Fachapplikationen

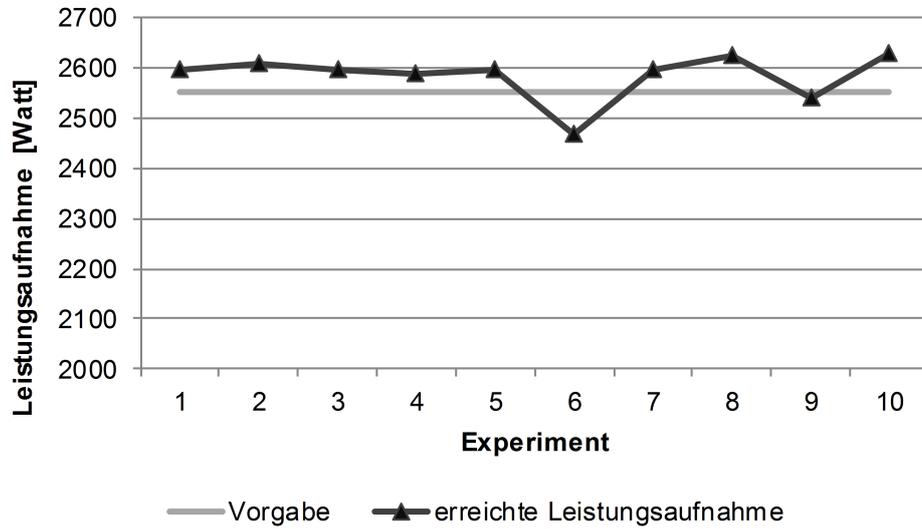


Abbildung 6.14.: Leistungsaufnahme mit Hysterese von 3%

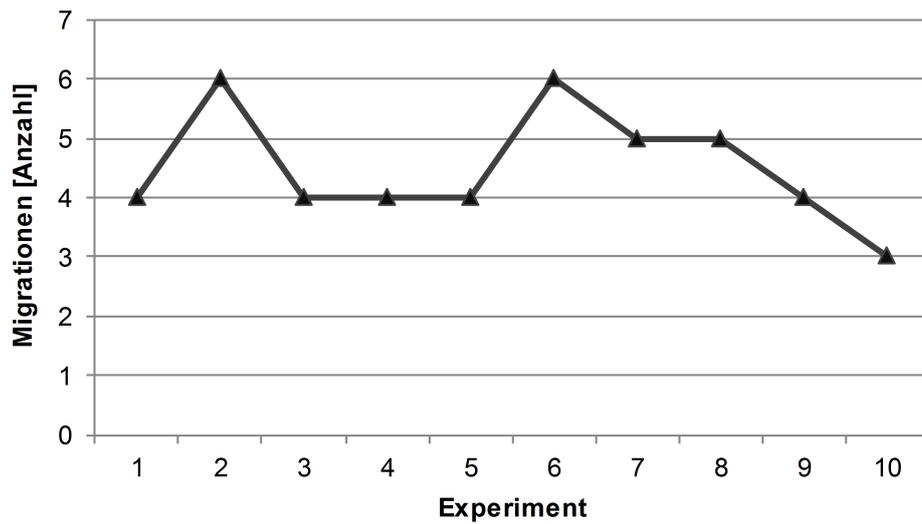


Abbildung 6.15.: Migrationen mit Hysterese von 3%

6. Potentialanalyse und Evaluation

abgebildeter, fachlicher Prozesse kann eine Eignung für das Verfahren vorliegen. Das in dieser Arbeit erstellte Verfahren sieht bewusst eine Veränderung der Performance von Applikationen vor und schließt damit bereits Applikationen mit Realzeitanforderungen aus. Welches Einsatzpotential für das Verfahren abseits dieser Restriktionen in der Industrie besteht, wird anhand der IT-Systeme und Prozesse eines Telekommunikationsunternehmens genauer betrachtet.

6.5.1. Billing-Prozess eines Telekommunikationsunternehmens

Ein fachlicher Prozess, der sehr hohe Last auf den Serversystemen hervorruft und damit ein idealer Kandidat für das vorgestellte Verfahren ist, ist der Abrechnungs- oder Billing-Prozess in Telekommunikationsunternehmen.

Zur Erstellung der Kundenrechnung muss zunächst ermittelt werden, welche Leistungen der Kunde tatsächlich in Anspruch genommen hat. Auf den Wählsystemen, die die Kommunikationsverbindungen zwischen den Teilnehmern herstellen, werden die Verbindungsdaten in sogenannten Call Data Records (CDRs) gespeichert. Die CDRs enthalten Details zu der aufgebauten Verbindung und werden aktiv aus den Systemen ausgelesen und in einem Rohdatenpool gesammelt. Hier laufen ebenso CDRs aus dem Mobilfunkbereich als auch Abrechnungsdaten der Online-Videothek und weitere verbrauchsabhängige Daten wie verbrauchtes Internetvolumen usw. ein. Die CDRs aus den verschiedenen Bereichen müssen für eine weitere Bearbeitung in ein einheitliches Format gebracht werden, dieser Vorgang wird Mediation genannt und stellt die erste Stufe in der Verarbeitung der Daten dar.

Im Anschluss erfolgt das Rating der CDRs. Hier werden den einzelnen CDRs mit Kundendaten angereichert und ein Preis zugeordnet. Im letzten Schritt erfolgt das Billing, die Erstellung der Kundenrechnung. Die für einen Kunden erbrachten Leistungen werden anhand der Vertragsstruktur zusammengefasst und für ein Kundenkonto berechnet. Ein Kundenkonto kann mehrere Verträge enthalten, Kostenpositionen werden dementsprechend getrennt. Beispielsweise sind Mobilfunkverbindungen dem Mobilfunkvertrag und Festnetzverbindungen dem Festnetzvertrag zugeordnet. Dabei müssen weitere Vertragsbestandteile wie Flatrates oder Minutenpakete berücksichtigt werden.

Der Abrechnungszeitraum beträgt in dem betrachteten Fall jeweils genau einen Kalendermonat. Ziel ist es, bis zum siebten Tag des folgenden Monats die Rechnungen an den Kunden versandt zu haben. Mit diesen Vorgaben lassen sich Zeitfenster definieren, in deren Rahmen die rechenintensiven Prozesse verschoben werden können. Insbesondere der Mediation-Prozess und das Rating können bereits tagesaktuell vorgenommen werden und bieten so Potential für eine bedarfsgerechte Adaption der Server-Leistungsaufnahme.

Abrechnungsprozesse und Rechnungsläufe von ähnlicher Struktur gibt es noch diverse weitere in dem betrachteten Unternehmen. Die Energiesparte erzeugt ebenfalls Kundenrechnungen, hinzu kommen Provisionsabrechnungen für Vertrieb und Marktpartner sowie Gehaltsabrechnungen.

6.5. Einsetzbarkeit des Verfahren im Kontext von Fachapplikationen

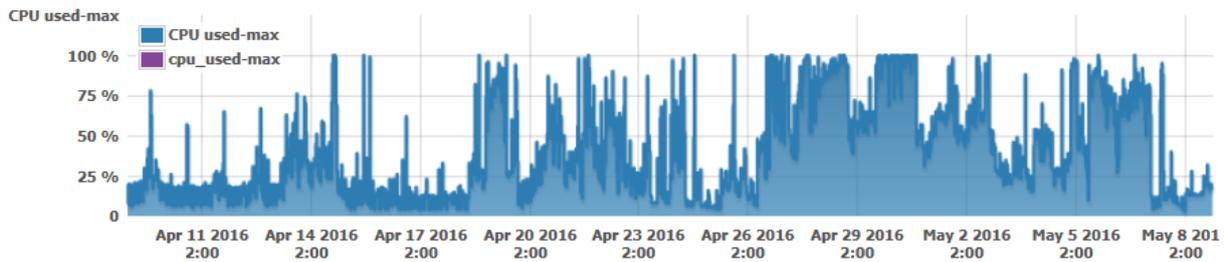


Abbildung 6.16.: Lastprofil eines Billing-Laufs

Die Abbildung 6.16 zeigt exemplarisch das Lastverhalten des Servers, auf dem der beschriebene Billing-Lauf stattfindet. Es ist deutlich das Ende des Abrechnungszeitraums mit ansteigender Server-Auslastung zu sehen, genauso wie auch das Auslastungsniveau am 8. Mai wieder abfällt, nachdem die Rechnungen erstellt worden sind.

6.5.2. Provisionierungsprozesse

Eine weitere Prozessart, die in Telekommunikationsunternehmen verbreitet vertreten ist und durch IT-Systeme unterstützt wird, ist die der Provisionierungsprozesse. Unter dem Provisionierungsprozess wird bei TK-Unternehmen der Prozess verstanden, der die Einrichtung von bestellten Leistungsmerkmalen auf den jeweiligen Systemen vornimmt, die diese Leistungen letztlich dem Kunden zur Verfügung stellen. Dazu zählen die Einrichtung eines Benutzerkontos auf einem Radius-Server, die Freischaltung für die Online-Videothek und die Konfiguration der Rufnummern auf den Sprachdienst-Systemen.

Der Provisionierungsprozess wird bei dem betrachteten TK-Unternehmen direkt mit dem Auftragsprozess angestoßen. Der Auftrag durchläuft alle notwendigen Maßnahmen sofort. Die Einrichtung der Leistungsbestandteile erfolgt in der Regel ebenfalls sofort, versehen mit einem entsprechenden Aktivdatum zum Vertragsstart. Die jeweiligen Leistungsmerkmale sind also bereits kurz nach Auftragseingang eingerichtet, können aber erst ab dem Aktivdatum vom Kunden genutzt werden.

Im Bereich der Provisionierungsprozesse besteht somit ein hohes Potential für die Nutzung des OHMA-Verfahrens. Zwischen der Beauftragung von Dienstmerkmalen durch den Kunden und des Aktivdatums liegen oftmals mehrere Wochen. Dies ist häufig auf bestehende Vertrags-Restlaufzeiten vorheriger Anbieter zurückzuführen. Oftmals müssen auch konkrete Vordienstleistungen bereitgestellt werden, wie Tiefbaumaßnahmen zur Verlegung von Glasfaserkabeln oder Ähnliches.

Durch diese Gegebenheiten können die Provisionierungsprozesse in ihrer Bearbeitung in einem großen Zeitfenster verschoben werden und bieten ebenfalls hohes Potential für eine Leistungsanpassung der unterstützenden Server-Systeme. Neben den hier ausführlicher beschriebenen Prozessen zeigten sich diverse weitere Prozesse aus dem Bereich des Big-Data und Business Intelligence, die sich ebenfalls für eine zeitlich variable Bearbeitung eignen.

6.5.3. Grenzen des Verfahrens

Bei der Analyse von Fachapplikationen und den darin abgebildeten Prozessen hat sich ein sehr breites Potential für die Anwendung des vorgestellten Verfahrens gezeigt. In nahezu allen betrachteten Applikationen werden asynchron laufende Prozesse bearbeitet und sind damit für eine variable Verschiebung in der Ausführung geeignet. Insbesondere größere Applikationen bestehen aus Verbunden mehrerer Kleinstapplikationen, die einzelne Teilaufgaben abbilden. In so einem Konstrukt lässt sich die Trennung von kritischen und flexiblen Bereichen leicht vollziehen und das OHMA-Verfahren zur Anwendung bringen.

Kleinere Fachapplikationen sind tendenziell allerdings eher monolithisch aufgebaut. Oftmals wird der UI-Teil und weitere Applikationslogik in derselben Java Virtual Machine betrieben. Eine Trennung des UI-Anteils, der keine Performance-Einbußen durch eine Leistungsanpassung erfahren darf, ist somit nicht möglich. Derartige Applikationskonstrukte sind nicht für den Einsatz des OHMA-Verfahrens geeignet.

6.6. Vergleich mit anderen Verfahren

In Tabelle 6.3 sind die jeweiligen Technologie-Ansätze aus Abschnitt 3.9 noch einmal dargestellt, hier wurde das OHMA-Verfahren ergänzt. Nur das im Rahmen dieser Arbeit erstellte OHMA-Verfahren kann alle beschriebenen Anforderungen (vgl. Abschnitt 4.6) abdecken.

In Abschnitt 3 wurden mehrere Ansätze zur Manipulation der Leistungsaufnahme und Steigerung der Effizienz von Rechenzentren vorgestellt. Verfahren, die eine umfassende Beeinflussung der Leistungsaufnahme realisieren, basieren hauptsächlich auf Scheduling-Verfahren, bei denen zu verarbeitende Tasks in dem gewünschten Umfang, passend zur Ziel-Leistungsaufnahme, scheduler-gesteuert zur Ausführung gebracht werden. Wird eine akute Reduzierung der Leistungsaufnahme erforderlich, werden diese Tasks abgebrochen. Bei einer derartigen Vorgehensweise sind zweifelsohne jederzeit jegliche Leistungsaufnahme-Ziele erreichbar. Bei diesen Ansätzen bleiben jedoch Anforderungen bezüglich der Service-Qualität unberücksichtigt oder werden bewusst ausgespart.

Der Ansatz, die Leistungsaufnahme eines Rechenzentrums an die Energieverfügbarkeit anzupassen, wurde auch in [CCC14] verfolgt. Hier werden allerdings QoS-Anforderungen bzw. SLAs berücksichtigt. Ebenfalls auf der Basis eines Scheduling-Verfahrens wird eine Leistungsreduzierung von bis zu 50% erreicht. Auch das hier vorgestellte Verfahren erreicht durch den technischen Aufbau eine Leistungsreduzierung von bis zu 50%. Weitere Verfahren mit dem Ziel einer Leistungssteuerung auf Basis von VM-Migrationen sind zum Zeitpunkt der Erstellung dieser Arbeit nicht bekannt.

Die Leistungsaufnahme von Rechenzentren in Deutschland liegt bei etwa 10 TWh pro Jahr, mit steigender Tendenz [Wis19]. Hieraus ergibt sich durchschnittlich eine permanente Leistungsaufnahme von 1141,5 MW. Würde nun das OHMA-Verfahren bei der Hälfte der Server Anwendung finden, ergäbe sich eine steuerbare Leistungsaufnahme von

6.6. Vergleich mit anderen Verfahren

Kurz-Beschreibung des Ansatzes	Effizienz im Fokus	lokal einsetzbar	Leistungsaufnahme beeinflussbar	agnostisch	Quellen
EDF-Schedule, Tasks werden entsprechend der Energieverfügbarkeit gestartet oder beendet	✗	✓	✓	✗	[KGA ⁺ 11], [CCC14]
Effizienzsteigerung der Klimatisierung. Taskverteilung anhand Servertemperatur	✓	✓	✗	✗	[TGV08], [BF07], [MCRS05], [TGSC06], [TGV07], [XFL13], [LCB ⁺ 12]
Effizienzsteigerung durch optimierte Ressourcennutzung, Migration von VMs	✓	✓	✗	✓	[DGC13], [PY10], [WTLS ⁺ 09], [CFF12], [MGD ⁺ 12], [PLM ⁺ 13]
Geografische Verschiebung. Rechenlast wird an Standorte mit höherer Energieverfügbarkeit verlagert, Migration von VMs	✗	✗	nur lokaler Effekt (✓)	✓	[LLW ⁺ 11], [ZWW11], [MDKS12]
<u>OHMA-Verfahren:</u> Ressourcen-Überprovisionierung, Migration von VMs	✓	✓	✓	✓	

Tabelle 6.3.: Aktuell verfügbare Verfahren im Vergleich mit dem OHMA-Verfahren

6. Potentialanalyse und Evaluation

228,3 MW, die in einer Zeitspanne von wenigen Minuten erbracht werden könnte. Zum Vergleich: Die negative Sekundärreserve (-SRL) der TSO EnBW Transportnetze AG liegt bei 311 MW [Con10].

6.7. Zusammenfassung

In diesem Abschnitt wurden erzielbare Ergebnisse des Verfahrens am Beispiel von realen Datensätzen aus der Industrie gezeigt. Die Effizienz der Serversysteme kann über die diversen Leistungsstufen auf einem hohen Niveau gehalten werden und die geforderten Leistungsaufnahmen sind innerhalb weniger Sekunden erreichbar.

Aufgrund des Designs und der gewählten Konfiguration sind Leistungsreduzierungen, ausgehend von der maximalen Leistungsaufnahme der Server, von bis zu 50% mit dem vorgestellten Verfahren zu erzielen. Eine vergleichbare Arbeit von Chen et. al [CCC14], die ebenfalls das Ziel einer Leistungsanpassung der Server zum Ziel hat, erreicht ebenfalls Reduzierungen in diesem Umfang, basiert allerdings auf Scheduling-Verfahren.

Zur Bewertung des Verfahrens wurden synthetische Experimente durchgeführt und die Ergebnisse des Verfahrens mit einer bekannten Optimal-Lösung verglichen. Die mit dem Verfahren gefundenen Lösungen erzeugen nur leichte Abweichungen in der Leistungsaufnahme und die Anzahl benötigter Migrationen ist vergleichbar.

Es wurde eine Komplexitätsbetrachtung durchgeführt, die einen deutlich reduzierten Berechnungsaufwand bei dem vorgestellten Verfahren mit linearer Komplexität ausweist. Bei der Erstellung einer Lösung mit dem absoluten Verfahren ergibt sich eine exponentielle Komplexität. Das OHMA-Verfahren zeigt dagegen eine lineare Komplexität, die mit weiteren Experimenten zur Berechnungsdauer belegt wurde.

Eine Betrachtung bezüglich Effizienz, Effektivität und Skalierbarkeit hat ergeben, dass die Effizienz maßgeblich von der Diversität der betriebenen VMs abhängig ist. Die Effektivität wird wiederum von Hysterese beeinflusst, die ebenfalls auf Basis der betriebenen VMs zu bestimmen ist. Eine Skalierung des Verfahrens wurde mit Ergebnissen aus Szenarien unterschiedlicher Größe gezeigt.

Das Einsatzpotential des Verfahrens in der Wirtschaft wurde in Abschnitt 6.5 an konkreten Beispielen beschrieben. Ein Großteil der Applikationen des betrachteten Telekommunikationsunternehmens verarbeitet asynchrone Prozesse, die für eine Integration in das OHMA-Verfahren geeignet sind. Insbesondere größere Applikationen sind hinreichend modularisiert, um asynchron-arbeitende Komponenten getrennt von kritischen betreiben zu können. Die Grenzen sind bei kleineren Applikationen erreicht, bei denen beispielsweise die Anwendungslogik und der UI-Teil in derselben Java Virtual Machine betrieben werden. Ebenso sind generell monolithisch aufgebaute Applikationen ungeeignet.

Im direkten Vergleich mit bereits verfügbaren Ansätzen schließt das vorgestellte OHMA-Verfahren die Lücke im Bereich der ad-hoc Reaktion, regionaler Anwendbarkeit und Steuerung der Leistungsaufnahme von Servern in Rechenzentren ohne fachliche Eingriffe in die

6.7. Zusammenfassung

betriebenen Applikationen. Die Ergebnisse zeigen ein Potential für eine Leistungsreduzierung von 50%, welches innerhalb von wenigen Minuten abgerufen werden kann.

7. Zusammenfassung und Ausblick

In dieser Arbeit wurde die Fragestellung bezüglich eines nicht-invasiven Verfahrens zur Steuerung der Leistungsaufnahme von Servern in Rechenzentren bearbeitet, um deren energetische Freiheitsgrade zu erweitern. In diesem Abschnitt werden zunächst die Aspekte der Forschungsfragen aus Abschnitt 1.3 aufgegriffen und die erlangten Erkenntnisse zusammengefasst. Im Anschluss erfolgt ein kurzer Ausblick auf weiterführende Themen.

7.1. Erkenntnisse aus der Bearbeitung der Forschungsfragen

Zur Beeinflussung der Leistungsaufnahme von Servern im Betriebskontext von IaaS (und SaaS) zeigten sich die erforderlichen Betriebsmittel und eine Steuerung ihrer Auslastung als dienlich. Es wurde in Experimenten (s. Abschnitt 6.1) gezeigt, wie über konkurrierende Ressourcen-Allokationen die Leistungsaufnahme der Server beeinflusst werden kann. Mit der Technik einer Ressourcen-Überprovisionierung ist hier eine passende Technologie für diesen Einsatzzweck gefunden und beantwortet Forschungsfrage Nr. 1. Die Effekte können mit nur einem Rechenzentrumsstandort realisiert werden.

Weiterhin wurden die Betriebsmittel von Servern im Hinblick auf ihr Verhalten bei unterschiedlichen Auslastungsniveaus und ihr Anteil an der Gesamtleistungsaufnahme beurteilt. Es zeigte sich, dass im Kontext von Rechenzentren eine Fokussierung auf die CPU-Ressource eines Servers sinnvoll erscheint (s. Abschnitt 4.1.1). Auf Basis der gewonnenen Erkenntnisse wurde ein Verfahren implementiert, welches mit der Technik einer Ressourcen-Überprovisionierung die Applikationsperformance verringern und wieder erhöhen kann. Server, die mit DVFS ausgestattet sind, adaptieren ihre Leistungsaufnahme entsprechend, sodass eine gewünschte Ziel-Leistungsaufnahme des Rechenzentrums erreicht werden kann. Somit ist Forschungsfrage Nr. 2 beantwortet.

Die Aufgabe, eine passende Allokation zu einer gewünschten Leistungsaufnahme zu finden, wurde in Abschnitt 4.4 hinsichtlich der Komplexität betrachtet und mit exponentiellem Aufwand bewertet. Eine vollständige Analyse aller Allokationen zum Erreichen einer Leistungsaufnahme ist daher nicht zielführend und mit den gestellten Anforderungen nicht vereinbar. Zur Herstellung der benötigten Allokationen wurde ein heuristisches Vorgehen (s. Abschnitt 5.1) gefunden, das mit linearer Komplexität und in angemessener Zeit zu einer geforderten Leistungsaufnahme die entsprechende VM-Allokation in einem Rechenzentrum herstellen kann. Es wird eine lokale Steuerung der Leistungsaufnahme realisiert, ein zweiter Rechenzentrumsstandort ist nicht erforderlich. Im Vergleich zum

7. Zusammenfassung und Ausblick

absoluten Verfahren zeigte sich eine signifikante Reduktion der Komplexität, die eine Bedienung von ad hoc gestellten Anforderungen für ein reaktives Verhalten erst ermöglicht. Zur Modellierung des Verfahrens wurden Automatenmodelle in dem Framework Uppaal erstellt (s. Abschnitt 5). In der Simulation ist es möglich, eine genaue Betrachtung der Systemzustände durchzuführen und eine Transparenz über die einzelnen Prozessschritte zu erzeugen, wie in Forschungsfrage Nr. 4 gefordert.

Bei der Modellierung des Verfahrens wurde das Thema Effizienz in mehreren Ebenen berücksichtigt, wie in Forschungsfrage Nr. 5 thematisiert wird. Scheduling-Verfahren, bei denen Rechenoperationen abgebrochen werden, wurden als Verfahrenstechnologie ausgeschlossen, denn das Abbrechen von Tasks stellt einen direkten Effizienzverlust dar. Stattdessen wird eine gezielte Provisionierung von Server-Ressourcen eingesetzt. Basierend auf einer Ressourcen-Überprovisionierung wurden die einzelnen Serverressourcen bezüglich ihrer Eignung hierfür untersucht. Bei der in dem vorgestellten Verfahren angewandten CPU-Überprovisionierung konnten keine Effekte gefunden werden, die auf einen Effizienzverlust hindeuten (vgl. Abschnitt 4.1.1).

Zu Effizienzverlusten kommt es in Rechenzentren auch durch die Entstehung von Hotspots. Das implementierte Verfahren wirkt dieser Entwicklung durch eine Gleichverteilung entgegen und unterstützt die gezielte Klimatisierung der Server durch die Aufteilung in zwei Gruppen (s. Abschnitt 5.9.2). Weiterhin wird die optionale Abschaltung nicht-genutzter Server ermöglicht, was sich ebenfalls positiv auf die Effizienz des Rechenzentrums auswirken kann (s. Abschnitt 6.1.1). Neben der Fokussierung auf eine Steuerung der Leistungsaufnahme von Servern konnten die genannten Aspekte in die Modellierung des Verfahrens einfließen und Effizienz-Gesichtspunkte berücksichtigt werden.

Das vorgestellte OHMA-Verfahren wurde bezüglich seiner Effektivität und Effizienz untersucht (s. Abschnitt 6.3) und konnte vergleichbare Ergebnisse wie bekannte Verfahren [CCC14] liefern, die allerdings Applikationszugriff benötigen und auf Task-Scheduling basieren. Es wird eine Reduzierung der Leistungsaufnahme bis zu 50% innerhalb weniger Minuten ermöglicht. Ebenfalls konnte gezeigt werden, dass der vorgestellte Ansatz im Vergleich zum absoluten Verfahren nur zu marginalen Qualitätseinbußen (s. Abschnitt 6.4) führt.

Im Bereich des Rechenzentrumsbetriebs ist die Servervirtualisierung eine etablierte Basistechnologie. Weitere VMs in Betrieb zu nehmen, ist dabei eine Selbstverständlichkeit zur Skalierung von Systemen ebenso wie die Inbetriebnahme weiterer PMs. Die Architektur des vorgestellten Verfahrens unterstützt diese Skalierungsmöglichkeiten durch die dezentrale Arbeitsweise (s. Abschnitt 5.1.1) und ermöglicht das Hinzufügen und Entfernen von PMs und VMs im laufenden Betrieb. Damit das Verfahren insgesamt als skalierbar bezeichnet werden kann, darf eine mögliche Systemerweiterung nicht zu exponentiell steigenden Berechnungszeiten führen, es ist ein Verfahren mit linearer Komplexität nötig. Diese Eigenschaft des Verfahrens wurde in Abschnitt 6.2 untersucht, bestätigt und beantwortet Forschungsfrage Nr. 3 mit der Forderung nach einem skalierbaren Ansatz.

Die Architektur des Verfahrens basiert auf einem dezentralen Ansatz mit zwei Servergruppen. Kommt es zu einem Ausfall einzelner PMs, arbeiten die restlichen Server weiter

an der Erreichung der Ziel-Leistungsaufnahme. Darüber hinaus wird die Ziel-Leistungsaufnahme stets überwacht. Sollte es zu Verschiebungen bei der Ressourcen-Auslastung kommen, die eine Abweichung von der Ziel-Leistungsaufnahme nach sich zieht, werden korrigierende VM-Migrationen eingeleitet. Die gewählte Architektur sorgt damit für einen zuverlässigen, stabilen Betrieb auch bei einem Ausfall von einzelnen PMs.

Es könnte ein Verfahren implementiert werden, das alle in Abschnitt 4.6 genannten Anforderungen erfüllt. Das OHMA-Verfahren erweitert die energetischen Freiheitsgrade von Rechenzentren, indem mit der Technik einer Ressourcen-Überprovisionierung in Kombination von VM-Migrationen im Kontext des IaaS und SaaS eine Leistungssteuerung ermöglicht wird. Alle gestellten Forschungsfragen konnten beantwortet werden.

In Abschnitt 6.5 wurden Einsatzpotentiale des OHMA-Verfahrens am Beispiel von Fachapplikationen beschrieben. In dem folgenden Ausblick werden nun Erweiterungen und Integrationsmöglichkeiten aufgezeigt, die das Potential des Verfahrens erweitern können.

7.2. Ausblick

Aufbauend auf das in dieser Arbeit erstellte OHMA-Verfahren ließen sich bestehende Management-Systeme um diese Technik zu erweitern. Hier bietet sich die Integration auf dem Hypervisor der Open-Source-Plattform OpenStack [ope19] an, um bestehende Systeme zur Erfüllung einer geforderten Leistungsaufnahme zu befähigen.

Neben der Integration der Energielieferung von erneuerbaren Energien können weitere Stakeholder in das Lastmanagement integriert werden. Wird beispielsweise die Abwärme des Rechenzentrums genutzt, wird ein Algorithmus zur Entscheidung, welchen Energiezustand das Rechenzentrum annehmen soll, ungleich komplexer. Es muss eine Bewertung vorgenommen werden, welcher Zustand den besten ökonomischen oder ökologischen Gesamteffekt hat. In diesem konkreten Fall wäre zu bewerten, ob ein erhöhter Strompreis in Kauf genommen wird, um die geforderte Abwärme zu produzieren oder ob die benötigte Wärme alternativ über einen Gasbrenner zur Verfügung gestellt werden soll. Um eine Gewichtung dieser Faktoren vorzunehmen, könnte sich der Einsatz von künstlichen neuronalen Netzen in Kombination mit Fuzzylogic als sinnvoll erweisen.

Um das Volumen der beeinflussbaren Leistungsaufnahme weiter zu erhöhen, wäre die Softwarearchitektur zu betrachten. Hier könnte durch fachliches Priorisieren von Aufgaben eine Transparenz über die eingangs erwähnte Dringlichkeit (vgl. Abschnitt 1.1) von Aufgaben geschaffen werden. Mittels dieser Informationen ließe sich in Anlehnung an des Scheduling-Verfahren, wie in [KGA⁺11] beschrieben, ein weiterer Handlungsspielraum generieren.

Im Folgenden werden weitere Möglichkeiten zur Integration und Erweiterung des OHMA-Verfahrens beschrieben.

7.2.1. Integration von RAM-Ähnlichkeitsprüfungen in das OHMA-Verfahren

Ein beschränkender Faktor bei der Konsolidierung von VMs ist häufig der verfügbare Arbeitsspeicher. Wie die Analyse der Serverressourcen in Abschnitt 2.3 gezeigt hat, ist eine Überprovisionierung kaum sinnvoll.

Um die Ressource Arbeitsspeicher besser zu nutzen, gibt es Verfahren wie [WTLS⁺09], die Ähnlichkeitsanalysen auf Arbeitsspeicherseiten durchführen und es ermöglichen, Seiten mit identischem Inhalt zu konsolidieren und über Zeiger zur Verfügung zu stellen. Das hier vorgestellte Verfahren könnte nun um eine vorgelagerte Ähnlichkeitsprüfung des Arbeitsspeichers erweitert werden. Bei der Auswahl einer VM zur Migration kann ihr Arbeitsspeicherinhalt geprüft, das Konsolidierungspotential ermittelt und bei der Auswahl berücksichtigt werden. Gegebenenfalls sind so weitere Migrationen von VMs möglich, die das Leistungsspektrum des Verfahrens erweitern.

7.2.2. Adaption zur Nutzung von Docker-Containern

Der Grund, weshalb für verschiedene Applikationen jeweils eigene Server verwendet werden, liegt häufig in der Möglichkeit begründet, eine technische Trennung herbeiführen zu können. Es besteht der Wunsch nach einer Trennung von Ressourcen, Rechten, Daten und Verwaltung, um eine gegenseitige Einflussnahme ausschließen zu können. Gewünscht ist eine möglichst abgeschlossene, autonome Verwaltungseinheit. Aktuell werden neue Betriebskonzepte wie Docker vermehrt für diese Zwecke evaluiert und eingesetzt. Ein Docker-Container bietet genau diese Möglichkeiten der Trennung.

In einer Docker-Umgebung kann eine Applikationsinstanz als Container direkt auf dem Zielsystem gestartet werden. Sobald diese den Service übernommen hat, kann der Container auf dem ursprünglichen System gestoppt werden. Der Schritt einer Migration würde somit entfallen. Es ist zu untersuchen, wie das vorgestellte Verfahren auf diese Technologie adaptiert werden kann. Ein Open-Source-System, welches die Verwaltung und Skalierung von Container-Anwendungen anbietet, ist Kubernetes [MRBA16]. Eine Integration in dieses System kann eine sinnvolle Erweiterung für Kubernetes darstellen.

7.2.3. Integration thermischer Effekte

Eine Änderung der Server-Leistungsaufnahme bewirkt auch eine Änderung der abgegebenen Wärmemenge. Dieser Umstand zieht wiederum einen veränderten Klimatisierungsbedarf nach sich. Diese Effekte sind mit entsprechenden Latenzen behaftet und wirken sich verzögert auf die Leistungsaufnahme des Rechenzentrums aus. Eine sinnvolle Erweiterung des hier vorgestellten Verfahrens wäre daher ein weiteres Modul, das diese Effekte bei der VM-Migration bereits berücksichtigt und die Migrationsstrategie entsprechend anpasst.

7.2.4. Weiteres Nutzungspotential durch Microservices

Ein aktueller Software-Entwicklungstrend geht in Richtung der Entwicklung von Microservices. Dabei werden keine monolithischen Applikationen mehr entwickelt und betrieben, sondern eine Orchestrierung von diversen Microservices. Jeder Microservice stellt seinem Namen entsprechend nur eine sehr kleine, überschaubare Funktionalität zur Verfügung. Mit dieser Architektur soll die Komplexität der einzelnen Services gering gehalten und damit eine flexible und schnelle Anpassung ermöglicht werden.

Durch die Segmentierung einer Applikation in Microservices kann eine Bewertung der Microservices hinsichtlich Kritikalität und Bedarf an synchronem Antwortzeitverhalten vorgenommen werden. Microservices, die bezüglich Realzeitanforderungen anspruchslos sind, könnten so separiert und auf entsprechender Infrastruktur in Kombination mit dem vorgestellten Verfahren betrieben werden. Die Komplexität der Einhaltung von SLAs würde so reduziert.

A. Anhang

Die in diesem Abschnitt dargestellten Tabellen enthalten die für Simulationen in Abschnitt 6 verwendeten VM-Profil Datensätze und die PM-Zuordnung der Start-Konfiguration.

Tabelle A.1.: Verwendete VM-Datensätze und Start-Konfiguration für Simulationen von 40 VMs

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-0	10,4	16.777.216	97,83	30,13	PM-0
VM-1	55,24	4.194.304	493,09	102,91	PM-0
VM-2	25,07	4.194.304	79,3	28,64	PM-0
VM-3	2,36	3.145.728	1,45	4,33	PM-0
VM-4	1,66	2.097.152	1,44	1,8	PM-0
VM-5	3,78	2.097.152	3,55	9,58	PM-0
VM-6	10,84	10.097.152	157,54	26,95	PM-1
VM-7	4,02	8.388.608	42,16	18,1	PM-1
VM-8	1,18	8.388.608	0,002	1,6	PM-1
VM-9	57,6	4.194.304	482,26	100,64	PM-1
VM-10	11,44	4.194.304	77,56	18,87	PM-1
VM-11	13,92	4.194.304	894,48	270,28	PM-1
VM-12	11,48	10.097.152	69,94	23,45	PM-2
VM-13	87,72	10.097.152	744,15	408,02	PM-2
VM-14	4,74	16.777.216	23,58	8,35	PM-3
VM-15	2,32	8.388.608	29,94	6,87	PM-3
VM-16	62,7	4.194.304	141,17	39,43	PM-3
VM-17	23,59	1.572.864	875,16	138,07	PM-3
VM-18	4,91	1.097.152	69,94	13,73	PM-3
VM-19	2,67	1.097.152	0,0081	1,33	PM-3
VM-20	10,4	16.777.216	97,83	30,13	PM-4
VM-21	55,24	4.194.304	493,09	102,91	PM-4
VM-22	25,07	4.194.304	79,3	28,64	PM-4
VM-23	2,36	3.145.728	1,45	4,33	PM-4
VM-24	1,66	2.097.152	1,44	1,8	PM-4
VM-25	3,78	2.097.152	3,55	9,58	PM-4
VM-26	10,84	10.097.152	157,54	26,95	PM-5
VM-27	4,02	8.388.608	42,16	18,1	PM-5
VM-28	1,18	8.388.608	0,002	1,6	PM-5

Fortsetzung auf der nächsten Seite

A. Anhang

Tabelle A.1 – Fortsetzung

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-29	57,6	4.194.304	482,26	100,64	PM-5
VM-30	11,44	4.194.304	77,56	18,87	PM-5
VM-31	13,92	4.194.304	894,48	270,28	PM-5
VM-32	11,48	10.097.152	69,94	23,45	PM-6
VM-33	87,72	10.097.152	744,15	408,02	PM-6
VM-34	4,74	16.777.216	23,58	8,35	PM-7
VM-35	2,32	8.388.608	29,94	6,87	PM-7
VM-36	62,7	4.194.304	141,17	39,43	PM-7
VM-37	23,59	1.572.864	875,16	138,07	PM-7
VM-38	4,91	1.097.152	69,94	13,73	PM-7
VM-39	2,67	1.097.152	0,0081	1,33	PM-7

Tabelle A.2.: Verwendete VM-Datensätze und Start-Konfiguration für Simulationen von 60 VMs

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-0	10,4	16.777.216	97,83	30,13	PM-0
VM-1	55,24	4.194.304	493,09	102,91	PM-0
VM-2	25,07	4.194.304	79,3	28,64	PM-0
VM-3	2,36	3.145.728	1,45	4,33	PM-0
VM-4	1,66	2.097.152	1,44	1,8	PM-0
VM-5	3,78	2.097.152	3,55	9,58	PM-0
VM-6	10,84	10.097.152	157,54	26,95	PM-1
VM-7	4,02	8.388.608	42,16	18,1	PM-1
VM-8	1,18	8.388.608	0,002	1,6	PM-1
VM-9	57,6	4.194.304	482,26	100,64	PM-1
VM-10	11,44	4.194.304	77,56	18,87	PM-1
VM-11	13,92	4.194.304	894,48	270,28	PM-1
VM-12	11,48	10.097.152	69,94	23,45	PM-2
VM-13	87,72	10.097.152	744,15	408,02	PM-2
VM-14	4,74	16.777.216	23,58	8,35	PM-3
VM-15	2,32	8.388.608	29,94	6,87	PM-3
VM-16	62,7	4.194.304	141,17	39,43	PM-3
VM-17	23,59	1.572.864	875,16	138,07	PM-3
VM-18	4,91	1.097.152	69,94	13,73	PM-3
VM-19	2,67	1.097.152	0,0081	1,33	PM-3
VM-20	10,4	16.777.216	97,83	30,13	PM-4
VM-21	55,24	4.194.304	493,09	102,91	PM-4
VM-22	25,07	4.194.304	79,3	28,64	PM-4

Fortsetzung auf der nächsten Seite

Tabelle A.2 – Fortsetzung

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-23	2,36	3.145.728	1,45	4,33	PM-4
VM-24	1,66	2.097.152	1,44	1,8	PM-4
VM-25	3,78	2.097.152	3,55	9,58	PM-4
VM-26	10,84	10.097.152	157,54	26,95	PM-5
VM-27	4,02	8.388.608	42,16	18,1	PM-5
VM-28	1,18	8.388.608	0,002	1,6	PM-5
VM-29	57,6	4.194.304	482,26	100,64	PM-5
VM-30	11,44	4.194.304	77,56	18,87	PM-5
VM-31	13,92	4.194.304	894,48	270,28	PM-5
VM-32	11,48	10.097.152	69,94	23,45	PM-6
VM-33	87,72	10.097.152	744,15	408,02	PM-6
VM-34	4,74	16.777.216	23,58	8,35	PM-7
VM-35	2,32	8.388.608	29,94	6,87	PM-7
VM-36	62,7	4.194.304	141,17	39,43	PM-7
VM-37	23,59	1.572.864	875,16	138,07	PM-7
VM-38	4,91	1.097.152	69,94	13,73	PM-7
VM-39	2,67	1.097.152	0,0081	1,33	PM-7
VM-40	10,4	16.777.216	97,83	30,13	PM-8
VM-41	55,24	4.194.304	493,09	102,91	PM-8
VM-42	25,07	4.194.304	79,3	28,64	PM-8
VM-43	2,36	3.145.728	1,45	4,33	PM-8
VM-44	1,66	2.097.152	1,44	1,8	PM-8
VM-45	3,78	2.097.152	3,55	9,58	PM-8
VM-46	10,84	10.097.152	157,54	26,95	PM-9
VM-47	4,02	8.388.608	42,16	18,1	PM-9
VM-48	1,18	8.388.608	0,002	1,6	PM-9
VM-49	57,6	4.194.304	482,26	100,64	PM-9
VM-50	11,44	4.194.304	77,56	18,87	PM-9
VM-51	13,92	4.194.304	894,48	270,28	PM-9
VM-52	11,48	10.097.152	69,94	23,45	PM-10
VM-53	87,72	10.097.152	744,15	408,02	PM-10
VM-54	4,74	16.777.216	23,58	8,35	PM-11
VM-55	2,32	8.388.608	29,94	6,87	PM-11
VM-56	62,7	4.194.304	141,17	39,43	PM-11
VM-57	23,59	1.572.864	875,16	138,07	PM-11
VM-58	4,91	1.097.152	69,94	13,73	PM-11
VM-59	2,67	1.097.152	0,0081	1,33	PM-11

A. Anhang

Tabelle A.3.: Verwendete VM-Datensätze und Start-Konfigurationen für Simulationen von 20, 40, 60, 80, 100 und 160 VMs

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-0	10,4	16.777.216	97,83	30,13	PM-0
VM-1	55,24	4.194.304	493,09	102,91	PM-0
VM-2	25,07	4.194.304	79,3	28,64	PM-0
VM-3	2,36	3.145.728	1,45	4,33	PM-0
VM-4	1,66	2.097.152	1,44	1,8	PM-0
VM-5	3,78	2.097.152	3,55	9,58	PM-0
VM-6	10,84	10.097.152	157,54	26,95	PM-1
VM-7	4,02	8.388.608	42,16	18,1	PM-1
VM-8	1,18	8.388.608	0,002	1,6	PM-1
VM-9	57,6	4.194.304	482,26	100,64	PM-1
VM-10	11,44	4.194.304	77,56	18,87	PM-1
VM-11	13,92	4.194.304	894,48	270,28	PM-1
VM-12	11,48	10.097.152	69,94	23,45	PM-2
VM-13	87,72	10.097.152	744,15	408,02	PM-2
VM-14	4,74	16.777.216	23,58	8,35	PM-3
VM-15	2,32	8.388.608	29,94	6,87	PM-3
VM-16	62,7	4.194.304	141,17	39,43	PM-3
VM-17	23,59	1.572.864	875,16	138,07	PM-3
VM-18	4,91	1.097.152	69,94	13,73	PM-3
VM-19	2,67	1.097.152	0,0081	1,33	PM-3
VM-20	10,4	16.777.216	97,83	30,13	PM-4
VM-21	55,24	4.194.304	493,09	102,91	PM-4
VM-22	25,07	4.194.304	79,3	28,64	PM-4
VM-23	2,36	3.145.728	1,45	4,33	PM-4
VM-24	1,66	2.097.152	1,44	1,8	PM-4
VM-25	3,78	2.097.152	3,55	9,58	PM-4
VM-26	10,84	10.097.152	157,54	26,95	PM-5
VM-27	4,02	8.388.608	42,16	18,1	PM-5
VM-28	1,18	8.388.608	0,002	1,6	PM-5
VM-29	57,6	4.194.304	482,26	100,64	PM-5
VM-30	11,44	4.194.304	77,56	18,87	PM-5
VM-31	13,92	4.194.304	894,48	270,28	PM-5
VM-32	11,48	10.097.152	69,94	23,45	PM-6
VM-33	87,72	10.097.152	744,15	408,02	PM-6
VM-34	4,74	16.777.216	23,58	8,35	PM-7
VM-35	2,32	8.388.608	29,94	6,87	PM-7
VM-36	62,7	4.194.304	141,17	39,43	PM-7
VM-37	23,59	1.572.864	875,16	138,07	PM-7
VM-38	4,91	1.097.152	69,94	13,73	PM-7
VM-39	2,67	1.097.152	0,0081	1,33	PM-7

Fortsetzung auf der nächsten Seite

Tabelle A.3 – Fortsetzung

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-40	10,4	16.777.216	97,83	30,13	PM-8
VM-41	55,24	4.194.304	493,09	102,91	PM-8
VM-42	25,07	4.194.304	79,3	28,64	PM-8
VM-43	2,36	3.145.728	1,45	4,33	PM-8
VM-44	1,66	2.097.152	1,44	1,8	PM-8
VM-45	3,78	2.097.152	3,55	9,58	PM-8
VM-46	10,84	10.097.152	157,54	26,95	PM-9
VM-47	4,02	8.388.608	42,16	18,1	PM-9
VM-48	1,18	8.388.608	0,002	1,6	PM-9
VM-49	57,6	4.194.304	482,26	100,64	PM-9
VM-50	11,44	4.194.304	77,56	18,87	PM-9
VM-51	13,92	4.194.304	894,48	270,28	PM-9
VM-52	11,48	10.097.152	69,94	23,45	PM-10
VM-53	87,72	10.097.152	744,15	408,02	PM-10
VM-54	4,74	16.777.216	23,58	8,35	PM-11
VM-55	2,32	8.388.608	29,94	6,87	PM-11
VM-56	62,7	4.194.304	141,17	39,43	PM-11
VM-57	23,59	1.572.864	875,16	138,07	PM-11
VM-58	4,91	1.097.152	69,94	13,73	PM-11
VM-59	2,67	1.097.152	0,0081	1,33	PM-11
VM-60	10,4	16.777.216	97,83	30,13	PM-12
VM-61	55,24	4.194.304	493,09	102,91	PM-12
VM-62	25,07	4.194.304	79,3	28,64	PM-12
VM-63	2,36	3.145.728	1,45	4,33	PM-12
VM-64	1,66	2.097.152	1,44	1,8	PM-12
VM-65	3,78	2.097.152	3,55	9,58	PM-12
VM-66	10,84	10.097.152	157,54	26,95	PM-13
VM-67	4,02	8.388.608	42,16	18,1	PM-13
VM-68	1,18	8.388.608	0,002	1,6	PM-13
VM-69	57,6	4.194.304	482,26	100,64	PM-13
VM-70	11,44	4.194.304	77,56	18,87	PM-13
VM-71	13,92	4.194.304	894,48	270,28	PM-13
VM-72	11,48	10.097.152	69,94	23,45	PM-14
VM-73	87,72	10.097.152	744,15	408,02	PM-14
VM-74	4,74	16.777.216	23,58	8,35	PM-15
VM-75	2,32	8.388.608	29,94	6,87	PM-15
VM-76	62,7	4.194.304	141,17	39,43	PM-15
VM-77	23,59	1.572.864	875,16	138,07	PM-15
VM-78	4,91	1.097.152	69,94	13,73	PM-15
VM-79	2,67	1.097.152	0,0081	1,33	PM-15
VM-80	10,4	16.777.216	97,83	30,13	PM-16
VM-81	55,24	4.194.304	493,09	102,91	PM-16

Fortsetzung auf der nächsten Seite

Tabelle A.3 – Fortsetzung

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-82	25,07	4.194.304	79,3	28,64	PM-16
VM-83	2,36	3.145.728	1,45	4,33	PM-16
VM-84	1,66	2.097.152	1,44	1,8	PM-16
VM-85	3,78	2.097.152	3,55	9,58	PM-16
VM-86	10,84	10.097.152	157,54	26,95	PM-17
VM-87	4,02	8.388.608	42,16	18,1	PM-17
VM-88	1,18	8.388.608	0,002	1,6	PM-17
VM-89	57,6	4.194.304	482,26	100,64	PM-17
VM-90	11,44	4.194.304	77,56	18,87	PM-17
VM-91	13,92	4.194.304	894,48	270,28	PM-17
VM-92	11,48	10.097.152	69,94	23,45	PM-18
VM-93	87,72	10.097.152	744,15	408,02	PM-18
VM-94	4,74	16.777.216	23,58	8,35	PM-19
VM-95	2,32	8.388.608	29,94	6,87	PM-19
VM-96	62,7	4.194.304	141,17	39,43	PM-19
VM-97	23,59	1.572.864	875,16	138,07	PM-19
VM-98	4,91	1.097.152	69,94	13,73	PM-19
VM-99	2,67	1.097.152	0,0081	1,33	PM-19
VM-100	10,4	16.777.216	97,83	30,13	PM-20
VM-101	55,24	4.194.304	493,09	102,91	PM-20
VM-102	25,07	4.194.304	79,3	28,64	PM-20
VM-103	2,36	3.145.728	1,45	4,33	PM-20
VM-104	1,66	2.097.152	1,44	1,8	PM-20
VM-105	3,78	2.097.152	3,55	9,58	PM-20
VM-106	10,84	10.097.152	157,54	26,95	PM-21
VM-107	4,02	8.388.608	42,16	18,1	PM-21
VM-108	1,18	8.388.608	0,002	1,6	PM-21
VM-109	57,6	4.194.304	482,26	100,64	PM-21
VM-110	11,44	4.194.304	77,56	18,87	PM-21
VM-111	13,92	4.194.304	894,48	270,28	PM-21
VM-112	11,48	10.097.152	69,94	23,45	PM-22
VM-113	87,72	10.097.152	744,15	408,02	PM-22
VM-114	4,74	16.777.216	23,58	8,35	PM-23
VM-115	2,32	8.388.608	29,94	6,87	PM-23
VM-116	62,7	4.194.304	141,17	39,43	PM-23
VM-117	23,59	1.572.864	875,16	138,07	PM-23
VM-118	4,91	1.097.152	69,94	13,73	PM-23
VM-119	2,67	1.097.152	0,0081	1,33	PM-23
VM-120	10,4	16.777.216	97,83	30,13	PM-24
VM-121	55,24	4.194.304	493,09	102,91	PM-24
VM-122	25,07	4.194.304	79,3	28,64	PM-24
VM-123	2,36	3.145.728	1,45	4,33	PM-24

Fortsetzung auf der nächsten Seite

Tabelle A.3 – Fortsetzung

VM	CPU-Usage [%]	Ram [KB]	Network [KB/s]	Disk [IOPS]	PM
VM-124	1,66	2.097.152	1,44	1,8	PM-24
VM-125	3,78	2.097.152	3,55	9,58	PM-24
VM-126	10,84	10.097.152	157,54	26,95	PM-25
VM-127	4,02	8.388.608	42,16	18,1	PM-25
VM-128	1,18	8.388.608	0,002	1,6	PM-25
VM-129	57,6	4.194.304	482,26	100,64	PM-25
VM-130	11,44	4.194.304	77,56	18,87	PM-25
VM-131	13,92	4.194.304	894,48	270,28	PM-25
VM-132	11,48	10.097.152	69,94	23,45	PM-26
VM-133	87,72	10.097.152	744,15	408,02	PM-26
VM-134	4,74	16.777.216	23,58	8,35	PM-27
VM-135	2,32	8.388.608	29,94	6,87	PM-27
VM-136	62,7	4.194.304	141,17	39,43	PM-27
VM-137	23,59	1.572.864	875,16	138,07	PM-27
VM-138	4,91	1.097.152	69,94	13,73	PM-27
VM-139	2,67	1.097.152	0,0081	1,33	PM-27
VM-140	10,4	16.777.216	97,83	30,13	PM-28
VM-141	55,24	4.194.304	493,09	102,91	PM-28
VM-142	25,07	4.194.304	79,3	28,64	PM-28
VM-143	2,36	3.145.728	1,45	4,33	PM-28
VM-144	1,66	2.097.152	1,44	1,8	PM-28
VM-145	3,78	2.097.152	3,55	9,58	PM-28
VM-146	10,84	10.097.152	157,54	26,95	PM-29
VM-147	4,02	8.388.608	42,16	18,1	PM-29
VM-148	1,18	8.388.608	0,002	1,6	PM-29
VM-149	57,6	4.194.304	482,26	100,64	PM-29
VM-150	11,44	4.194.304	77,56	18,87	PM-29
VM-151	13,92	4.194.304	894,48	270,28	PM-29
VM-152	11,48	10.097.152	69,94	23,45	PM-30
VM-153	87,72	10.097.152	744,15	408,02	PM-30
VM-154	4,74	16.777.216	23,58	8,35	PM-31
VM-155	2,32	8.388.608	29,94	6,87	PM-31
VM-156	62,7	4.194.304	141,17	39,43	PM-31
VM-157	23,59	1.572.864	875,16	138,07	PM-31
VM-158	4,91	1.097.152	69,94	13,73	PM-31
VM-159	2,67	1.097.152	0,0081	1,33	PM-31

Abbildungsverzeichnis

1.1.	Energie-Informationssäule am Campus Cloppenburger Straße der EWE AG	4
1.2.	Nutzungsszenario für regelbare Leistungsaufnahme von Servern in einem Rechenzentrum	6
2.1.	Anteilige Darstellung des Energiebedarfs in Rechenzentren; eigene Darstellung, Daten entnommen aus [DRTB14]	12
2.2.	Schematisches Blockschaltbild einer Online-USV	13
2.3.	Gesamtübersicht eines Klimatisierungsstrangs mit Einsatz von Kompressoren; Grafik unverändert übernommen aus [ZWZ17]	14
2.4.	Benchmark-Ergebnisse eines Servers mit dynamischer Leistungsaufnahme; Grafik unverändert aus [Sta16] übernommen	16
2.5.	Durchschnittliche Leistungsaufnahme von Quell- und Ziel-Server während der Migration, verglichen mit der Idle-Leistungsaufnahme (horizontale Linie); Grafik unverändert aus [SD13] zitiert	23
2.6.	Benchmark des Servers Fujitsu PRIMERGY TX2540 M1; Grafik unverändert aus [Sta15] übernommen	25
2.7.	Benchmark des Servers Sun Netra X4250; Grafik unverändert aus [Sta09] übernommen	26
2.8.	Zeitliche Abfolge der Aktivierung von Regelleistung	27
3.1.	Leistungsaufnahme eines Servers in Abhängigkeit zur Auslastung; Grafik unverändert aus [Sta16] übernommen	33
4.1.	Ausgangssituation einer VM-Verteilung auf physikalische Server mit Hypervisor	44
4.2.	Durch Permutationen von VMs geänderte Ressourcenauslastung	44
4.3.	Performance-Ergebnisse aus Experimenten mit Überprovisionierung der CPU-Ressourcen	45
5.1.	Komponenten des Systems: Überwachung der Systemdaten	57
5.2.	Komponenten des Systems: Propagieren der Ziel-Leistungsaufnahme	57
5.3.	Komponenten des Systems: Austausch von Kenndaten der betriebenen VMs	58
5.4.	Aufteilung der Server in zwei Gruppen	59
5.5.	Server mit geringem adaptiven Verhalten bezüglich der Leistungsaufnahme; Grafik unverändert aus [Sta09] übernommen	60
5.6.	Server mit adaptivem Verhalten bezüglich der Leistungsaufnahme; Grafik unverändert aus [Sta16] übernommen	61

Abbildungsverzeichnis

5.9. Beispiel für die Modellierung eines Bahnübergangs; Screenshot eines mit Uppaal ausgelieferten Implementierungsbeispiels	64
5.10. Hauptfenster des Simulationssystems Uppaal	65
5.11. Automaten-Template einer VM; Original-Ansicht in dem grafischen Editor des Uppaal-Frameworks	66
5.12. Uppaal-Template einer PM	69
5.13. Schematische Darstellung des Automaten-Templates zur Simulation einer PM	70
5.14. Darstellung der Prozessschritte des OHMA-Verfahrens in einem Flussdiagramm	81
5.15. Alternierende Aufteilung der Servergruppen	82
5.16. Aufteilung der Servergruppen bei individueller Klimatisierung der Racks .	83
5.17. Ergebnisse aus der Bestimmung eines Hysteresewertes für ein konkretes Szenario	85
5.18. Konvergenzverhalten bei zu klein gewählter Hysterese	87
5.19. Konvergenzverhalten bei passendem Hysteresewert	88
6.1. Ergebnisse von Simulationen mit unterschiedlichen Zielwert-Vorgaben zum Absenken der Leistungsaufnahme	92
6.2. Ergebnisse von Simulationen mit unterschiedlichen Zielwert-Vorgaben und Serverabschaltung zum Absenken der Leistungsaufnahme	93
6.3. Verlauf der Leistungsaufnahme während des Absenkens auf einen Zielwert von 50% der Maximalleistung	94
6.4. Ergebnisse von Simulationen mit unterschiedlichen Zielwert-Vorgaben und Serverabschaltung zur Erhöhung der Leistungsaufnahme	95
6.5. Berechnungsdauer des OHMA-Verfahrens zur Auswahl von VMs	96
6.6. Effizienz beim Absenken der Leistungsaufnahme, Ergebnisse aus 10 Simulationen je Szenario	98
6.7. Durchschnittliche Effizienz der Migrationen beim Absenken der Leistungsaufnahme	98
6.8. Effizienz beim Erhöhen der Leistungsaufnahme	99
6.9. Durchschnittliche Effizienz der Migrationen beim Erhöhen der Leistungsaufnahme	100
6.10. Effektivität beim Absenken der Leistungsaufnahme	101
6.11. Effektivität beim Erhöhen der Leistungsaufnahme	102
6.12. Leistungsaufnahme mit Hysterese von 1,25%	103
6.13. Anzahl benötigter Migrationen mit Hysterese von 1,25%	104
6.14. Leistungsaufnahme mit Hysterese von 3%	105
6.15. Migrationen mit Hysterese von 3%	105
6.16. Lastprofil eines Billing-Laufs	107

Stichwortverzeichnis

- Überprovisionierung, 45
- agnostisch, 42
- Allokation, 48
- applikationsinvasiv, 4
- Behälterproblem, 47
- CPU, 16
- Diversität, 76
- DVFS, 17, 59
- Forschungsfrage, 8, 37, 41, 113
- Freiheitsgrad, 45
- Hotspots, 15
- Hypervisor, 21
- Hysterese, 83
- IaaS, 5
- IT-Last, 42
- Klimatisierung, 14
- Leistungsaufnahme, 12
- Live-Migration, 22, 61
- Microservice, 62
- Migrationsverfahren, 21
- NEA, 12
- Offline-Migration, 21
- Offline-Problem, 56
- Online Verfahren, 56
- PM, 21
- PUE, 23
- Rechenzentrum, 11
- Regelleistung, 24
- SaaS, 5
- SAN, 19
- Servervirtualisierung, 20
- SLA, 23
- SLM, 23
- SLO, 23
- Smart-Grid, 4
- SSD, 19
- Stirling-Zahl, 49
- Throttling, 32
- Uppaal, 63
- USV, 12
- VM, 21

Literaturverzeichnis

- [50H19] 50HERTZ TRANSMISSION GMBH ; AMPRION GMBH ; TENNET TSO GMBH ; TRANSNETBW GMBH: *Präqualifikation für die Vorhaltung und Erbringung von Regelreserve*. <https://www.regelleistung.net/ext/static/prequalification>. Version: 2019. – letzter Zugriff 17.04.2019
- [Adv00] ADVANCED MICRO DEVICES: *AMD PowerNow Technology*. <http://www.amd-k6.com/wp-content/uploads/2012/07/24404a.pdf>. Version: November 2000. – letzter Zugriff 04.12.2020
- [Alb10] ALBERS, Susanne: Energy-Efficient Algorithms. In: *Communications of the ACM* 53 (2010), Mai, Nr. 5, 86-96. <http://dx.doi.org/10.1145/1735223.1735245>. – DOI 10.1145/1735223.1735245. – ISSN 0001-0782
- [Ama19] AMASOL: *Apptio Digital Fuel ITBM - Transparenz und Kontrolle der IT-Service-Kosten und Service-Qualität*. <https://amasol.de/partner-und-produkte/apptio/digital-fuel-itbm.html>. Version: 2019. – letzter Zugriff 10.04.2019
- [AVZR11] AKSANLI, Baris ; VENKATESH, Jagannathan ; ZHANG, Liuyi ; ROSING, Tajana: Utilizing Green Energy Prediction to Schedule Mixed Batch and Service Jobs in Data Centers. In: *Proceedings of the 4th Workshop on Power-Aware Computing and Systems*. New York, NY, USA : ACM, 2011 (HotPower '11). – ISBN 978-1-4503-0981-3, 5:1-5:5
- [BAB12] BELOGLAZOV, Anton ; ABAWAJY, Jemal ; BUYYA, Rajkumar: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. In: *Future generation computer systems* 28 (2012), Nr. 5, S. 755-768
- [BB10] BELOGLAZOV, Anton ; BUYYA, Rajkumar: Adaptive Threshold-based Approach for Energy-efficient Consolidation of Virtual Machines in Cloud Data Centers. In: *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*. New York, NY, USA : ACM, 2010 (MGC '10). – ISBN 978-1-4503-0453-5, 4:1-4:6
- [BB12] BELOGLAZOV, Anton ; BUYYA, Rajkumar: Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers. In: *Concurr. Comput. : Pract. Exper.* 24 (2012), September, Nr. 13, 1397-1420. <http://dx.doi.org/10.1002/cpe.1867>. – DOI 10.1002/cpe.1867. – ISSN 1532-0626

Literaturverzeichnis

- [BB15] BELOGLAZOV, Anton ; BUYYA, Rajkumar: OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds. In: *Concurrency and Computation: Practice and Experience* 27 (2015), Nr. 5, S. 1310–1333
- [Bel13] BELOGLAZOV, Anton: *Energy-efficient management of virtual machines in data centers for cloud computing*, University of Melbourne, Diss., 2013
- [BF07] BASH, Cullen ; FORMAN, George: Cool Job Allocation: Measuring the Power Savings of Placing Jobs at Cooling-Efficient Locations in the Data Center. In: *USENIX Annual Technical Conference* Bd. 138, 2007, S. 140
- [BHJ⁺18] BALALAIE, Armin ; HEYDARNOORI, Abbas ; JAMSHIDI, Pooyan ; TAMBURRI, Damian ; LYNN, Theodore: Microservices migration patterns. In: *Software: Practice and Experience* 48 (2018), 07. <http://dx.doi.org/10.1002/spe.2608>. – DOI 10.1002/spe.2608
- [BKB07] BOBROFF, N. ; KOCHUT, A. ; BEATY, K.: Dynamic Placement of Virtual Machines for Managing SLA Violations. In: *2007 10th IFIP/IEEE International Symposium on Integrated Network Management*, 2007. – ISSN 1573–0077, S. 119–128
- [BPS06] BASH, C. B. ; PATEL, C. D. ; SHARMA, R. K.: Dynamic thermal management of air cooled data centers. In: *Thermal and Thermomechanical Proceedings 10th Intersociety Conference on Phenomena in Electronics Systems*, 2006, S. 8 pp.
- [BR11] BROWN, Michael ; RENAU, Jose: ReRack: power simulation for data centers with renewable energy generation. In: *SIGMETRICS Performance Evaluation Review* 39 (2011), Nr. 3, 77-81. <http://dblp.uni-trier.de/db/journals/sigmetrics/sigmetrics39.html#BrownR11>
- [BR17] BORGERDING, Alexander ; ROSINGER, Sven: Extending Energetic Potential of Data Centers to Participate in Smart Grid Networks. In: *Smart Energy Research. At the Crossroads of Engineering, Economics, and Computer Science*. Springer, 2017, S. 107–120
- [BS16] BORGERDING, Alexander ; SCHOMAKER, Gunnar: Extending energetic potentials of data centers by resource optimization to improve carbon footprint. In: *Advances and New Trends in Environmental and Energy Informatics*. Springer, 2016, S. 3–19
- [Bun12] BUNDESNETZAGENTUR: *Positionspapier zu den technischen Vorgaben nach § 6 Abs. 1 und Abs. 2 EEG 2012*. https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Energie/Unternehmen_Institutionen/ErneuerbareEnergien/EEG/TechnischeVorgaben6EEG/Positionspapier_TechVorg_6EEG.pdf?__blob=publicationFile&v=2. Version: Dezember 2012. – letzter Zugriff 04.12.2020
- [Bun19] BUNDESNETZAGENTUR: *Regelenergie*. https://www.bundesnetzagentur.de/DE/Sachgebiete/ElektrizitaetundGas/Unternehmen_

Institutionen/Versorgungssicherheit/Engpassmanagement/
Regelenergie/regelenergie-node.html. Version: 2019. – letzter
Zugriff 17.04.2019

- [BWSS12] BARKER, Sean ; WOOD, Timothy ; SHENOY, Prashant ; SITARAMAN, Ramesh: An Empirical Study of Memory Sharing in Virtual Machines. In: *USENIX Annual Technical Conference (USENIX ATC 12)*, 2012, 273–284
- [CBJ10] COOPER-BALIS, Elliott ; JACOB, Bruce: Fine-grained activation for power reduction in DRAM. In: *IEEE Micro* 30 (2010), Nr. 3, S. 34–47
- [CCC14] CHEN, H. ; CARAMANIS, M. C. ; COSKUN, A. K.: The data center as a grid load stabilizer. In: *2014 19th Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2014. – ISSN 2153–6961, S. 105–112
- [CFF12] CORRADI, A. ; FANELLI, M. ; FOSCHINI, L.: VM Consolidation: a Real Case Based on OpenStack Cloud. In: *Elsevier Future Generation Computer Systems* (2012). <http://dx.doi.org/10.1016/j.future.2012.05.012>. – DOI 10.1016/j.future.2012.05.012
- [CFH⁺05] CLARK, Christopher ; FRASER, Keir ; HAND, Steven ; HANSEN, Jacob G. ; JUL, Eric ; LIMPACH, Christian ; PRATT, Ian ; WARFIELD, Andrew: Live Migration of Virtual Machines. In: *Proceedings of the 2Nd Conference on Symposium on Networked Systems Design & Implementation - Volume 2*. Berkeley, CA, USA : USENIX Association, 2005 (NSDI'05), 273–286
- [CGJ⁺10] CHOI, Jeonghwan ; GOVINDAN, Sriram ; JEONG, Jinkyu ; URGANONKAR, Bhuvan ; SIVASUBRAMANIAM, Anand: Power Consumption Prediction and Power-Aware Packing in Consolidated Environments. In: *IEEE Transactions on Computers* 59 (2010), Nr. 12, S. 1640–1654
- [CHL⁺08] CHEN, Gong ; HE, Wenbo ; LIU, Jie ; NATH, Suman ; RIGAS, Leonidas ; XIAO, Lin ; ZHAO, Feng: Energy-aware Server Provisioning and Load Dispatching for Connection-intensive Internet Services. In: *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*. Berkeley, CA, USA : USENIX Association, 2008 (NSDI'08). – ISBN 111–999–5555–22–1, 337–350
- [CHT12] CHEN, Changbing ; HE, Bingsheng ; TANG, Xueyan: Green-aware workload scheduling in geographically distributed data centers. In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings* (2012), S. 82–89. <http://dx.doi.org/http://doi.ieeecomputersociety.org/10.1109/CloudCom.2012.6427545>. – DOI <http://doi.ieeecomputersociety.org/10.1109/CloudCom.2012.6427545>. ISBN 978–1–4673–4511–8
- [Con10] CONSENTEC GMBH: *Gutachten zur Dimensionierung des Regelleistungbedarfs unter dem NRV*. <https://www.regelleistung.net/ext/download/dimensionierungBnetza>. Version: 2010. – letzter Zugriff 02.12.2020

Literaturverzeichnis

- [Con14] CONSENTEC GMBH: *Beschreibung von Regelleistungskonzepten und Regelleistungsmarkt*. <https://www.regelleistung.net/ext/download/marktbeschreibung>. Version: 2014. – letzter Zugriff 02.12.2020
- [CRB⁺11] CALHEIROS, Rodrigo N. ; RANJAN, Rajiv ; BELOGLAZOV, Anton ; DE ROSE, César A. F. ; BUYYA, Rajkumar: CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. In: *Software: Practice and experience* 41 (2011), Januar, Nr. 1, 23–50. <http://dx.doi.org/10.1002/spe.995>. – DOI 10.1002/spe.995. – ISSN 0038–0644
- [DGC13] DALVANDI, A ; GURUSAMY, M. ; CHUA, Kee C.: Time-Aware VM-Placement and Routing with Bandwidth Guarantees in Green Cloud Data Centers. In: *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on* Bd. 1, 2013, S. 212–217
- [DILS09] DAVID, Alexandre ; ILLUM, Jacob ; LARSEN, Kim G. ; SKOU, Arne: Model-based framework for schedulability analysis using UPPAAL 4.1. In: *Model-based design for embedded systems* 1 (2009), Nr. 1, S. 93–119
- [DRTB14] DIE RITTAL TECHNIK-BIBLIOTHEK, Band 4.: *Die Welt der IT-Infrastrukturen*. Rittal, 2014
- [EMW09] EIBLMAIER, Matthias ; MAO, Rukun ; WANG, Xiaorui: Power management for main memory with access latency control. In: *Fourth International Workshop on Feedback Control Implementation and Design in Computing Systems and Networks* Citeseer, 2009, S. 22–29
- [FSM] FSM SIMULATOR: *Visually simulate your DFAs, NFAs and Epsilon-NFAs*. http://ivanzuzak.info/noam/webapps/fsm_simulator/. – letzter Zugriff 27.10.2020
- [FWB07] FAN, Xiaobo ; WEBER, Wolf-Dietrich ; BARROSO, Luiz A.: Power Provisioning for a Warehouse-sized Computer. In: *SIGARCH Comput. Archit. News* 35 (2007), Juni, Nr. 2, 13–23. <http://dx.doi.org/10.1145/1273440.1250665>. – DOI 10.1145/1273440.1250665. – ISSN 0163–5964
- [GCNS08] GUNARATNE, C. ; CHRISTENSEN, K. ; NORDMAN, B. ; SUEN, S.: Reducing the Energy Consumption of Ethernet with Adaptive Link Rate (ALR). In: *IEEE Transactions on Computers* 57 (2008), April, Nr. 4, S. 448–461. <http://dx.doi.org/10.1109/TC.2007.70836>. – DOI 10.1109/TC.2007.70836. – ISSN 0018–9340
- [GHZ13] GHRIBI, C. ; HADJI, M. ; ZEGHLACHE, D.: Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms. In: *2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, 2013, S. 671–678
- [GJN⁺10] GOIRI, I. ; JULIA, F. ; NOU, R. ; BERRAL, J. L. ; GUITART, J. ; TORRES, J.: Energy-Aware Scheduling in Virtualized Datacenters. In: *2010 IEEE*

- International Conference on Cluster Computing*, 2010. – ISSN 1552–5244, S. 58–67
- [Her11] HERBOLD, Astrid: *Wie viel Strom verbraucht das Netz?* Zeit. <https://www.zeit.de/digital/internet/2011-05/internet-facebook-green-it>.
Version: Mai 2011. – letzter Zugriff 19.12.2020
- [HGWQ11] HUANG, Q. ; GAO, F. ; WANG, R. ; QI, Z.: Power Consumption of Virtual Machine Live Migration in Clouds. In: *Third International Conference on Communications and Mobile Computing (CMC 2011)*, 2011, 122-125
- [HMGW07] HYSER, Chris ; MCKEE, Bret ; GARDNER, Rob ; WATSON, Brian J.: Autonomic virtual machine placement in the data center. In: *Hewlett Packard Laboratories, Tech. Rep. HPL-2007-189* 189 (2007)
- [Hoy11] HOYER, Marko: *Resource management in virtualized data centers regarding performance and energy aspects*, Carl von Ossietzky University of Oldenburg, Diss., 2011. <http://oops.uni-oldenburg.de/volltexte/2011/1200/>
- [HSL⁺04] HUANG, Hai ; SHIN, Kang G. ; LEFURGY, Charles ; RAJAMANI, Karthick ; KELLER, Tom ; HENSBERGEN, Eric V. ; RAWSON, Freeman: Cooperative software-hardware power management for main memory. In: *Workshop on Power-Aware Computer Systems*, 2004
- [Int04] INTEL: *Enhanced Intel SpeedStep Technology for the Intel Pentium M Processor*. <http://download.intel.com/design/network/papers/30117401.pdf>.
Version: 2004. – letzter Zugriff 18.12.2020
- [JFL20] JFLAP: *JFLAP 7.0 Tutorial*. <http://www.jflap.org/tutorial/>.
Version: 2020. – letzter Zugriff 27.10.2020
- [KGA⁺11] KRIOUKOV, Andrew ; GOEBEL, Christoph ; ALSPAUGH, Sara ; CHEN, Yanpei ; CULLER, David E. ; KATZ, Randy H.: Integrating Renewable Energy Using Data Analytics Systems: Challenges and Opportunities. In: *IEEE Data Eng. Bull.* 34 (2011), Nr. 1, 3-11. <http://dblp.uni-trier.de/db/journals/debu/debu34.html#KrioukovGACCK11>
- [Kre05] KREILING, Jörg: *Energieabsicherung im Rechenzentrum durch modulare USV Anlagen*. Rittal Whitepaper, 2005
- [KSK14] KUMAR, Pradeep ; SINGH, Dilbag ; KAUSHIK, Ankur: Power and Data Aware Best Fit Algorithm for Energy Saving in Cloud Computing. In: *International Journal of Computer Science and Information Technologies* 5 (2014), Nr. 5, S. 6712–6715
- [LCB⁺12] LIU, Zhenhua ; CHEN, Yuan ; BASH, Cullen ; WIERMAN, Adam ; GMACH, Daniel ; WANG, Zhikui ; MARWAH, Manish ; HYSER, Chris: Renewable and Cooling Aware Workload Management for Sustainable Data Centers. In: *SIGMETRICS Perform. Eval. Rev.* 40 (2012), Juni, Nr. 1, 175–186. <http://dx.doi.org/10.1145/2318857.2254779>. – DOI 10.1145/2318857.2254779. – ISSN 0163–5999

- [LLW⁺11] LIU, Zhenhua ; LIN, Minghong ; WIERMAN, Adam ; LOW, Steven H. ; ANDREW, Lachlan L.: Greening Geographical Load Balancing. In: *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*. New York, NY, USA : ACM, 2011 (SIGMETRICS '11). – ISBN 978-1-4503-0814-4, 233–244
- [LNY07] LEE, Jeabin ; NAM, Byeong-Gyu ; YOO, Hoi-Jun: Dynamic voltage and frequency scaling (DVFS) scheme for multi-domains power management. In: *IEEE Asian Solid-State Circuits Conference*, 2007, S. 360–363
- [LPY97] LARSEN, Kim G. ; PETTERSSON, Paul ; YI, Wang: Uppaal in a nutshell. In: *International Journal on Software Tools for Technology Transfer* 1 (1997), Dec, Nr. 1, 134–152. <http://dx.doi.org/10.1007/s100090050010>. – DOI 10.1007/s100090050010. – ISSN 1433-2779
- [LSH10] LE SUEUR, Etienne ; HEISER, Gernot: Dynamic Voltage and Frequency Scaling: The Laws of Diminishing Returns. In: *Proceedings of the 2010 International Conference on Power Aware Computing and Systems*. Berkeley, CA, USA : USENIX Association, 2010 (HotPower'10), 1–8
- [LWYH09] LASZEWSKI, G. von ; WANG, L. ; YOUNGE, A. J. ; HE, X.: Power-aware scheduling of virtual machines in DVFS-enabled clusters. In: *2009 IEEE International Conference on Cluster Computing and Workshops*, 2009. – ISSN 1552-5244, S. 1–10
- [MCRS05] MOORE, Justin ; CHASE, Jeff ; RANGANATHAN, Parthasarathy ; SHARMA, Ratnesh: Making Scheduling ‘Cool’: Temperature-aware Workload Placement in Data Centers. In: *Proceedings of the Annual Conference on USENIX Annual Technical Conference*. Berkeley, CA, USA : USENIX Association, 2005 (ATEC '05), 61–75
- [MDKS12] MISHRA, M. ; DAS, A. ; KULKARNI, P. ; SAHOO, A.: Dynamic resource management using virtual machine migrations. In: *IEEE Communications Magazine* 50 (2012), September, Nr. 9, S. 34–40. <http://dx.doi.org/10.1109/MCOM.2012.6295709>. – DOI 10.1109/MCOM.2012.6295709. – ISSN 0163-6804
- [MGD⁺12] MANN, Vijay ; GUPTA, Akanksha ; DUTTA, Partha ; VISHNOI, Anilkumar ; BHATTACHARYA, Parantapa ; PODDAR, Rishabh ; IYER, Aakash: Remedy: Network-Aware Steady State VM Management for Data Centers. In: BESTAK, Robert (Hrsg.) ; KENCL, Lukas (Hrsg.) ; LI, Li E. (Hrsg.) ; WIDMER, Joerg (Hrsg.) ; YIN, Hao (Hrsg.): *NETWORKING 2012*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012. – ISBN 978-3-642-30045-5, S. 190–204
- [MGW09] MEISNER, David ; GOLD, Brian T. ; WENISCH, Thomas F.: PowerNap: Eliminating Server Idle Power. In: *SIGARCH Comput. Archit. News* 37 (2009), März, Nr. 1, 205–216. <http://dx.doi.org/10.1145/2528521.1508269>. – DOI 10.1145/2528521.1508269. – ISSN 0163-5964

- [MLL12] MA, Fei ; LIU, Feng ; LIU, Zhen: Multi-objective optimization for initial virtual machine placement in cloud data center. In: *Journal of Information & Computational Science* 9 (2012), 12, Nr. 16, S. 5029–5038
- [MLP⁺13] MEI, Y. ; LIU, L. ; PU, X. ; SIVATHANU, S. ; DONG, X.: Performance Analysis of Network I/O Workloads in Virtualized Data Centers. In: *IEEE Transactions on Services Computing* 6 (2013), Nr. 1, S. 48–63. <http://dx.doi.org/10.1109/TSC.2011.36>. – DOI 10.1109/TSC.2011.36. – ISSN 1939–1374
- [MRBA16] MEDEL, Victor ; RANA, Omer ; BAÑARES, José á. ; ARRONATEGUI, Unai: Modelling Performance & Resource Management in Kubernetes. In: *Proceedings of the 9th International Conference on Utility and Cloud Computing*. New York, NY, USA : ACM, 2016 (UCC '16). – ISBN 978–1–4503–4616–0, 257–262
- [MVG16] MEMARI, Ammar ; VORNBERGER, Jan ; GÓMEZ, Jorge M. ; NEBEL, Wolfgang: A data center simulation framework based on an ontological foundation. In: *Advances and New Trends in Environmental and Energy Informatics*. Springer, 2016, S. 39–57
- [MW11] MEISNER, David ; WENISCH, Thomas F.: Does low-power design imply energy efficiency for data centers? In: CHANG, Naehyuck (Hrsg.) ; NAKAMURA, Hiroshi (Hrsg.) ; INOUE, Koji (Hrsg.) ; OSADA, Kenichi (Hrsg.) ; PONCINO, Massimo (Hrsg.): *ISLPED*, IEEE/ACM, 2011. – ISBN 978–1–61284–660–6, 109–114
- [NHSS09] NEBEL, Wolfgang ; HOYER, Marko ; SCHRÖDER, Kiril ; SCHLITT, Daniel: Untersuchung des Potentials von rechenzentrenübergreifendem Lastmanagement zur Reduzierung des Energieverbrauchs in der IKT. In: *Studie für das Bundesministerium für Wirtschaft und Technologie, OFFIS* (2009)
- [NLM18] NADJAH, Chayan ; LOUAHLIA, Hasna ; MASSON, Stephane: A review of thermal management and innovative cooling strategies for data center. In: *Sustainable Computing: Informatics and Systems* 19 (2018), 05. <http://dx.doi.org/10.1016/j.suscom.2018.05.002>. – DOI 10.1016/j.suscom.2018.05.002
- [NSL⁺06] NEIGER, Gil ; SANTONI, Amy ; LEUNG, Felix ; RODGERS, Dion ; UHLIG, Richard: Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization. In: *Intel Technology Journal* 10 (2006), 08, Nr. 3. <http://dx.doi.org/10.1535/itj.1003>. – DOI 10.1535/itj.1003
- [OP16] OEVEL, Gudrun ; POROMBKA, Sebastian: Energy Efficient Ethernet in der Praxis. In: *9. DFN-Forum Kommunikations-technologien* Gesellschaft für Informatik eV, 2016, S. 21
- [ope19] OPENSTACK: *Vision for OpenStack Clouds*. https://governance.openstack.org/tc/reference/technical-vision.html?_ga=

Literaturverzeichnis

2.174942242.1680374033.1606918350-838222666.1606918350.
Version: 2019. – letzter Zugriff 02.12.2020

- [PLM⁺13] PU, X. ; LIU, L. ; MEI, Y. ; SIVATHANU, S. ; KOH, Y. ; PU, C. ; CAO, Y.: Who Is Your Neighbor: Net I/O Performance Interference in Virtualized Clouds. In: *IEEE Transactions on Services Computing* 6 (2013), July, Nr. 3, S. 314–329. <http://dx.doi.org/10.1109/TSC.2012.2>. – DOI 10.1109/TSC.2012.2. – ISSN 1939–1374
- [PP09] PAKBAZNIA, Ehsan ; PEDRAM, Massoud: Minimizing Data Center Cooling and Server Power Costs. In: *Proceedings of the 2009 ACM/IEEE International Symposium on Low Power Electronics and Design*. New York, NY, USA : ACM, 2009 (ISLPED '09). – ISBN 978–1–60558–684–7, 145–150
- [Pre12] PRESSE- UND INFORMATIONSAMT DER BUNDESREGIERUNG: *AUF DEM WEG INS ZEITALTER DER ERNEUERBAREN ENERGIEN*. <https://www.bundesregierung.de/ContentArchiv/DE/Archiv17/Artikel/2012/06/2012-06-04-artikel-energiewende-langfassung.html>. Version: Juni 2012. – letzter Zugriff 18.4.2018
- [Pre19a] PRESSE- UND INFORMATIONSAMT DER BUNDESREGIERUNG: *Abschied von der Kohleverstromung*. <https://www.bundesregierung.de/breg-de/aktuelles/kohleausstiegsgesetz-1716678>. Version: Oktober 2019. – letzter Zugriff 14.02.2020
- [Pre19b] PRESSE- UND INFORMATIONSAMT DER BUNDESREGIERUNG: *Der Einstieg in den Kohleausstieg*. <https://www.bundesregierung.de/breg-de/aktuelles/der-einstieg-in-den-kohleausstieg-1574264>. Version: Januar 2019. – letzter Zugriff 14.02.2020
- [PSBB02] PATEL, C. D. ; SHARMA, R. ; BASH, C. E. ; BEITELMAL, A.: Thermal considerations in cooling large scale high compute density data centers. In: *ITherm 2002. Eighth Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems (Cat. No.02CH37258)*, 2002. – ISSN 1089–9870, S. 767–776
- [PY10] PIAO, Jing T. ; YAN, Jun: A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing. In: *Proceedings of the 2010 Ninth International Conference on Grid and Cloud Computing*. Washington, DC, USA : IEEE Computer Society, 2010 (GCC '10). – ISBN 978–0–7695–4313–0, 87–92
- [QWB⁺09] QURESHI, Asfandyar ; WEBER, Rick ; BALAKRISHNAN, Hari ; GUTTAG, John ; MAGGS, Bruce: Cutting the Electric Bill for Internet-scale Systems. In: *SIGCOMM Comput. Commun. Rev.* 39 (2009), August, Nr. 4, 123–134. <http://dx.doi.org/10.1145/1594977.1592584>. – DOI 10.1145/1594977.1592584. – ISSN 0146–4833
- [RAL15] RAL UND UMWELTBUNDESAMT: *Energieeffizienter Rechenzentrumsbetrieb*. <https://produktinfo.blauer-engel.de/uploads/criteriafile/>

- de/DE-UZ%20161-201502-de%20Kriterien.pdf. Version: Februar 2015. – letzter Zugriff 04.12.2020
- [RDSM15] RYBINA, K. ; DARGIE, W. ; SCHÖNE, R. ; MALAKUTI, S.: Mutual Influence of Application- and Platform-Level Adaptations on Energy-Efficient Computing. In: *23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, 2015, S. 446–450
- [RLL10] RAO, Lei ; LIU, Xue ; LIU, Wenyu: Minimizing electricity cost: Optimization of distributed internet data centers in a multi-electricity-market environment. In: *2010 Proceedings IEEE INFOCOM IEEE*, 2010, S. 1–9
- [Sch16] SCHLITT, Daniel: *Entwicklung einer auslastungsabhängigen Energieeffizienzmetrik für Rechenzentren*, Carl von Ossietzky Universität Oldenburg, Diss., 2016
- [Sci] SCIENCE OF COMPUTING SUITE (SOCS): *Automaton Simulator 1.2*. <http://www.cburch.com/proj/autosim/index.html>. – letzter Zugriff 27.10.2020
- [SD13] STRUNK, Anja ; DARGIE, Walteneus: Does Live Migration of Virtual Machines Cost Energy? In: *Proceedings of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications*. Washington, DC, USA : IEEE Computer Society, 2013 (AINA '13). – ISBN 978-0-7695-4953-8, 514–521
- [SGIS10] SHARMA, Navin ; GUMMESON, Jeremy ; IRWIN, David E. ; SHENOY, Prashant J.: Cloudy Computing: Leveraging Weather Forecasts in Energy Harvesting Sensor Systems. In: *SECON*, IEEE, 2010, 136-144
- [SNP+09] STOBBE, Lutz ; NISSEN, Nils F. ; PROSKE, Marina ; MIDDENDORF, Andreas ; SCHLOMANN, Barbara ; FRIEDEWALD, Michael ; GEORGIEFF, Peter ; LEIMBACH, Timo: Abschätzung des Energiebedarfs der weiteren Entwicklung der Informationsgesellschaft. In: *Abschlussbericht an das Bundesministerium für Wirtschaft und Technologie. Berlin, Karlsruhe: Fraunhofer IZM* (2009)
- [Spi07] SPIEGEL ONLINE: *Energieschleuder Ethernet: Stromverbrauch von Netzwerken*. <http://www.spiegel.de/netzwelt/tech/stromverbrauch-von-netzwerken-energieschleuder-ethernet-a-464470.html>. Version: 2007. – letzter Zugriff 01.12.2020
- [SPV09] STEVEN PELLEY, Thomas F. W. David Meisner M. David Meisner ; VANGILDER, James W.: Understanding and Abstracting Total Data Center Power. In: *Proc. of the 2009 Workshop on Energy Efficient Design (WEED)* 11 (2009), S. 1–6
- [SPZ+15] STOBBE, Lutz ; PROSKE, Marina ; ZEDEL, Hannes ; HINTEMANN, Ralph ; CLAUSEN, Jens ; BEUCKER, Severin: Entwicklung des IKT-bedingten Strombedarfs in Deutschland. In: *development of ICT related electricity demand in Germany. Study for the German Federal Ministry for Economy and Energy (BMWi), Berlin* (2015), 11

Literaturverzeichnis

- [Sta] STANDARD PERFORMANCE EVALUATION CORPORATION (SPEC): *Published SPEC Benchmark Results*. <http://www.spec.org/results.html>. – letzter Zugriff 10.12.2020
- [Sta09] STANDARD PERFORMANCE EVALUATION CORPORATION (SPEC): *Benchmark Results Summary Sun Microsystems, Inc. Sun Netra X4250*. http://spec.org/power_ssj2008/results/res2009q1/power_ssj2008-20090310-00128.html. Version: 2009. – letzter Zugriff 11.03.2020
- [Sta12] STANDARD PERFORMANCE EVALUATION CORPORATION (SPEC): *SSJ Workload SPEC power_ssj2008*. https://www.spec.org/power/docs/SPECpower_ssj2008-Design_ssj.pdf. Version: 2012. – letzter Zugriff 22.12.2020
- [Sta15] STANDARD PERFORMANCE EVALUATION CORPORATION (SPEC): *Fujitsu FUJITSU Server PRIMERGY TX2560 M1 Benchmark Results Summary*. http://spec.org/power_ssj2008/results/res2015q2/power_ssj2008-20150317-00689.html. Version: 2015. – letzter Zugriff 27.07.2018
- [Sta16] STANDARD PERFORMANCE EVALUATION CORPORATION (SPEC): *Benchmark Results Summary Dell Inc. PowerEdge R830*. https://www.spec.org/power_ssj2008/results/res2016q3/power_ssj2008-20160705-00737.html. Version: 2016. – letzter Zugriff 22.07.2019
- [SVT11] SEMENOV, Evgeny ; VERSICK, Daniel ; TAVANGARIAN, Djamshid: Ansätze zur Reduktion der Live-Migrationsdauer virtueller Maschinen. In: HEISS, Hans-Ulrich (Hrsg.) ; PEPPER, Peter (Hrsg.) ; SCHLINGLOFF, Holger (Hrsg.) ; SCHNEIDER, Jörg (Hrsg.): *INFORMATIK 2011 - Informatik schafft Communities*. Bonn : Gesellschaft für Informatik e.V., 2011, S. 78–78
- [TGSC06] TANG, Q. ; GUPTA, S. K. S. ; STANZIONE, D. ; CAYTON, P.: Thermal-Aware Task Scheduling to Minimize Energy Usage of Blade Server Based Datacenters. In: *2006 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, 2006, S. 195–202
- [TGV07] TANG, Q. ; GUPTA, S. K. S. ; VARSAMOPOULOS, G.: Thermal-aware task scheduling for data centers through minimizing heat recirculation. In: *2007 IEEE International Conference on Cluster Computing*, 2007. – ISSN 1552–5244, S. 129–138
- [TGV08] TANG, Qinghui ; GUPTA, Sandeep Kumar S. ; VARSAMOPOULOS, Georgios: Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. In: *IEEE Transactions on Parallel and Distributed Systems* 19 (2008), Nr. 11, S. 1458–1472
- [TKBL13] TIGHE, M. ; KELLER, G. ; BAUER, M. ; LUTFIYYA, H.: A distributed approach to dynamic VM management. In: *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*, 2013. – ISSN 2165–9605, S. 166–170

- [VAN08] VERMA, Akshat ; AHUJA, Puneet ; NEOGI, Anindya: pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems. In: ISSARNY, Valerie (Hrsg.) ; SCHANTZ, Richard (Hrsg.): *Middleware 2008: ACM/IFIP/USENIX 9th International Middleware Conference Leuven, Belgium, December 1-5, 2008 Proceedings*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2008. – ISBN 978-3-540-89856-6, 243–264
- [VBVB09] VOORSLUYS, William ; BROBERG, James ; VENUGOPAL, Srikumar ; BUYYA, Rajkumar: Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. In: JAATUN, Martin G. (Hrsg.) ; ZHAO, Gansen (Hrsg.) ; RONG, Chunming (Hrsg.): *Cloud Computing*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009. – ISBN 978-3-642-10665-1, S. 254–265
- [WEL18] WELT: *Stromnetz: Tennet meldet Rekordkosten für Noteingriffe.* https://www.welt.de/newsticker/dpa_nt/infoline_nt/wirtschaft_nt/article172058288/Stromnetz-Tennet-meldet-Rekordkosten-fuer-Noteingriffe.html. Version: 2018. – letzter Zugriff 19.12.2020
- [Wis19] WISSENSCHAFTLICHE DIENSTE DEUTSCHER BUNDESTAG: *Energieverbrauch von Rechenzentren.* <https://www.bundestag.de/resource/blob/651446/d226ff9ff67a3c29d893859121cfc5fe/WD-8-041-19-pdf-data.pdf>. Version: April 2019. – letzter Zugriff 09.12.2020
- [Wol18] WOLFF, Eberhard: *Microservices: Grundlagen flexibler Softwarearchitekturen.* dpunkt.verlag, 2018
- [WTLS⁺09] WOOD, Timothy ; TARASUK-LEVIN, Gabriel ; SHENOY, Prashant ; DESNOYERS, Peter ; CECCHET, Emmanuel ; CORNER, Mark D.: Memory buddies: exploiting page sharing for smart colocation in virtualized data centers. In: *ACM SIGOPS Operating Systems Review* 43 (2009), Nr. 3, S. 27–36
- [XFL13] XU, Hong ; FENG, Chen ; LI, Baochun: Temperature Aware Workload Management in Geo-distributed Datacenters. In: *Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems*. New York, NY, USA : ACM, 2013 (SIGMETRICS '13). – ISBN 978-1-4503-1900-3, 373–374
- [ZWW11] ZHANG, Yanwei ; WANG, Yefu ; WANG, Xiaorui: Greenware: Greening cloud-scale data centers to maximize the use of renewable energy. In: *ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing*, Springer Berlin Heidelberg, 2011, S. 143–164
- [ZWZ17] ZHANG, Yin ; WEI, Zhiyuan ; ZHANG, Mingshan: Free cooling technologies for data centers: energy saving mechanism and applications. In: *Energy Procedia* 143 (2017), 410 - 415. <http://dx.doi.org/https://doi.org/10.1016/j.egypro.2017.12.703>. – DOI <https://doi.org/10.1016/j.egypro.2017.12.703>. – ISSN 1876-6102

Erklärung

Erklärung zu meiner Dissertation mit dem Titel:

Ressourcen-Provisionierungsverfahren zur Erweiterung energetischer Freiheitsgrade von Rechenzentren unter Veränderung der IT-Last

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel benutzt habe. Alle wörtlich oder inhaltlich übernommenen Inhalte sind als solche gekennzeichnet.

Oldenburg, 31.12.2020