The CACTOS Vision of Context-Aware Cloud Topology Optimization and Simulation

Ostberg, P. O., Groenda, H., Wesner, S., Byrne, J., Nikolopoulos, D. S., Sheridan, C., Krzywda, J., Ali-Eldin, A., Tordsson, J., Elmroth, E., Stier, C., Krogmann, K., Domaschka, J., Hauser, C. B., Byrne, PJ., Svorobej, S., McCollum, B., Papazachos, Z., Whigham, D., ... Paurevic, D. (2014). The CACTOS Vision of Context-Aware Cloud Topology Optimization and Simulation. In *2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)* (pp. 26-31). Institute of Electrical and Electronics Engineers Inc.. https://doi.org/10.1109/CloudCom.2014.62

Published in:
2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)

Document Version:
Peer reviewed version

The CACTOS Vision of Context-Aware Cloud Topology Optimization and Simulation

Per-Olov Östberg*, Henning Groenda†, Stefan Wesner‡, James Byrne§, Dimitrios S. Nikolopoulos¶, Craig Sheridan‖, Jakub Krzywda*, Ahmed Ali-Eldin*, Johan Tordsson*, Erik Elmroth*, Christian Stier†, Klaus Krogmann†, Jörg Domaschka‡, Christopher B. Hauser‡, PJ Byrne§, Sergej Svorobej§, Barry McCollum¶, Zafeirios Papazachos¶, Darren Whigham‖, Stephan Rüth**, Dragana Paurevic**

*Dept. of Computing Science, Umeå University, Sweden
{p-o, jakub, ahmeda, tordsson, elmroth}@cs.umu.se

†Software Engineering, Forschungszentrum Informatik, Germany
{groenda, stier, krogmann}@fzi.de

‡Institute of Information Resource Management, Ulm University, Germany
{stefan.wesner, joerg.domaschka, christopher.hauser}@uni-ulm.de

§DCU Business School, Dublin City University, Ireland
{james.byrne, pj.byrne, sergej.svorobej2}@dcu.ie

¶School of EEECS, Queen's University of Belfast, United Kingdom
{d.nikolopoulos, b.mccollum, z.papazachos}@qub.ac.uk

‖Flexiant Ltd., Edinburgh, Scotland
{csheridan, dwhigham}@flexiant.com

**Realtech AG., Walldorf, Gemany
{stephan.rueth, dragana.paurevic}@realtech.com

***Abstract*—Recent advances in hardware development coupled with the rapid adoption and broad applicability of cloud computing have introduced widespread heterogeneity in data centers, significantly complicating the management of cloud applications and data center resources. This paper presents the CACTOS approach to cloud infrastructure automation and optimization, which addresses heterogeneity through a combination of in-depth analysis of application behavior with insights from commercial cloud providers. The aim of the approach is threefold: to model applications and data center resources, to simulate applications and resources for planning and operation, and to optimize application deployment and resource use in an autonomic manner. The approach is based on case studies from the areas of business analytics, enterprise applications, and scientific computing.**

I. Introduction

The broad cross-domain applicability of cloud computing has lead to the emergence of a variety of technological options and resource profiles, and a substantial degree of heterogeneity in data center resources and service offerings. For example, cloud computing infrastructures now more commonly feature specialized hardware such as many-core systems or general purpose computing on graphics processing units (GPGPUs), as well as CPU architectures, infrastructure layouts, and facility management tools optimized for energy efficiency.

In addition, cloud data centers are also exhibiting non-negligible resource heterogeneity [6] stemming from the acquisition of new resources, incremental upgrades of existing resource sets, and successive changes in resource configuration and software deployment. At the level of software stacks, cloud service models have evolved from Infrastructure-as-a-Service (IaaS) oriented models to more advanced and complex service models, e.g., interactive service offerings such as remote rendering or gaming applications [20], featuring stacks of complex services on top of basic infrastructure and platform services. These increases in scale, heterogeneity, and complexity necessitate development of autonomous and automated data center optimization and management tools. To cope with the challenges of optimizing automated mappings of services to hardware and software (e.g., virtual machine and container) resources, such tools require topology-aware mapping techniques that consider holistic optimization of placement of services across heterogeneous data centers.

This paper presents the CACTOS vision of Context-Aware Cloud Topology Optimization and Simulation (Section II), identifies some of the challenges that need to be addressed to realize this vision (Section III), and discusses the approach taken of the work (Section IV). Initial results (Section V) are presented along with discussions of the planned validation of results (Section VI) and related work (Section VII).

II. The CACTOS Vision

The vision of CACTOS is to produce new data center optimization and simulation mechanisms that can handle the scale, heterogeneity, and complexity of modern cloud application workloads while providing advanced infrastructure capabilities such as resource elasticity and controllable application quality of service (QoS). The long-term goal of this work is to develop integrated monitoring, simulation, and management tools that accurately capture the dynamics of complex workloads, abstract the heterogeneity of resource sets, and optimize virtual machine and resource configurations to increase the resource

and energy efficiency of cloud data centers. To this end, CACTOS emphasizes the three core concepts:

Context-awareness. Applications and resources are modelled both individually and together to accurately reflect not only the direct interactions between applications and resources but also the impact co-location and scheduling of workloads have on resource pool effectiveness and application service-level objectives. Modeling of application behavior as well as prediction of future resource requirements at both component (virtual machine) and system (application) level are used for infrastructure optimization.

Topology optimization. Data center topology (resource and resource configuration) optimization mechanisms are designed as autonomic systems [17] for semi- and fully automated infrastructure tuning and control. Mapping of virtual machines (VMs) to resources (scheduling and placement), control of the amount of resource capacity allocated to virtual machines, as well as control of admission and migration of workloads are used to construct elastic and controllable cloud data centers.

Simulation. Discrete-event simulation techniques are used to model, simulate, and evaluate software and hardware resources in large-scale, heterogeneous data center environments. Simulations are used to evaluate the effectiveness of optimization strategies as well as for iterative resource planning and operations decision support.

III. Challenges

A number of challenges exist for realizing the CACTOS vision. From the perspective of a cloud data center operator, we here describe a set of challenges that (primarily) arise from the heterogeneity of cloud environments.

A. Cloud System Scale and Complexity

The scale of cloud applications ranges from basic services running in individual virtual machines to complex and distributed applications spanning multiple services hosted in multiple geographically distributed data centers. The complexity of cloud applications consisting of multiple distributed services quickly becomes an issue as the behavior and performance of individual participating services and resources determine the overall performance of the application. While complex, these non-linear effects must be taken into account when reasoning in topology optimization.

Efficient management of cloud applications requires in-depth knowledge of the resource, capacity, and environmental requirements of individual components in applications. To complicate the matter, service-level objectives are typically formulated in terms of high-level application-oriented quality-of-service metrics such as response time or throughput, which are poorly aligned with the monitoring metrics resource management systems operate on, e.g., resource utilization. Furthermore, while service-level objectives are typically defined at service level, the mappings between intended service behavior and resource performance are often poorly understood and (at best) based on coarse-grained models derived from empirical experiences. Monitoring tools typically provide metrics at a level distinctly different (i.e. much lower) from that of service-level objectives. These differences in abstraction levels require placement algorithms and topology optimization models to incorporate translation functions that map from low-level monitoring metrics to domain-specific high-level application metrics and take into account the behavior of components in the application. In addition to complicating the management process itself, this also introduces discrepancies in simulation models, complicating accurate emulation of data center environments for system evaluation purposes.

Current data centers are now at a scale that in itself prohibits the use of global optimization techniques. Optimized placement of virtual machines must now (when considering proactive resource scheduling) be done in hierarchical layers, e.g., first assigning virtual machines to clusters, then scheduling them within clusters. Holistic optimization of data center operations must consider not only virtual machine placement, but also factors such as resource configuration, load mixing, impact of workload co-location, energy consumption, and heat production constraints, as well as external constraints specified by application owners or site administrators.

In these settings, topology optimization becomes a multi-criteria optimization problem with multiple (often conflicting) objective functions. Taken together, the scale and complexity of these optimization problems require use of near-optimal solutions, such as heuristics-based optimization strategies.

B. Cloud Workload and Infrastructure Heterogeneity

Heterogeneity permeates both cloud workloads and infrastructures at multiple levels. Modelling and prediction of workloads requires understanding of application behavior and heterogeneous workload characteristics [8]. Due to the varied use of cloud resources, workload execution time requirements range from sub-microsecond transaction task executions to continuously running services that measure uptime in months or years. Similarly, resource requirements may vary as memory, storage, I/O, network bandwidth, and CPU requirements vary with factors such as parametrization of problems, and degree and granularity of parallelism in computations.

Application resource demand depends heavily on the resource usage profile. This can range from single periodic requests to burst periods with several orders of magnitude changes of seasonal patterns for resource requirements, e.g., in financial calculations at the end of a business quarter.

Modelling and characterizing changes in the behavior of heterogeneous workloads on heterogeneous hardware poses major challenges. Understanding the performance, scalability, energy, and resilience implications of resource heterogeneity, elasticity, and contention for cloud workloads is essential for optimizing data center infrastructures and services.

Hardware heterogeneity in data center infrastructures tend to either be a result of design (e.g., inclusion of specialized resources to complement general purpose resource sets) or the product of incremental resource upgrades over time. Regardless of source, variations in resource capacity, capability, and topology complicates optimization of data center operations.

IV. Infrastructure Topology Optimization

Key challenges in cloud infrastructure topology optimization include the identification of monitorable key performance indicators and management actions that can be used to control data center resources. In CACTOS, data centers are modeled in a sensor-actuator model where sensor (monitoring) data are captured in *infrastructure topology and load models* and actuator actions are represented in *optimization plans* that list recommended changes to the infrastructure using instructions from a predefined optimization plan language. Using this model, data center operations are then described in a closed *Observe-Plan-Act* loop, where the state of the data center resources and applications are continuously monitored, and plans (changes to resource configurations and application mappings) are made and enacted to optimize data center operations towards selected objective functions.

A. Observe (Monitoring and Data Analysis)

Monitoring information can be broadly classified into three categories: resource (hardware), application (software), and fault (anomaly) data. For data center resources, modeling of resource status is typically done through monitoring of available resource hardware capacity, e.g., the number and status of operational servers. In CACTOS, this data is captured in infrastructure topology and load models that describe the structure (physical and virtual topology models) of the data center resources at a fine-grained level and the load applications place on these resources. The CACTOS load models track, e.g., resource load (in dimensions such as CPU, RAM, network, and storage I/O), capacity and utilization of shared mediums (such as network bandwidth and storage capacity), as well as indirect properties such as data center energy consumption and heat production.

The behavior of applications running in data centers can be similarly modelled using benchmarking and monitoring techniques to quantify application-level resource usage patterns (e.g., CPU, RAM, and storage usage). In application behavior modeling, it is often important to nuance the sensitivity of applications in resource capacity dimensions to capture how changes in available resource capacity affect application performance. For cloud applications, metrics that explicitly focus on the characteristics of cloud data center deployment are needed [1]. In CACTOS, application resource capacity request patterns are analyzed to build application behavior models that capture the context of cloud deployment and execution.

Application behavior models facilitate classification of applications (to, e.g., distinguish CPU-intensive computational applications from I/O-intensive transaction systems) as well as description of phases in application behavior (which can be used to, e.g., recognize and predict I/O-intensive phases that should not be scheduled to overlap with I/O-intensive phases of co-hosted applications). Application behavior models can also be used to detect anomalous (application and resource) behavior, e.g., by monitoring and detecting large deviations in expected application performance, or to perform simulation-based application characterization experiments, e.g., to model how sensitive response time oriented interactive applications are to changes in network latency and bandwidth.

B. Plan (Optimization Planning)

Planning can be described as the process of scheduling actions towards intended goal behaviors. In topology optimization, planning complexity is significantly increased by the presence of multiple conflicting and overlapping objective functions, e.g., maximization of factors such as computational throughput, resource utilization, and data center cost efficiency, combined with simultaneous minimization of factors such as application response time, energy consumption, heat production, and application SLA violations.

In CACTOS, topology optimization is performed on multiple granularity levels (ranging from fine-grained application tuning and resource configuration to holistic data center level), as well as at multiple time-scales (periodic and request-driven predictive planning of resource usage versus event-triggered planning for fault recovery). In topology optimization, planning finds applications not only in direct infrastructure optimization, but also in formulation of decision support systems that, e.g., allow system administrators to plan for future extensions and adaptations of data center infrastructures, as well as in scenario-driven (”what if”) analyses and simulation-driven experimentation for unexpected events (e.g., hardware failures and power outages).

C. Act (Execution of Plans)

Hardware-enabled (para-)virtualization techniques such as Xen and KVM allow construction of hypervisor-based infrastructure management tools that define resources as virtual machines and offer high management flexibility at low performance overheads. Virtual machines can be dynamically instantiated, started, paused, reconfigured, migrated, resumed, stopped, and removed using network-accessible application programming interfaces. From a non-management point of view, use of virtualization techniques also facilitates other significant benefits, e.g., application-specific execution environments and server consolidation. In addition, virtualization techniques can also be used to virtualize other parts of data center infrastructures, e.g., software-defined networks and virtual network overlays that redefine network topologies and enforce link-level bandwidth restrictions.

In CACTOS, virtualization technologies are used to define optimization actuators that control virtual data center resources through virtualization middlewares. The *optimization plans* formulated by the optimization engine are sent to middleware implementation components using a purposely developed optimization plan language, allowing the optimizer to give recommendations for, e.g., placement, scheduling, migration, and hardware assignments of virtual machines.

Taken together, the CACTOS Observe-Plan-Act loop allows construction of virtual infrastructures that can be (through planning and scheduling of data center operations) controlled to optimize the performance of data center infrastructures.

V. The CACTOS Toolkit

To tame heterogeneity the CACTOS toolkit is based on context models, e.g., infrastructure and application performance models. CACTOS introduces context awareness by modeling the implications of the availability and heterogeneity of the resources on which VMs are executed, the impact of workload co-location has on servers, as well as the impact environmental constraints have on application performance.

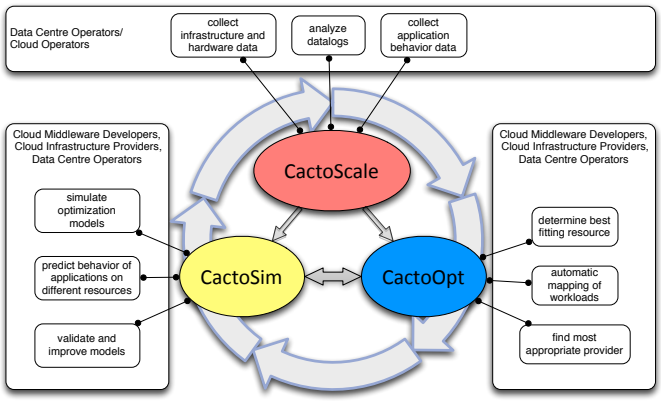

Fig. 1. The continuous cycle of the CACTOS Observe-Plan-Act loop.

As illustrated in Figure 1, the CACTOS toolkit for data center operations management and planning consists of three main components deployed in a continuous observe-plan-act feedback loop: *CactoScale* is a set of tools and methods to acquire and analyze application behavior and resource performance data. *CactoOpt* consists of mathematical models and their realizations to optimize application-resource mappings within a provider context. *CactoSim* is a prediction and simulation environment for diverse application workloads. It facilitates emulation of data centers and validation of optimization models in simulation environments.

A. CactoScale

The design of CactoScale aims at fast, scalable, and continuous processing of performance monitoring and log data streams. CactoScale integrates multiple sources of performance and error monitoring data into a consolidated architecture with a unified interface and storage architecture. The CactoScale interface enables both in-situ and off-loaded data processing. Monitoring agents co-located with data center servers collect log traces from user-defined data sources, including application-specific sources and system-level sensors. The agents (optionally) process this data in place before forwarding it to dedicated CactoScale servers. The storage architecture employs in-memory processing to cope with high volumes of log and trace data. The design integrates resilient and archival data logging off the critical path. This is essential for implementing resource management strategies based on historical data. CactoScale is based on Chukwa [3], with a modified design and implementation to enable:

- Flexible choice of processor architecture for log analytics, including choice of computational accelerators.
- Flexible choice of memory and storage architectures for logging and archiving.
- Placement of log processing nodes together with data center processing nodes and flexible choice of data path for log data delivery to processing nodes.
- Hybrid in-situ / off-loaded log analytics.

B. CactoOpt

CactoOpt employs a library of multi-objective optimization algorithms to model and optimize cloud data center operations towards a set of selected objective functions, e.g., energy efficiency parameters, resource utilization metrics, or negotiated quality-of-service requirements (SLAs). CactoOpt explores use of many different optimization approaches including, e.g., hybrid meta-heuristic and constraint programming approaches.

Hybridization of meta-heuristic approaches has in recent years proven to be effective in solving scheduling and related resource allocation problems [9]. This is particularly the case where multiple objectives are present within the evaluation function. In particular, hybridization has proven useful between population-based methods and other local search based approaches. In general, population-based methods are better in identifying promising areas within the overall search space, whereas local search methods are better in exploring resulting promising localized areas. Thus, meta-heuristic hybrids in some way manage to combine the advantages of population-based methods with the strength of local or trajectory methods.

As a complementary approach, constraint programming techniques [18] are used to formulate constraint-based rules for pruning and exploring optimization search spaces. Constraint-based programming approaches have the benefit of allowing formulation of optimization problems in terms of problem-specific constraints in problems that can be assessed using standardized solvers. CactoOpt also deploys evolutionary algorithms, such as genetic and ant colony algorithms, to the problem of optimizing resource provisioning under performance or energy constraints. The hybridization of these algorithms utilizes a global best model inspired from particle swarm optimization to enhance the global exploration ability while hybridizing with the great deluge algorithm in order to improve the local exploitation ability. Using this approach, an effective balance between exploration and exploitation is attained [9].

C. CactoSim

CactoSim is a discrete event simulation framework that allows experimentation with (and validation of) cloud optimization strategies in simulated environments. This offers significant benefit over the use of testbeds with high complexities and costs. CactoSim allows for reproducible and controlled experimentation with workload mix and resource performance evaluation scenarios, enabling both cost and risk analysis to be performed in conjunction with tuning of systems. Consequently, use of this tool leads to more robust optimization algorithms and enhanced decision support.

As illustrated in Figure 2, the simulation framework supports both behavior and system modelling of heterogeneous

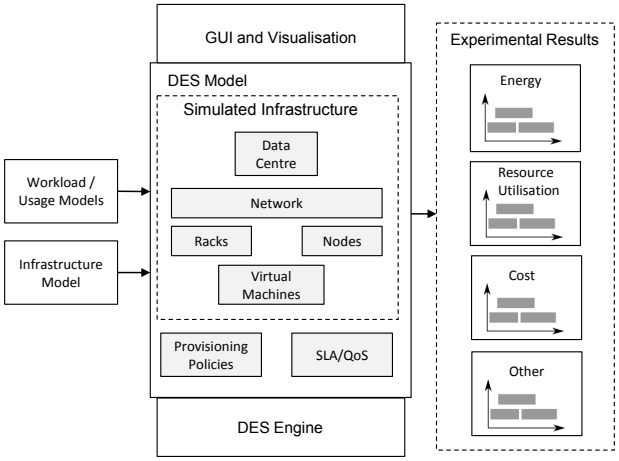

Fig. 2. CactoSim architecture.

components within a cloud computing environment, taking into account data centers, clusters, racks, hosts, and virtual machines as well as provisioning policies for resources and SLA/QoS constraints. Central to CactoSim is the ability to predict the behavior of different types of workloads running on different simulated infrastructures over time. As a prediction and experimentation platform, CactoSim is designed to output energy and cost conscious experimental results such as system energy consumption, resource utilization, application response time, and costs (related to operational costs and potential SLA violations).

VI. Validation

The planned validation of CACTOS results is based on use cases from the areas of business analytics, enterprise applications, and technical computing.

A. Enterprise Applications

Realtech runs a data center with a dedicated set of enterprise applications including an enterprise resource planning application in a physically and virtually heterogeneous environment featuring IBM LPAR, VMware ESX, Citrix XenServer and Microsoft HyperV. A typical use case is resource efficient deployment while maintaining SLA levels, e.g., in terms of end-to-end response times of defined user transactions. The current state of practice includes manual customization of hosted applications and deployment on physical machines. Automation and customization of this process will serve as validation of CactoSim results.

This case study is chosen for CACTOS as it reflects a typical setup for enterprise applications. The set of applications and deployment options are limited, but workload dependencies may heavily depend on parametrization. Optimization of proactive (re-)deployment is critical to fulfilling customer-specific SLAs and energy efficiency, and will in CACTOS serve as a validation scenario for CactoOpt results.

B. Business Analytics

Flexiant are using functionality provided by CactoSim and CactoOpt to provide intelligence in decision making for data center operations. Through CactoScale, the *Flexiant Cloud Orchestrator* platform exposes data about data center topology, resource identifiers, capabilities, and a set of key run-time metrics. This data is analyzed and used in CactoOpt to produce optimization recommendations to tune and optimize resource configurations and workload assignments in the platform.

In addition, the Flexiant platform allows collation and aggregation of cloud characterization data, e.g., performance and utilization metrics, that facilitates resource modeling and validation of results in CactoSim. Use of this data will represent a workflow to validate CACTOS results and have a direct correlation to moving beyond the current capabilities of cloud infrastructures. The benefits the Flexiant Cloud Orchestration software derives from CACTOS innovations include:

- Differentiation of compute nodes in order to match the most apt target option to the needs of the workload.
- Dynamic node management to spread workloads in a linear manner across available resources.
- Capture, store and utilize node utilization data for intelligent decision making.
- Power down unused compute nodes automatically.

In addition to a testbed, Flexiant also provides 3 years of real data related to node uptime, failure codes, VM resource usage and many other parameters. This part of the case study will support model calibration for CactoOpt as well as validate results from CactoScale and CactoSim.

C. Technical Computing

Technical or scientific computing is a term used for applications that based on a domain (e.g., physics or chemistry) model derive numerical models for simulation and prediction of system behaviors. For validation of CACTOS results, a computational quantum chemistry application with high computational requirements and a particular sensitivity to memory capacity and I/O speed is selected. The application execution time depends heavily on the settings for configuration interaction and coupled-cluster methods, and typically runs for several hours to days.

This type of application is chosen for CACTOS as the application behavior is comparably easy to predict, which allows modeling of computational phases and resource requirements in estimation of execution time on different hardware sets. While this type of application can be considered an extreme case of cloud deployment (due to its high demands on the underlying infrastructure), it will (as application executions exhibit predictable and slow changes in resource usage patterns) allow validation of different approaches for optimization models in CactoOpt, application behavior modelling in CactoSim, as well as application monitoring in CactoScale.

VII. Related Work

Workload and infrastructure modelling has been studied extensively for both academic [8] and commercial [10] work-

loads and deployment scenarios. The purpose of such evaluation is often goal-oriented, e.g., to understand task placement constraints in compute cluster scheduling [22], but rarely encompasses the challenges that arise from the growing heterogeneity of cloud infrastructures and workloads. A number of approaches dealing with cloud-specific challenges exist as well, e.g., the OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) [2] that aims to enhance the portability of cloud applications and services, and the Descartes Meta-Model [14], which covers modelling of workloads, cloud infrastructure, and the dynamics of operations that can be executed in cloud data centers [13].

A wide range of cloud optimization scenarios have been investigated [15] and demonstrated to improve data center efficiency, e.g., joint optimization of placement and routing [16], modelling of correlation-aware demand [5] and interference effects of co-located workloads [26], thermal management [21], energy and power consumption [11], workload migration [25], and incorporation of predictions of IT demand and renewable energy [19]. However, while these approaches all improve some aspect of cloud data center efficiency, currently no single framework exists to tackle the challenges of cloud heterogeneity and cloud deployment optimization in combination.

The current state-of-the-art in cloud simulators is compared in [24]. Examples of available cloud simulators include, e.g., GloudSim [7], simulators based on networks of queues [12], and CloudSim [4]. For resource modelling and evaluation, there exists also a number of cloud benchmarking tools including, e.g., C-Mart [24] and CloudStone [23]. However, there is currently no framework that specifically supports experimentation, validation and design for robustness of the proposed CACTOS combined optimizations.

VIII. Conclusions

This paper presents the CACTOS approach to addressing heterogeneity in data center scale, complexity, and workloads through a combination of in-depth analysis of application behavior and insights from commercial cloud providers. Key to the approach is predictive modeling and simulation of application resource requirements in conjunction with context-aware optimization methods. We show how the CACTOS building blocks CactoScale, CactoOpt, and CactoSim make up a holistic toolkit that can be used in data center operations and present our validation plans for envisioned results.

Acknowledgements

The authors extend their gratitude to Dr. Laura Moore of SAP Global Research and Business Incubation Belfast, Belfast, Northern Ireland for work related to this research. This work is funded by the European Union's Seventh Framework Programme under grant agreement 610711.

References

[1] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing. How is the weather tomorrow?: towards a benchmark for the cloud. In *Proceedings of the Second International Workshop on Testing Database Systems*, page 9. ACM, 2009.

[2] T. Binz, G. Breiter, F. Leyman, and T. Spatzier. Portable cloud services using tosca. *IEEE Internet Computing*, 16(3), 2012.

[3] J. Boulon, A. Konwinski, R. Qi, A. Rabkin, E. Yang, and M. Yang. Chukwa, a large-scale monitoring system. In *Proceedings of CCA*, volume 8, pages 1–5, 2008.

[4] R. Buyya, R. Ranjan, and R. N. Calheiros. Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities. In *HPCS*, pages 1–11. IEEE, 2009.

[5] M. Chen, H. Zhang, Y.-Y. Su, X. Wang, G. Jiang, and K. Yoshihira. Effective vm sizing in virtualized data centers. In *IM*, pages 594–601. IEEE, 2011.

[6] J. Dejun, G. Pierre, and C.-H. Chi. Resource provisioning of web applications in heterogeneous clouds. In *WebApps*, pages 5–5. USENIX Association, 2011.

[7] S. Di12 and F. Cappello. Gloudsim: Google trace based cloud simulator with virtual machines. Technical Report ANL/MCS-P5017-0913, Argonne National Laboratory, 2013.

[8] D. G. Feitelson. *Workload modeling for computer systems performance evaluation*. Cambridge University Press, 2014.

[9] C. W. Fong, H. Asmuni, B. McCollum, P. McMullan, and S. Omatu. A new hybrid imperialist swarm-based optimization algorithm for university timetabling problems. *Information Sciences*, 283:1–21, 2014.

[10] A. Ganapathi, Y. Chen, A. Fox, R. Katz, and D. Patterson. Statistics-driven workload modeling for the cloud. In *ICDEW*, pages 87–92. IEEE, 2010.

[11] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown. Elastictree: Saving energy in data center networks. In *NSDI*, volume 10, pages 249–264, 2010.

[12] S. Herrero-Lopez, J. R. Williams, and A. Sanchez. Large-scale simulator for global data infrastructure optimization. In *CLUSTER*, pages 54–64. IEEE, 2011.

[13] N. Huber, F. Brosig, and S. Kounev. Modeling dynamic virtualized resource landscapes. In *SIGSOFT QoSA*, pages 81–90. ACM, 2012.

[14] N. Huber, A. van Hoorn, A. Koziolek, F. Brosig, and S. Kounev. S/t/a: meta-modeling run-time adaptation in component-based system architectures. In *ICEBE*, pages 70–77. IEEE, 2012.

[15] B. Jennings and R. Stadler. Resource management in clouds: Survey and research challenges. *Journal of Network and Systems Management*, pages 1–53, 2014.

[16] J. W. Jiang, T. Lan, S. Ha, M. Chen, and M. Chiang. Joint vm placement and routing for data center traffic engineering. In *INFOCOM*, pages 2876–2880. IEEE, 2012.

[17] J. O. Kephart and D. M. Chess. The Vision of Autonomic Computing. *Computer*, 36:41–50, 2003.

[18] K. Kuchcinski. Constraints-driven scheduling and resource assignment. *ACM Trans. Des. Autom. Electron. Syst.*, 8(3):355–383, July 2003.

[19] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. In *SIGMETRICS PER*, number 1 in SIGMETRICS '12, pages 175–186. ACM, 2012.

[20] A. Ojala and P. Tyrvainen. Developing cloud business models: A case study on cloud gaming. *IEEE software*, 28(4), 2011.

[21] I. Rodero, H. Viswanathan, E. K. Lee, M. Gamell, D. Pompili, and M. Parashar. Energy-efficient thermal-aware autonomic management of virtualized hpc cloud infrastructure. *Journal of Grid Computing*, 10(3):447–473, 2012.

[22] B. Sharma, V. Chudnovsky, J. L. Hellerstein, R. Rifaat, and C. R. Das. Modeling and synthesizing task placement constraints in google compute clusters. In *SoCC*, page 3. ACM, 2011.

[23] W. Sobel, S. Subramanyam, A. Sucharitakul, J. Nguyen, H. Wong, A. Klepchukov, S. Patil, A. Fox, and D. Patterson. Cloudstone: Multi-platform, multi-language benchmark and measurement tools for web 2.0. In *CCA*, volume 8, 2008.

[24] A. Turner, A. Fox, J. Payne, and H. S. Kim. C-mart: Benchmarking the cloud. *TPDS*, 24(6):1256–1266, 2013.

[25] Y. Yao, L. Huang, A. Sharma, L. Golubchik, and M. Neely. Data centers power reduction: A two time scale approach for delay tolerant workloads. In *INFOCOM*, pages 1431–1439. IEEE, 2012.

[26] Q. Zhu and T. Tung. A performance interference model for managing consolidated workloads in qos-aware clouds. In *CLOUD*, pages 170–179. IEEE, 2012.