

Kubernetes-native Messung der Energieeffizienz und adaptives Scheduling in heterogenen Clustern

Thomas Stadler, BSc

FH-Burgenland, Eisenstadt, Österreich

KURZFASSUNG: Der Leser der Kurzfassung soll verstehen, welche Problemstellung / Fragestellung durch die vorliegende Arbeit bearbeitet wird und welche Erkenntnisse und Ergebnisse vorliegen.

1 EINLEITUNG UND PROBLEMHINTERGRUND

Rechenzentren bilden den größten Teil der Infrastruktur, auf der moderne verteilte service-orientierte Systeme beruhen. Es handelt sich um komplexe Systeme mit vielen interagierenden Elementen, die enorme Mengen an Energie verbrauchen. Die Nachfrage nach Rechenzentren steigt rapide an, was zu erheblichen globalen Umweltauswirkungen führt. Die Rechenzentrumsbranche hat bereits viel Forschung zur Effizienzverbesserung betrieben, allerdings hauptsächlich auf der Ebene der physischen Infrastruktur. Die Erforschung von softwarebasierten Lösungen zur Verbesserung der Effizienz ist dringend erforderlich. Die meisten aktuellen Forschungsarbeiten betrachten das Rechenzentrum jedoch nicht aus einem ganzheitlichen Blickwinkel, der sowohl virtuelle und physische Infrastrukturen als auch Geschäftsprozesse berücksichtigt. Dies ist entscheidend, wenn eine Lösung in einer realistischen Umgebung angewendet werden soll.

Immer mehr Softwarehersteller setzen auf cloud-native Architektur, allermeistens basierend auf der open-source Container-Plattform Kubernetes, welche als Basis für Workloads aller Art, sowie für cloud-native Control Planes gilt. Kubernetes ist unter der Governance der Cloud Native Computing Foundation (CNCF) seit 10. März 2016 und verfügt über den Reifegrad *graduated*.

Mit Kubernetes lassen sich als Container abgepackte Microservices im großen Maßstab auf mehreren tausend Nodes orchestrieren. Das Monitoring zielt derzeit hauptsächlich auf Metriken der CPU und Arbeitsspeicher-Auslastung ab. Dabei können für jeden Kubernetes Workload diese Metriken gemessen und somit feingranular überwacht werden, sowie im Anschluss weitere Automatisierung, gerichtet auf diversen Service Level Objectives (SLO) stattfinden, oder beispielsweise horizontales und vertikales Auto-Scaling konfiguriert werden, neben anderen Automatisierungen.

Derzeit kann der Stromverbrauch nur je Node gemessen werden. Das Problem ist, dass in Kubernetes ein Microservice bzw. Workload nicht mehr wie vor der Zeit der Virtualisierung und Containerisierung, auf einer physischen oder virtuellen Maschinen betrieben wird, sondern sich über potenziell tausende Nodes verteilt. Kubernetes Workloads sind in sogenannte Pods logisch in kleine Einheiten aufgeteilt. Im Optimalfall enkapsuliert ein Pod einen oder mehrere verwandte Container, in

welcher jeweils ein Prozess läuft. Somit ist es äußerst schwierig, den Stromverbrauch einer Applikation festzustellen.

2 STAND DES WISSENS / STAND DER TECHNIK

Colmant et al. (2015) erforschen "Process-level Power Estimation in VM-based Systems". Dabei wird die Software BitWatts implementiert, welches auf dem Python-Framework PowerAPI basiert. Lösungen auf dem aktuellen Stand der Technik können eine grobkörnige Leistungsabschätzung in virtualisierten Umgebungen durchführen, wobei virtuelle Maschinen (VMs) in der Regel als Blackbox behandelt werden. Colmant et al. (2015) schlagen eine feinkörnige Überwachungs-Middleware vor, die eine genaue Leistungsabschätzung von Softwareprozessen in Echtzeit ermöglicht, welche auf jeder Virtualisierungsebene in einem System laufen. Die Middleware-Implementierung mit dem Namen BitWatts baut auf einer verteilten Actor-Implementierung auf, um die Prozessnutzung zu erfassen und einen feinkörnigen Stromverbrauch abzuleiten, ohne dass Hardware-Investitionen, wie Stromzähler erforderlich sind (Colmant et al., 2015).

Fieni et al. (2020) forschen in "SmartWatts: Self-Calibrating Software-Defined Power Meter for Containers" über ein leichtgewichtiges Energieüberwachungssystem, das die CPU- und DRAM-Leistungsmodelle durch Online-Kalibrierung automatisch anpasst, um die Genauigkeit der Leistungsabschätzung von Containern während der Laufzeit zu maximieren. Im Gegensatz zu state-of-the-art Techniken erfordert SmartWatts keine vorherige Trainingsphase oder Hardware-Ausrüstung zur Konfiguration der Leistungsmodelle und kann daher ohne Kosten auf einer Vielzahl von Rechnern eingesetzt werden, einschließlich der neuesten Leistungsoptimierungen (Fieni et al., 2020).

Townend et al. (2019) beschreiben die komplexe System-of-Systems-Natur von Rechenzentren und diskutieren die in der Branche verwendeten Servicemodelle. Sie beschreiben ein ganzheitliches Scheduling-System, das den Standard-Scheduler im Kubernetes-Container-System ersetzt und sowohl Software- als auch Hardware-Modelle berücksichtigt. Sie erörtern die ersten Ergebnisse des Einsatzes dieses Schemas in einem realen Rechenzentrum, in dem eine Reduzierung des Stromverbrauchs um 10-20 % beobachtet wird. Es wird gezeigt, dass ein intelligenter Scheduler durch die Einführung von Hardwaremodellen in eine softwarebasierte Lösung die Effizienz von Rechenzentren erheblich verbessern kann (Townend et al., 2019).

3 WISSENSCHAFTLICHE FRAGESTELLUNG

In dieser Arbeit soll die Messung des Stromverbrauchs auf Ebene von Kubernetes Workloads erforscht werden. Dies würde es beispielsweise erlauben, einen stromverbrauchs-orientierten Kubernetes-nativen Scheduler zu implementieren, welcher Nodes für Workloads adaptiv bestimmt, je nachdem auf welchen Node-Typen ein bestimmter Workload die beste Energieeffizienz aufweist. Node-Typen können sich beispielsweise unterscheiden durch verschiedene Prozessor-Architekturen

Das Ziel dieser Arbeit ist es, einen Einblick zu gewinnen, inwieweit sich die Messung des Stromverbrauchs von Kubernetes Workloads nativ in Kubernetes implementieren lässt, sodass gemessene Metriken auf der Ebene von Kubernetes Workloads für weitere Automatisierungen und Kubernetes-native Integrationen zur Verfügung stehen.

Die konkrete Forschungsfrage lautet:

Inwieweit lässt sich der Stromverbrauch von Kubernetes Workloads bzw. Applikationen feststellen und ein effizienz-orientiertes Scheduling implementieren?

4 FORSCHUNGSMETHODIK

Zuerst wird bestehende wissenschaftliche Literatur zum Thema, sowie der konkreten Forschungsfrage herangezogen. Weiters werden existierende Softwarelösungen, die unter Open-Source-Lizenzen zur Verfügung stehen, für den zweckmäßigen Einsatz für die vorliegende Forschung evaluiert. Schließlich wird eine Baseline als State-of-the-Art definiert.

Sind existierende Lösungen bzw. wissenschaftliche Prototypen für das vorliegende Forschungsvorhaben geeignet, dann werden diese genutzt und erweitert oder abgeändert um für den Zweck dieser Forschung zu dienen. Andernfalls wird ein Prototyp für die Kubernetes-native Stromverbrauchs-Messung auf Workload-Ebene entwickelt.

Nach Houde und Hill (1997) ist ein Prototyp jede Darstellung einer Design-Idee, unabhängig vom Medium. Ein Prototyp ist etwas, das als Modell oder Inspiration für spätere Entwicklungen dient (Houde & Hill, 1997).

Nach erfolgreicher Implementierung des Prototyps, soll mittels einer Scheduler-Extension in einem Laborexperiment gezeigt werden, dass Kubernetes Workloads adaptiv Nodes zugeordnet werden können, auf welchen diese die beste Energieeffizienz aufweisen.

Montgomery (2017) definiert ein Experiment als einen Test oder eine Reihe von Durchläufen, wobei absichtlich Änderungen an den Eingangsvariablen vorgenommen werden, um dann Gründe für Veränderungen am Ausgangsergebnis beobachten und identifizieren zu können (Montgomery, 2017).

5 ERGEBNISSE

Die Ergebnisse der bisherigen Literaturrecherche zeigen, dass bereits wissenschaftliche Untersuchungen zum Thema der software-basierten dynamischen Stromverbrauchs-Messungs unternommen wurden. Im Bereich von VM-basierten Systemen, prozess-basierter Messung, sowie container-basierter Messung von Stromverbrauch. Dabei wird oft der Gesamtverbrauch der physischen Maschine - primär die CPU-Auslastung und deren Stromverbrauch - mittels Hardware-Indikatoren und Low-Level-Schnittstellen, welche die Hardware zur Verfügung stellt, ausgelesen, und mittels mathematischer Kalkulationen auf kleinere logische Einheiten versucht zuzuordnen.

Als primäres Ergebnis dieser Arbeit wird die Kubernetes-native Messung des Stromverbrauchs von Kubernetes Workloads erwartet. Der entwickelte Prototyp soll als modulare Extension zu Kubernetes dienen. Es sollen die gewonnenen Metriken über den OpenTelemetry-Standard, für Observability-Werkzeuge als Service Level Indicators (SLI) zur Verfügung stehen.

In einem Laborexperiment soll bewiesen werden, dass anhand der preisgegebenen Metriken des Kubernetes Workload basierten Stromverbrauchs Kubernetes Workloads adaptiv auf geeignete Nodes zugeordnet werden können, auf welchen sie am effizientesten laufen.

6 SCHLUSSFOLGERUNGEN UND AUSBLICK

Mit rasant steigender Nachfrage an Rechenzentren, welche zu einem großen Teil zum gesamten Stromverbrauch auf der Erde beitragen, wird es immer wichtiger, auf die Energieeffizienz zu achten. Energieeffizienz lässt sich an verschiedenen Punkten verbessern. In dieser Arbeit soll die Effizienz von Workloads bzw. Applikationen in Kubernetes dynamisch bestimmt werden. Schließlich sollen mit den gewonnenen Metriken Workloads zu bestimmten Nodes zugeordnet werden können, auf welchen sie am effizientesten laufen. Bestimmte Algorithmen funktionieren beispielsweise effizienter oder weniger effizient auf verschiedenen Prozessor-Architekturen. Das Auswählen der optimalen Node bzw. zugrundeliegenden Hardware soll völlig transparent funktionieren.

References

- Colmant, M., Kurpicz, M., Felber, P., Huertas, L., Rouvoy, R., & Sobe, A. (2015). Process-level Power Estimation in VM-based Systems. In T. Harris & M. Herlihy (Hrsg.), *European Conference on Computer Systems (EuroSys)* (S. 14). ACM.
<https://doi.org/10.1145/2741948.2741971>
- Fieni, G., Rouvoy, R., & Seinturier, L. (2020). SmartWatts: Self-Calibrating Software-Defined Power Meter for Containers. *CCGRID 2020 - 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing*. <https://doi.org/10.1109/CCGrid49817.2020.00-45>
- Houde, S., & Hill, C. (1997). Chapter 16 - What do Prototypes Prototype? In M. G. Helander, T. K. Landauer & P. V. Prabhu (Hrsg.), *Handbook of Human-Computer Interaction (Second Edition)* (Second Edition, S. 367–381). North-Holland.
<https://doi.org/https://doi.org/10.1016/B978-044481862-1.50082-0>
- Montgomery, D. C. (2017). *Design and analysis of experiments*. John wiley & sons.
- Townend, P., Clement, S., Burdett, D., Yang, R., Shaw, J., Slater, B., & Xu, J. (2019). Invited Paper: Improving Data Center Efficiency Through Holistic Scheduling In Kubernetes. *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, 156–15610.
<https://doi.org/10.1109/SOSE.2019.00030>