Maschinelles Lernen für Ressourcenplanung in Verteilten Systemen[1]

Michael Borkowski[2]

Abstract: Verteilte Rechensysteme sind aus der heutigen digitalen Welt nicht mehr wegzudenken: Suchmaschinen wie Google, Cloud-Speichersysteme wie Dropbox, Streaming-Dienste wie Netflix oder wissenschaftliche Großrechner führen komplexe Aufgaben auf verteilter IT-Infrastruktur aus. Dabei müssen entsprechende Systeme laufend Ressourcenoptimierung betreiben. Beispielsweise können durch Aktivierung von Ressourcen kurz vor Lastspitzen und anschließender Passivierung enorme Kostenersparnisse erzielt werden. Statt konventioneller Wenn-Dann-Beziehungen oder starrer Regelkreise beschreibe ich in meiner Dissertation adaptive und Vorhersage-basierte Techniken, wie sie in einer dynamischen Umgebung wie dem heutigen Internet unabdingbar sind. Hierfür verwende ich Modelle für maschinelles Lernen, insbesondere künstliche neuronale Netze und Kalman-Filter. Meine Ergebnisse zeigen, dass der Einsatz solcher Methoden Kosten und Ressourcenverbrauch senkt sowie die Verfügbarkeit und Verlässlichkeit der Systeme erhöht.

1 Einführung

Das Versagen eines verteilten Systems kann 66.240 US-Dollar kosten – pro Minute. So erlitt Amazon Web Services (AWS) im April 2011 einen Teilausfall, der binnen 12 Stunden einen Gesamtschaden von 48 Millionen US-Dollar verursachte [Ah17]. Zahlen wie diese verdeutlichen den heutigen wirtschaftlichen Stellenwert von verteilten Systemen. Sie machen deutlich, dass Ansätze zur Verbesserung der Systemstabilität und -performanz in der heutigen digitalen Welt unentbehrlich sind. Um ihre Systeme kostenoptimiert zu betreiben, können Betreiber auf Methoden des maschinellen Lernens zurückgreifen. Beispiele solcher Methoden entwickle und beschreibe ich in meiner Dissertation [Bo20].

Verteilte Systeme spielen eine entscheidende Rolle in vielen Aspekten der heutigen digitalen Infrastruktur und ermöglichen Cloud-Speicherlösungen [Gr15], Smart Cities [PLM17] oder das Internet of Things (IoT) [Bo16]. Die Forschung innerhalb der verteilten Systeme umfasst Bereiche wie Cloud Computing [Ar10], Data Stream Processing (DSP) [Ca18b], Business Process Management (BPM) [Sc15] oder dezentrale Consensus-Technologien wie Blockchains [Zh18]. Zu den gängigen Zielen gehören Elastizität, Flexibilität und Skalierbarkeit, während gleichzeitig Kosten [Sc15] oder Qualitätseinbußen [Ca18a] eingeschränkt werden sollen.

Moderne verteilte Systeme weisen aufgrund ihrer verteilten Architektur einen hohen Grad an Komplexität auf. Die für den Betrieb und die Wartung solcher Systeme erforderlichen Komponenten bilden eine unübersichtliche Landschaft, welche oft sehr heterogen ist, da

[1] Englischer Titel der Dissertation: „Predictive Approaches for Resource Provisioning in Distributed Systems“

[2] Deutsches Zentrum für Luft- und Raumfahrt (DLR), Institut für Flugführung, michael.borkowski@dlr.de

einzelne Komponenten oft von verschiedenen Interessengruppen bereitgestellt und auf uneinheitlicher Hardware mit unterschiedlichen Technologien betrieben werden.

In vielen Situationen muss ein verteiltes System Ressourcenplanung durchführen: Ressourcen wie etwa Rechenleistung müssen in ausreichender – aber nicht überschüssiger – Menge zur Verfügung gestellt werden (Skalierung), Aufgaben müssen auf die verfügbaren Ressourcen aufgeteilt werden (Platzierung bzw. Allocation) und die zeitliche Abfolge der Ausführung einzelner Aufgaben muss bestimmt werden (Zeitplanung bzw. Scheduling).

In modernen verteilten System müssen vielfältige, sich oft kurzfristig ändernde Einsatzbereiche berücksichtigt werden. Es ist daher für einen kostenoptimierten Betrieb unbedingt notwendig, dass die Ressourcenplanung eines verteilten Systems nicht durch starre Regeln (etwa Last-Schwellwerte, die für das Hinzufügen bzw. Entfernen von Rechenressourcen verwendet werden) oder konventionelle Wenn-Dann-Regelwerke definiert werden, wie dies aktuell meist der Fall ist. Vielmehr müssen proaktiv, Vorhersage-basiert und automatisiert Entscheidungen zur Ressourcenplanung getroffen werden. Beispielsweise kann ein System bereits vor einem vorhergesagten Anstieg der Last Rechenressourcen hinzufügen, um Einbußen in der Verfügbarkeit zu vermeiden.

In meiner Dissertation [Bo20] präsentiere ich Vorhersage-basierte Ansätze zur Ressourcenplanung in verteilten Systemen, welche zur Senkung von Kosten und gleichzeitig zur Erhöhung der Systemstabilität und Verfügbarkeit führen. Die Ansätze verwenden Techniken aus dem Bereich des maschinellen Lernens (ML), insbesondere künstliche neuronale Netze (ANNs). Ich zeige mittels Vergleichen zum Stand der Technik, wie diese Ansätze zu einem optimierten Betrieb verschiedener Arten von verteilten Systemen beitragen.

2 Ressourcenplanung in Verteilten Systemen

Der Prozess der Ressourcenplanung, etwa die Zuweisung von Ressourcen zu bestimmten Aufgaben, ist eines der wichtigsten Gebiete in verteilten Systemen [Ha17]. Oft wird das Placement-Problem, also das Platzieren von Anwendungen auf Virtuellen Maschinen (VMs) oder VMs auf Physischen Maschinen (PMs), betrachtet [Ca16]. Die Ressource, auf der eine Aufgabe platziert wird, ist für eine bestimmte Zeit und in einem bestimmten Ausmaß belegt, weswegen die genaue Zuordnung zwischen Ressourcen und Aufgaben einen erheblichen Einfluss auf die gesamte Ressourcenauslastung hat. Dies wirkt sich auf die Betriebskosten und die Leistung des gesamten Systems aus [CLN12].

Zusätzlich zum Finden einer geeigneten Platzierung müssen moderne verteilte Systeme auch Elastizität sicherstellen [Du11, LBMAL14], was durch eine dynamische und automatische Anpassung an Änderungen der Arbeitslast sowie Aktivierung und Passivierung von Ressourcen erreicht wird. Diese Fähigkeit wird als Skalierbarkeit bezeichnet [HKR13]. Die Entscheidung, wann und wie skaliert werden soll ist ein nicht-triviales Problem, und oft gibt es mehrere widersprüchliche Optimierungszielmetriken [Co13]. Beispielsweise kann ein Cloud-Computing-Anbieter daran interessiert sein, die Betriebskosten seiner Infrastruktur zu reduzieren, muss aber gleichzeitig eine bestimmte Dienstgüte (engl. Quality

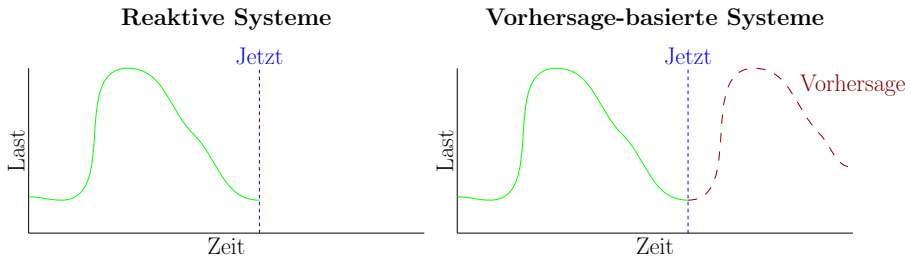

Abb. 1: Reaktive und Vorhersage-basierte Verfahren und ihr Umgang mit Werten einer Systemmetrik

of Service – QoS) einhalten, um bestimmte Leistungsvereinbarungen (engl. Service Level Agreements – SLAs) zu erfüllen und somit Strafzahlungen zu vermeiden [Ca18a].

Gängige Lösungen zur Ressourcenplanung können in reaktive sowie proaktive Ansätze unterteilt werden [LBMAL14], wie in Abbildung 1 illustriert ist. Reaktive Ansätze zeichnen sich dadurch aus, dass eine Metrik beobachtet wird, und aufgrund des aktuellen Werts Entscheidungen getroffen werden. Ein klassisches Beispiel hierfür ist ein Schwellwert-gesteuertes Skalierungsverhalten, bei dem zusätzliche Rechenleistung aktiviert wird, wenn die Systemlast einen vordefinierten oberen Schwellwert überschreitet, und wieder passiviert wird, wenn die Systemlast den unteren Schwellwert unterschreitet [LBMAL14]. Im Gegensatz dazu zeichnen sich proaktive, insbesondere Vorhersage-basierte Ansätze dadurch aus, dass nicht nur der aktuelle Wert einer Metrik, sondern auch eine Vorhersage über ihren zukünftigen Wert getroffen wird. Anhand dieser Vorhersage kann proaktiv auf mögliche zukünftige Veränderungen reagiert werden, etwa in Form von zusätzlichen Rechenressourcen bereits vor einer Lastspitze [LBMAL14].

In meiner Dissertation [Bo20] untersuche ich Vorhersage-basierte Ansätze zur Ressourcenplanung in verteilten Systemen. Insbesondere schaffe ich Grundlagen zum Treffen von Vorhersagen in diesem Bereich, wie etwa die Auswahl an Methoden und Werkzeugen, die in verschiedenen Situationen zu brauchbaren Vorhersagen führen. Ich stelle dar, wie diese Methoden auf verschiedene Probleme und Anwendungsfälle in modernen verteilten Systemen angewendet werden können, und bewerte ihre Leistung – also die Auswirkung auf die Betriebskosten und die Leistungsfähigkeit der Systeme – quantitativ.

In meiner Arbeit betrachte ich dabei verschiedene Arten von verteilten Systemen, insbesondere Cloud Computing [BSH16], BPM-Systeme [Bo19a], DSP [BHS19] und Blockchain-Technologien [Bo19b]. Ich untersuche verschiedene Aspekte von Vorhersage-basierten Ansätzen, darunter die Vorhersage und Filterung von Metrikwerten sowie die Abschätzung der Eintrittswahrscheinlichkeit von Fehlerzuständen in Geschäftsprozessen.

3 Maschinelles Lernen, Neuronale Netze und Kalman-Filter

In den vorgeschlagenen Ansätzen zur Vorhersage-basierten Ressourcenplanung verwende ich Methoden des ML, also des maschinellen Erlernens von Mustern und Regelmäßigkeiten aus Trainingsdaten, ohne dass dabei durch menschliches Zutun diese Muster explizit als

Regeln definiert werden müssen. Dies setzt sich deutlich von regelbasierten Methodiken ab, da bei Anwendung von ML im Allgemeinen das notwendige Fachwissen über die zugrundeliegenden Charakteristika entfällt [Hu14]. Stattdessen erlernt das System diese Charakteristika anhand von beispielhaften Daten. Es existiert eine Vielfalt an Möglichkeiten, ein solches System zu trainieren, und es kann zwischen überwachtem, teilüberwachtem und unüberwachtem Lernen [SA13] sowie zwischen Online- und Offline-Lernen unterschieden werden [BDKM97].

Insbesondere ANN sind ein weitläufig verwendetes Werkzeug bei der Vorhersage von Werten anhand von Trainingsdaten [Ha98]. ANN bestehen zumeist aus mehreren Schichten sogenannter künstlicher Neuronen, die miteinander verbunden sind. Beispielhaft ist dies in Abbildung 2 dargestellt. Ein ANN nimmt einen Vektor an Werten als Eingabe an der Eingabeschicht an, und verwendet oft mehrere verborgene Schichten und schlussendlich die Ausgabeschicht, um je nach Anwendungsfall einen oder mehrere Ausgabewerte zu erzeugen. Die Ausgabe eines ANN stellt dabei die Vorhersage auf Basis des zur Verfügung gestellten Eingabevektors dar.

Ein weiteres wichtiges Werkzeug im Kontext von ML sind Kalman-Filter (KF) [KB61]. Im Wesentlichen dienen KF dazu, Daten aus mehreren, verschiedenartigen Sensorquellen mit verschiedenen und zeitlich variablen Unschärfen zu einer gemeinsamen Messgröße zusammenzufassen, und stellen dabei auch die Unschärfe dieser Messgröße zur Verfügung. Zur Anwendung von KF wird ein System mittels eines Zustandsvektors definiert. Jeder Einzelwert des Zustandsvektors stellt dabei eine Systemgröße dar, die von Interesse ist, wie etwa CPU- und RAM-Auslastung bei einem verteilten System [Gu12, CFF14]. Außerdem wird ein Eingabevektor definiert, welcher die Umgebung des beobachteten Systems bzw. dessen Eingabe (darauf wirkende Einflüsse) definiert. Jeder Einzelwert des Eingabevektors stellt dabei einen äußeren Einfluss auf das System dar, wie etwa die Menge der zu verarbeitenden Daten. Darüber hinaus wird eine Zustandsübergangsmatrix definiert, anhand der aus dem aktuellen Zustands- und Eingabevektor ein neuer Zustandsvektor erzeugt wird. Diese Matrix definiert die Systemdynamik. Während reguläre KF rein lineare Übergangsmatrizen verwenden, verallgemeinert der erweiterte KF (EKF) den Ansatz für nichtlineare Modelle [Ja07]. Anstelle von Matrizen verwenden EKF Funktionen als

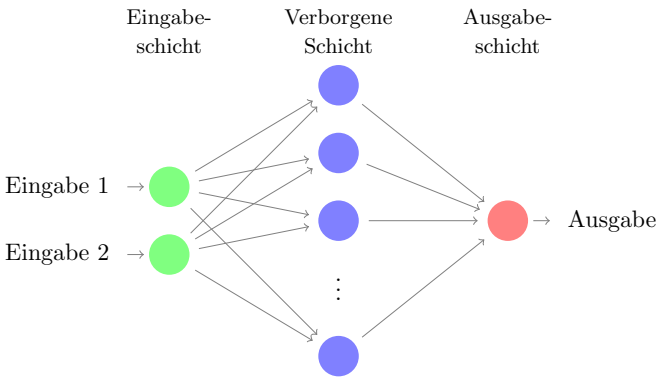

Abb. 2: Skizze eines mehrschichtigen ANN, vgl. [Bo20]

Übergangsmodelle, um so nichtlineare Systemdynamiken abzubilden. Eine Besonderheit von KF ist es, dass der tatsächliche Systemzustand nicht direkt bekannt sein muss, sondern nur indirekt über Beobachtungen abgeleitet werden kann. Dabei können sowohl die Beobachtungen als auch der Systemzustand Rauschen unterliegen, welches vom KF modelliert und als Ergebnis ausgegeben wird.

KF arbeiten somit in Zyklen, bei denen aus einer bekannten Eingabe und einer bekannten Beobachtung der (grundsätzlich unbekannte) innere Systemzustand ableitbar gemacht wird. Abbildung 3 zeigt exemplarisch einen Ausschnitt aus dem Zyklusdurchlauf im Systemmodell von KF, bei denen die Eingabe, der Zustand sowie die Übergangsfunktion f und Messfunktion h zu sehen sind. Eine detailliertere Beschreibung samt Formeln des KF-Modells findet sich in meiner Dissertation [Bo20].

4 Ansätze und Ergebnisse

Wie in Aschnitt 1 beschrieben, existiert eine große Vielfalt an Arten von verteilten Systemen in verschiedenen Bereichen, beispielsweise Cloud Computing [Ar10], BPM [Sc15] oder DSP [Ca18b]. Jeder dieser Bereiche bringt ein eigenes Anforderungsprofil an Vorhersage-basierten Ansätze mit sich, außerdem finden sich in jedem der Bereiche je nach Situation verschiedenartige Anwendungsfälle.

Verteilte Systeme müssen daher differenziert betrachtet werden. In meiner Dissertation beleuchte ich zunächst, wie diversen Anforderungen an verteilte Systeme mit verschiedenen Methoden entgegengetreten werden kann. So gibt es etwa Anwendungsfälle, in denen Vorhersagen zeitverzögert getroffen werden kann, und Offline-Lernen das Mittel der Wahl darstellt [BDKM97]. In anderen Situationen ist die Vorhersage hingegen bereits während der Ausführung des Systems notwendig – in diesen Fällen müssen Online-Learning-Techniken angewendet werden [La96]. Eine andere bedeutende Hürde beim maschinellen Lernen

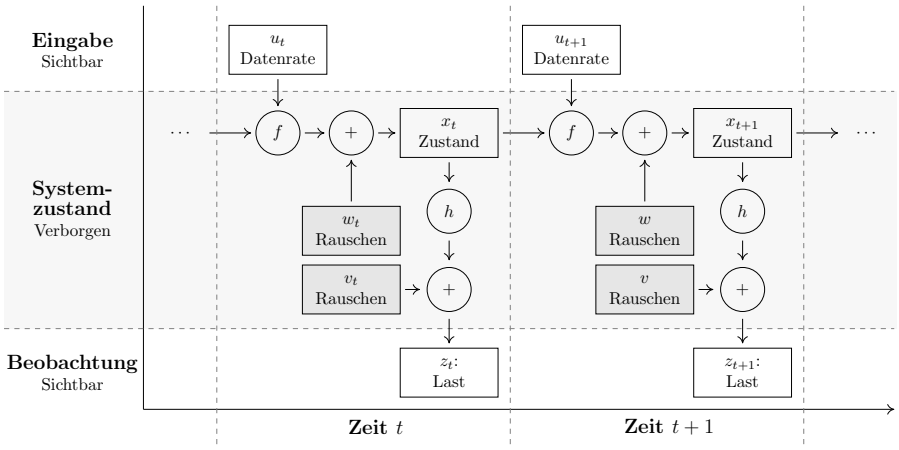

Abb. 3: Übersicht des Systemmodells in KF, vgl. [Bo20]

besteht in der Verfügbarkeit von Trainingsdaten. Wie oben beschrieben wird zwischen überwachtem, teilüberwachtem und unüberwachtem Lernen unterschieden [SA13], und die anzuwendende Methode hängt unter anderem von der Art der zugrundeliegenden Daten ab. Wenn diese Daten als sogenannte beschriftete Daten (labelled data) vorliegen, also als Eingabedaten gemeinsam mit den Soll-Ausgabedaten, werden überwachte Trainingstechniken (supervised learning) eingesetzt. Wenn hingegen keine oder nur begrenzte Soll-Ausgabedaten vorliegen, werden entweder auf teilüberwachte (semi-supervised) oder unüberwachte (unsupervised) Techniken angewendet.

In meiner Dissertation beschreibe ich daher einige Anwendungsfälle für maschinelles Lernen in verteilten Systemen, welche im Folgenden kurz umrissen werden, und stelle Lösungsmöglichkeiten vor. Ein Überblick über die einzelnen Beiträge findet sich in Tabelle 1. Ich variiere dabei sowohl die Domäne (Cloud Computing, BPMS, DSP) als auch die Zielaufgabe (Vorhersage des Ressourcenverbrauchs, Vorhersage von Fehlerzuständen, Vorhersage der Systemlast).

Tab. 1: Übersicht über die Beiträge der Dissertation zu Anwendungsfällen des maschinellen Lernens im Bereich der verteilten Systeme

Domäne	Methode	Ergebnis	Referenz
Cloud Computing	Regression mittels ANNs	Verbesserung d. Abweichung: 23 %	[BSH16]
EBS/BPMS	Klassifikation mittels ANNs	Präzision d. Vorhersage: 87 %	[Bo19a]
DSP	Skalierung mittels EKF	Verminderung d. Skalierungen: 88 %	[BHS19]

4.1 Vorhersage von Ressourcenverbrauch im Cloud Computing

Cloud Computing bezeichnet die Bereitstellung von IT-Ressourcen auf Bedarfsbasis in Form von Cloud-Diensten [Ar10], also als Dienste, welche bei Bedarf abgerufen und nach Verbrauch verrechnet werden. Der Anbieter eines Cloud-Dienstes muss eine Infrastruktur betreiben, die für Konsumenten zur Verfügung steht. Entsprechend liegt ein Hauptaugenmerk von Cloud-Anbietern darauf, diese Infrastruktur kostengünstig zu betreiben. Im Vordergrund stehen hierbei die Platzierung von Aufgaben (also Anforderungen von Rechenressourcen) auf Infrastrukturressourcen (etwa VMs) sowie die Skalierung, also die Aktivierung und Passivierung von Infrastruktur auf Basis des Bedarfs. Für diese beiden Vorgänge ist es essenziell, den Ressourcenbedarf der eintreffenden Aufgaben abzuschätzen, um diese Aufgaben optimal auf die bestehende bzw. neu zu aktivierende Infrastruktur zu verteilen.

Zu diesem Zweck präsentiere ich in Kapitel 3 meiner Dissertation sowie in der dazugehörigen Publikation [BSH16], wie ANNs zur Regression eingesetzt werden können. Vorhergesagt wird hierbei der Verbrauch von Systemressourcen für jede eintreffende Aufgabe auf Basis vergangener Aufgabenausführungen. Zur Evaluierung verwende ich einen Datensatz, welcher von Travis CI, einem Cloud-basierten Anbieter von Continuous Integration, gesammelt wurde. Vorhergesagt wird die benötigte Zeit sowie CPU-Auslastung.

Mittels mehrschichtigen ANNs erreiche ich im Median eine Verkleinerung der Vorhersageabweichung von 20 % (Zeit) bzw. 23 % (CPU-Auslastung).

4.2 Abschätzung der Fehlereintrittswahrscheinlichkeiten in Geschäftsprozessen

Geschäftsprozesse stellen in unserer vernetzten Welt ein wichtiges Werkzeug für große Unternehmen dar, bei dem Abläufe definiert und oft auch automatisiert werden. Die Abläufe können bei komplexen Prozessen vielfache Verzweigungen und Parallelitäten aufweisen und verschiedene Geschäftspartner umfassen [Fd12].

In Kapitel 4 meiner Dissertation sowie der dazugehörigen Publikation [Bo19a] stelle ich vor, wie aus Ereignissen, die während eines Geschäftsprozesses auftreten, die Eintrittswahrscheinlichkeit von Fehlern für verschiedene Prozessschritte abgeschätzt, Fehler also vorhergesagt werden können. Eine (hinreichend wahrscheinliche) Vorhersage von Fehlerzuständen kann bei Geschäftsprozessen dazu dienen, dass Geschäftspartner schon vorzeitig reagieren können, indem entweder vorbeugende Maßnahmen getroffen oder neue Prozessinstanzen gestartet werden. Als Beispiel dient die Lieferung von verderblichen Gütern, bei der aufgrund von Ereignissen (etwa zu hoher Temperatur im Kühlcontainer) ein Fehler (eine mangelhafte Lieferung) vorhergesagt werden kann.

Ich präsentiere einen Ansatz, der mittels rekurrenten und gefalteten ANNs (recurrent and convolutional ANNs) die während eines Geschäftsprozesses auftretenden Ereignisse konsolidiert und für jeden zukünftigen Prozessschritt eine Fehlerwahrscheinlichkeit generiert. Hierbei präsentiere ich auch Optimierungstechniken zur Eingrenzung des Suchraums, um etwa mit Prozessschleifen umzugehen und lange Laufzeiten der Vorhersage zu verhindern. Die Aggregation der Ergebnisse zeigt eine Präzision von 87 % in der Fehlervorhersage.

4.3 Reduktion von Skaliervorgängen im Data Stream Processing

DSP-Systeme zeichnen sich oft durch variierende Last aus und müssen demnach auch Skalierbarkeit und Elastizität aufweisen [AdSVB18]. Ein dabei auftretendes Problem liegt in den indirekten Kosten, die durch Skaliervorgänge entstehen: Insbesondere das Aktivieren von Ressourcen kann Kosten verursachen, etwa das Hochfahren von zusätzlichen VMs [GGW10, MH11]. Somit sind Skaliervorgänge, auch wenn sie für Elastizität notwendig sind, auf das notwendige Minimum zu begrenzen. Im Kapitel 5 meiner Arbeit sowie in der dazugehörigen Publikation [BHS19] betrachte ich ein DSP-System mit seiner Variabilität als dynamisches System, von dem die Systemlast gemessen wird, kurzzeitige Schwankungen hierbei allerdings als unerwünschtes Rauschen herausgerechnet werden sollen. Dies trägt zu einer besseren Vorhersage der tatsächlichen Systemlast bei.

Ich verwende hierfür eine EKF-basierte Methodik und beschreibe, wie diese in einem DSP-System angewendet werden kann. Für die Evaluierung verwende ich unter anderem einen Echtdatensatz aus einem biomedizinischen Labor, bei dem Mikroskopaufnahmen für Tissue Engineering verarbeitet werden. Die variable Last wurde aus den im Labor anfallenden Datenmengen übernommen, um die Experimente möglichst nahe an der Realität

zu halten. Ich erreiche in meinen Experimenten unter anderem eine Verminderung der Skaliervorgänge um bis zu 88 %, führe aber auch eine Kostenanalyse durch, in der ich detailliert beschreibe, in welchen Fällen der EKF-basierte Ansatz vorteilhaft ist.

5 Fazit

In meiner Dissertation zeige ich, dass der Einsatz von Vorhersage-basierten Ansätzen für Ressourcenplanung in verteilten Systemen die Performanz und Verlässlichkeit der Systeme deutlich steigert. Die Verwendung von maschinellem Lernen führt dabei zu einer maßgeblichen Verbesserung der Wirtschaftlichkeit. Ich präsentiere konkrete Ansätze, die auf dynamische, selbstlernende und autonome Art und Weise verteilten Systemen ermöglichen, proaktiv auf Lastwechsel und eine sich ändernde Umgebung zu reagieren. Die gemessenen Werte zeigen eine Verbesserung von bis zu 88 % im Vergleich zum Stand der Technik.

Literaturverzeichnis

[AdSVB18] Assunção, Marcos Dias; da Silva Veith, Alexandre; Buyya, Rajkumar: Distributed Data Stream Processing and Edge Computing: A Survey on Resource Elasticity and Future Directions. Journal of Network and Computer Applications, 103:1–17, 2018.

[Ah17] Ahmad, Waqar; Hasan, Osman; Pervez, Usman; Qadir, Junaid: Reliability modeling and analysis of communication networks. Journal of Network and Computer Applications, 78:191–215, 2017.

[Ar10] Armbrust, Michael; Fox, Armando; Griffith, Rean; Joseph, Anthony D.; Katz, Randy; Konwinski, Andy; Lee, Gunho; Patterson, David; Rabkin, Ariel; Stoica, Ion; Zaharia, Matei: A View of Cloud Computing. Communications of the ACM, 53(4):50–58, 2010.

[BDKM97] Ben-David, Shai; Kushilevitz, Eyal; Mansour, Yishay: Online Learning versus Offline Learning. Machine Learning, 29(1):45–63, 1997.

[BHS19] Borkowski, Michael; Hochreiner, Christoph; Schulte, Stefan: Minimizing Cost by Reducing Scaling Operations in Distributed Stream Processing. PVLDB, 12(7):724–737, 2019.

[Bo16] Botta, Alessio; de Donato, Walter; Persico, Valerio; Pescapé, Antonio: Integration of Cloud Computing and Internet of Things: A Survey. Future Generation Computer Systems, 56:684–700, 2016.

[Bo19a] Borkowski, Michael; Fdhila, Walid; Nardelli, Matteo; Rinderle-Ma, Stefanie; Schulte, Stefan: Event-Based Failure Prediction in Distributed Business Processes. Information Systems, 81:220–235, 2019.

[Bo19b] Borkowski, Michael; Sigwart, Marten; Frauenthaler, Philipp; Hukkinen, Taneli; Schulte, Stefan: DeXTT: Deterministic Cross-Blockchain Token Transfers. IEEE Access, 7(1):111030–111042, 2019.

[Bo20] Borkowski, Michael: Predictive Approaches for Resource Provisioning in Distributed Systems. Dissertation, TU Wien, 2020.

[BSH16] Borkowski, Michael; Schulte, Stefan; Hochreiner, Christoph: Predicting Cloud Resource Utilization. In: 9th IEEE/ACM International Conference on Utility and Cloud Computing (UCC). IEEE/ACM, S. 37–42, 2016.

[Ca16] Cardellini, Valeria; Grassi, Vincenzo; Lo Presti, Francesco; Nardelli, Matteo: Optimal Operator Placement for Distributed Stream Processing Applications. In: 10th ACM International Conference on Distributed and Event-based Systems (DEBS). ACM, S. 69–80, 2016.

[Ca18a] Cardellini, Valeria; Grbac, Tihana Galinac; Nardelli, Matteo; Tanković, Nikola; Truong, Hong-Linh: QoS-Based Elasticity for Service Chains in Distributed Edge Cloud Environments. In: Autonomous Control for a Reliable Internet of Services, S. 182–211. Springer, 2018.

[Ca18b] Cardellini, Valeria; Lo Presti, Francesco; Nardelli, Matteo; Russo Russo, Gabriele: Optimal Operator Deployment and Replication for Elastic Distributed Data Stream Processing. Concurrency and Computation: Practice and Experience, 30(9):article e4334, 2018.

[CFF14] Corradi, Antonio; Fanelli, Mario; Foschini, Luca: VM Consolidation: A Real Case Based on OpenStack Cloud. Future Generation Computer Systems, 32:118–127, 2014.

[CLN12] Chaisiri, Sivadon; Lee, Bu-Sung; Niyato, Dusit: Optimization of Resource Provisioning Cost in Cloud Computing. IEEE Transactions on Services Computing, 5(2):164–177, 2012.

[Co13] Copil, Georgiana; Moldovan, Daniel; Truong, Hong-Linh; Dustdar, Schahram: Multi-level Elasticity Control of Cloud Services. In: International Conference on Service-Oriented Computing (ICSOC). LNCS 8274. Springer, S. 429–436, 2013.

[Du11] Dustdar, Schahram; Guo, Yike; Satzger, Benjamin; Truong, Hong-Linh: Principles of Elastic Processes. IEEE Internet Computing, 15(5):66–71, 2011.

[Fd12] Fdhila, Walid; Rinderle-Ma, Stefanie; Baouab, Aymen; Perrin, Olivier; Godart, Claude: On Evolving Partitioned Web Service Orchestrations. In: 5th IEEE International Conference on Service-Oriented Computing and Applications (SOCA). S. 1–6, 2012.

[GGW10] Gong, Zhenhuan; Gu, Xiaohui; Wilkes, John: PRESS: Predictive Elastic Resource Scaling for Cloud Systems. In: International Conference on Network and Service Management (CNSM). IEEE, S. 9–16, 2010.

[Gr15] Gracia-Tinedo, Raúl; Tian, Yongchao; Sampé, Josep; Harkous, Hamza; Lenton, John; García-López, Pedro; Sánchez-Artigas, Marc; Vukolic, Marko: Dissecting UbuntuOne: Autopsy of a Global-Scale Personal Cloud Back-End. In: Internet Measurement Conference (IMC). ACM, S. 155–168, 2015.

[Gu12] Gulisano, Vincenzo; Jimenez-Peris, Ricardo; Patino-Martinez, Marta; Soriente, Claudio; Valduriez, Patrick: StreamCloud: An Elastic and Scalable Data Streaming System. IEEE Transactions on Parallel and Distributed Systems, 23(12):2351–2365, 2012.

[Ha98] Haykin, Simon: Neural Networks: A Comprehensive Foundation. Prentice Hall, 2nd edition. Auflage, 1998.

[Ha17] Hao, Fang; Kodialam, Murali; Lakshman, T. V.; Mukherjee, Sarit: Online Allocation of Virtual Machines in a Distributed Cloud. IEEE/ACM Transactions on Networking, 25(1):238–249, 2017.

[HKR13] Herbst, Nikolas Roman; Kounev, Samuel; Reussner, Ralf H: Elasticity in Cloud Computing: What It Is, and What It Is Not. In: 10th International Conference on Autonomic Computing (ICAC). Jgg. 13. USENIX, S. 23–27, 2013.

[Hu14] Huang, Gao; Song, Shiji; Gupta, Jatinder N. D.; Wu, Cheng: Semi-Supervised and Unsupervised Extreme Learning Machines. IEEE Transactions on Cybernetics, 44(12):2405–2417, 2014.

[Ja07] Jazwinski, Andrew H.: Stochastic Processes and Filtering Theory. Dover, 2007.

[KB61] Kalman, Rudolph E.; Bucy, Richard S.: New Results in Linear Filtering and Prediction Theory. Journal of Basic Engineering, 83(1):95–108, 1961.

[La96] Langley, Pat: Elements of Machine Learning. Morgan Kaufmann, 1996.

[LBMAL14] Lorido-Botran, Tania; Miguel-Alonso, José; Lozano, Jose Antonio: A Review of Auto-Scaling Techniques for Elastic Applications in Cloud Environments. Journal of Grid Computing, 12(4):559–592, 2014.

[MH11] Mao, Ming; Humphrey, Marty: Auto-Scaling to Minimize Cost and Meet Application Deadlines in Cloud Workflows. In: International Conference for High Performance Computing, Networking, Storage and Analysis (SC). IEEE, 2011. article 49.

[PLM17] Petrolo, Riccardo; Loscri, Valeria; Mitton, Nathalie: Towards a Smart City Based on Cloud of Things, a Survey on the Smart City Vision and Paradigms. Transactions on Emerging Telecommunications Technologies, 28(1):article e2931, 2017.

[SA13] Sathya, Ramadass; Abraham, Annamma: Comparison of Supervised and Unsupervised Learning Algorithms for Pattern Classification. International Journal of Advanced Research in Artificial Intelligence, 2(2):34–38, 2013.

[Sc15] Schulte, Stefan; Janiesch, Christian; Venugopal, Srikumar; Weber, Ingo; Hoenisch, Philipp: Elastic Business Process Management: State of the Art and Open Challenges for BPM in the Cloud. Future Generation Computer Systems, 46:36–50, 2015.

[Zh18] Zheng, Zibin; Xie, Shaoan; Dai, Hong-Ning; Chen, Xiangping; Wang, Huaimin: Blockchain Challenges and Opportunities: A Survey. International Journal of Web and Grid Services, 14(4):352–375, 2018.

Michael Borkowski wurde am 26. März 1991 in Wien geboren. Er schloss 2015 sein Diplomstudium im Bereich Software Engineering an der TU Wien ab. Von 2015 bis 2019 war er als Projektassistent und Doktorand an der Distributed Systems Group der TU Wien tätig, wo er in den Bereichen des maschinellen Lernens, der verteilten Rechensysteme sowie Blockchains forschte und 2020 mit Auszeichnung promovierte. Seit April 2019 ist er am Deutschen Zentrum für Luft- und Raumfahrt (DLR) in Braunschweig als wissenschaftlicher Mitarbeiter im Bereich Unbemannte Luftfahrzeugsysteme (UAS) tätig und wirkt unter anderem an zahlreichen H2020-EU-Projekten mit. Seine wissenschaftliche Arbeit umfasst 22 begutachtete Publikationen, darunter fünf Artikel in wissenschaftlichen Fachzeitschriften, unter anderem in Elsevier Information Systems sowie Proceedings of the VLDB Endowment und eine mit *Best Paper Award* ausgezeichnete Publikation bei der IEEE International Conference on Blockchain.