RZ 3858 (# ZUR1311-058) 12/04/2013
Computer Science 16 pages

# Research Report

## Cloud Modeling and Simulations

Peter Altevogt‡, Wolfgang Denzel*, Tibor Kiss#

‡IBM Germany Research & Development GmbH
Germany

*IBM Research – Zurich
8803 Rüschlikon
Switzerland

#Gamax Kft
Budapest



**Research**
**Africa • Almaden • Austin • Australia • Brazil • China • Haifa • India • Ireland • Tokyo • Watson • Zurich**

# Cloud Modeling and Simulations

Peter Altevogt*, IBM Germany Research & Development GmbH, Germany,
peter.altevogt@de.ibm.com
Wolfgang Denzel, IBM Research GmbH, Zurich Research Laboratory, Switzerland,
wde@zurich.ibm.com
Tibor Kiss, Gamax Kft, Budapest, tibor.kiss@hu.ibm.com

***Keywords***— cloud; performance simulation; performance modeling; queuing modeling, discrete-event simulations; scalability

## SUMMARY

This chapter provides an introduction to cloud modeling and simulation technologies with focus on performance and scalability. We will emphasize the value of modeling and simulation technologies in the context of cloud computing and provide a short general introduction to these technologies to make the chapter more self-contained. Specific requirements against modeling and simulations applied to cloud computing are discussed motivating our simulation-based approach. We will outline in some detail how various hardware and software objects of clouds can be modeled as individual entities and then glued together to form a simulation model of a complete cloud. Besides addressing key concepts like modeling of contention and data segmentation, we will describe workload generation and also address some general challenges associated with cloud modeling and simulations. Finally, we will describe the various stages of a cloud modeling project using the simulation of OpenStack image deployment on clouds as an example.

## INTRODUCTION

Cloud computing is perceived as a game changing technology to provide respectively to consume data center resources [1]. Various non-functional attributes like performance and scalability, cost and energy efficiency as well as resiliency, high-availability and security are considered vital to enable further growth of cloud computing [2].

To ensure a cloud design addressing these attributes, an engineering approach limited only to measurements, tuning and fixing defects is not sufficient. In general these activities happen late in the development respectively deployment cycle. Therefore they

can not sufficiently address cloud design issues. Furthermore, only a very limited set of scenarios can be handled this way due to time and resource limitations[1].

Modeling and simulation technologies can help to alleviate these issues. They can be applied to model various cloud attributes like energy dissipation or to evaluate different pricing models. For definiteness, this chapter will focus on performance and scalability. The discrete-event simulation approach described here could be generalized to also address non-functional cloud features like energy efficiency or costs, but it needs to be augmented with other simulation techniques like thermal simulations for the sake of completeness. Such multi-domain simulations are beyond the scope of this chapter.

Performance modeling and simulation techniques enable a performance analysis of cloud designs and capacity planning early in the development cycle at large scale with moderate costs and respecting the strict time constraints of an industrial development project. Although being widely used and well established in various branches of information and telecommunication industries, the application of these techniques to cloud computing provides some new challenges due to complexity, diversity, agility and scale.

In the next sections we will first provide a short introduction to performance modeling and simulation technologies and provide some rationale to choose simulation technologies in the industrial context.

Then we will discuss specific requirements against modeling[2] clouds, followed by a survey of publications and tools currently available here.

The next section addresses the question, where in the development and deployment life-cycle of a cloud performance modeling and simulation methods can be applied.

We will then identify the various objects of a cloud that need to be modeled. This includes hardware and software components as well as workloads consisting of various types of requests. We will also propose appropriate levels of modeling abstractions for the various cloud objects. Furthermore, we will discuss the modeling of some key performance related concepts like contention and segmentation in the cloud.

Some challenges associated with a cloud simulation project are described next, especially the specification of objectives, parameterization, calibration, execution and result analysis.

Finally, a real-life cloud simulation project will be described including all project phases, execution characteristics and some typical results.

We will end this chapter with a conclusion.

[1] Although most development projects face these limitations, their impact on cloud computing is especially severe due to the huge resources and costs required to create and manage large-scale clouds.
[2] We will frequently use the terms „modeling“ and „simulating“ as synonyms throughout the chapter.

## MODELING AND SIMULATION OVERVIEW

To make this chapter more self-contained, we provide a short survey of the two major approaches used to model performance and scalability of hardware and software components, namely analytic queuing modeling and discrete-event simulation technologies.

Analytic queuing modeling uses mathematical models to describe the performance of systems consisting of hardware and software components as well as workload requests interacting with them [3][4][5][6][3]. As a simple example here we may think of requests (SQL queries) posted by clients and arriving at a database server. Both the arrival of requests and their handling at the server can be modeled as stochastic processes using probability distributions for their inter-arrival times respectively the service times. In some especially simple cases, these stochastic processes can then be solved analytically[4] to obtain explicit expressions for observables like average response times and queue lengths, see Fig. 1.

$$R = \frac{S}{1-\rho}$$
$$Q = \frac{\rho}{1-\rho}$$

Fig. 1. Typical results of simple analytic queuing modeling: the average response time *R* and the average number of requests as a function of the service time *S* and the utilization $\rho$.

Unfortunately, in the majority of real-life cases such explicit solutions are not available so that significant idealizations are required for the system components and workloads under consideration. In fact there are quite a few well-known and highly important performance relevant features in the context of clouds where analytic methods are difficult to apply [6], e.g.:

- Blocking:
  All hardware (active) resources like processor cycles and network bandwidth as well as all software (passive) resources like database connection pool sizes or number of operating system threads in a cloud are finite. When a request has to wait for such a resource due to contention, its execution blocks. This may impact the execution rate at other resources. E.g. a request waiting for a database connection currently in use by other requests cannot proceed using available network bandwidth[5] to access the database server.

[3] In fact analytic modeling can be considered as a branch of applied mathematics, namely probability and stochastic processes. It has a long tradition and was first applied by A. Erlang to model performance of telephone systems at the beginning of the last century [7].
[4] I.e. using a „paper-and-pencil“ approach and not relying on approximations or numerical methods.
[5] Therefore sizes of connection pools are always good candidates for performance tuning in clouds.

- Bulk arrivals:
  A single "launch instances" IaaS[6] request may contain many instances to be provisioned at the same instance of time in clouds.

- Forking / rejoining of requests:
  Basically all data transfer in clouds is done in form of blocks of packets requiring a segmentation (forking) of one large request into smaller ones; this request is completed when all of its child requests have completed (rejoined).

- Load balancing:
  Adaptive load balancing is heavily used within clouds to ensure an optimal utilization of resources.

- Mutual exclusion:
  Frequently data structures in clouds require an atomic access of requests for updating to ensure their consistency (e.g. network configuration data at hypervisors). This is frequently implemented by so-called "critical sections" of code protected by a lock which can only be owned by one request at each instance of time.

- Non-exponential service times:
  The usage of (negative) exponential probability distributions to model service times is key in analytic modeling, but in clouds service times will in general violate this assumption.

- Simultaneous resource possession:
  One of the key methods to enhance IO performance in clouds is the use of asynchronous IO resulting in requests owning several resources at the same instant of time, namely processor cycles and resources of the IO subsystem.

- Transient states:
  Significant workload fluctuations (e.g. depending on the time of the day) result in cloud states far away from equilibrium.

On the other hand, various approximation schemes are available here and analytical modeling is highly efficient in terms of resource consumption and execution time [3].

A simulation is an imitation of the operation of a real-world process or system over time implemented on a computer system [8][9]. In science and engineering simulation technologies have a long tradition and are extensively applied [10] for processes or systems that change their state continuously over time (see Fig. 2) and can be described by (systems of) differential equations. However, these methods are of limited value when applied to cloud computing, because of the inherently discrete, non-continuous behavior

[6] IaaS stands for Infrastructure as a Service; for more details see [1].

of cloud components and workloads[7]. This motivates the application of another simulation technology, namely discrete-event simulations (also known as event-driven simulations) which only model state changes at discrete times (see Fig. 2).

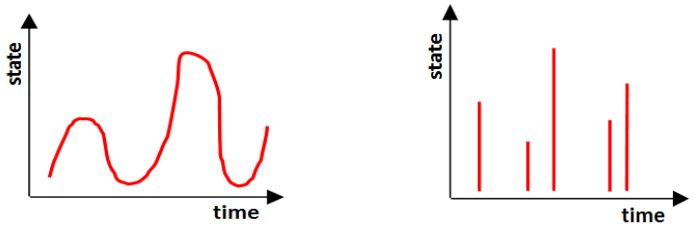


Fig. 2. The left figure shows a ystem with state changing continuously as a function of time. Such systems frequently occur in physics and engineering and are modeled by (systems of) differential equations. The right figure shows a system with state changing only at discrete times. Basically all systems used in information technology including clouds fall into this category. Differential equations are not well suited to model such systems because of their discrete nature.

Although a detailed discussion of discrete-event simulations [8][9] is beyond the scope of this chapter, we provide here a brief description. The key idea of discrete-event simulations is to model the system operation only by the sequence of events that cause changes in system state at discrete points in time. Typical events are, e.g., arrivals or departures of requests occurring at system components. Future events that are already scheduled at a particular point in time are kept in a future event list. As illustrated in the core discrete-event algorithm in Fig. 3, the simulator repeatedly picks from this event list the earliest next event, hands it over to the particular event handler that is associated with the event, and sets the simulation clock to the event's occurrence time. A particular event handler, e.g., might be responsible for performing the necessary state update for a specific system component at arrival of a specific request. Typically, a handler also schedules new upcoming events into the future event list which may be events modeling the next arrival of the same kind or a departing request targeted to another system component. This core process is repeated until the simulation ending condition is reached. This way, the dynamic system behavior is correctly modeled by immediate time jumps across the non-relevant time periods between subsequent events. These time jumps make discrete-event simulation superior in efficiency to the also known activity-based simulation method (also known as time-driven simulations) where time is incremented in sufficiently small equidistant steps thereby wasting computing time during non-relevant time periods.

In cloud simulations, the time between two events is typically several seconds, while the average time to handle one event is approximately a millisecond on contemporary computer systems. Leveraging parallel simulation technologies to handle many events concurrently, this enables the accurate simulation of large clouds for long time intervals.

Furthermore, the remarkable simplicity of the core discrete-event algorithm results in a high flexibility and enables a straightforward simulation of the results of analytic queuing modeling and especially also of the items mentioned in the previous section. Applying

[7] E.g. on processor level request execution is driven by the discrete processor clock and on cloud level by events like arrival and departure of launch instances IaaS requests at compute nodes.

discrete-event simulations to cloud computing, it seems feasible to simulate almost all performance relevant features of clouds with accuracies beyond what can be measured with the currently available cloud measurement tools.

Still, significant efforts are required to implement meaningful cloud simulation models and their execution may require significant resources and execution time.

To sum up, for cloud modeling discrete-event simulations seem to be superior in terms of accuracy and flexibility. This is essential in an industrial context. The greater efficiency of analytic methods can be offset (at least partially) by leveraging parallel simulation technologies on appropriate contemporary computer systems. Therefore we have selected discrete-event simulations as our primary tool to model clouds with the option to use simulation models as a basis for analytic queuing modeling efforts.

## CLOUD MODELING AND SIMULATION REQUIREMENTS

Although performance modeling and simulation technologies as introduced in the previous section are widely used and well established in various branches of information and telecommunication industries [11][12], their application to clouds provides some new challenges due to complexity, diversity, agility and scale [13].

- The hardware infrastructure of clouds consists of servers, networking switches and storage subsystems and all of these components need to be taken into account on an equal footing. This is in contrast to most of the performance simulation work focusing on (parts of) just one of these components.
- A cloud being a complex system with intricate interactions between hardware and software modules, we need to treat both software workflows and hardware infrastructure as first class citizens when simulating end-to-end performance.
- In general the software heuristics for managing and using a cloud change at a much higher rate than the available cloud hardware infrastructures; therefore it is important to introduce separate modules for simulating software heuristics and the hardware infrastructure to support a rapid implementation of new cloud software heuristics for an unchanged hardware infrastructure and vice versa.
- The market for cloud solutions being highly dynamic, simulations of new clouds must be provided in a timely manner, i.e. we need to support a rapid prototyping.
- We need to allow for selectively and rapidly adding details to the simulation of specific hardware or software components to increase the credibility of the simulation effort if required by the stakeholders of a simulation effort.

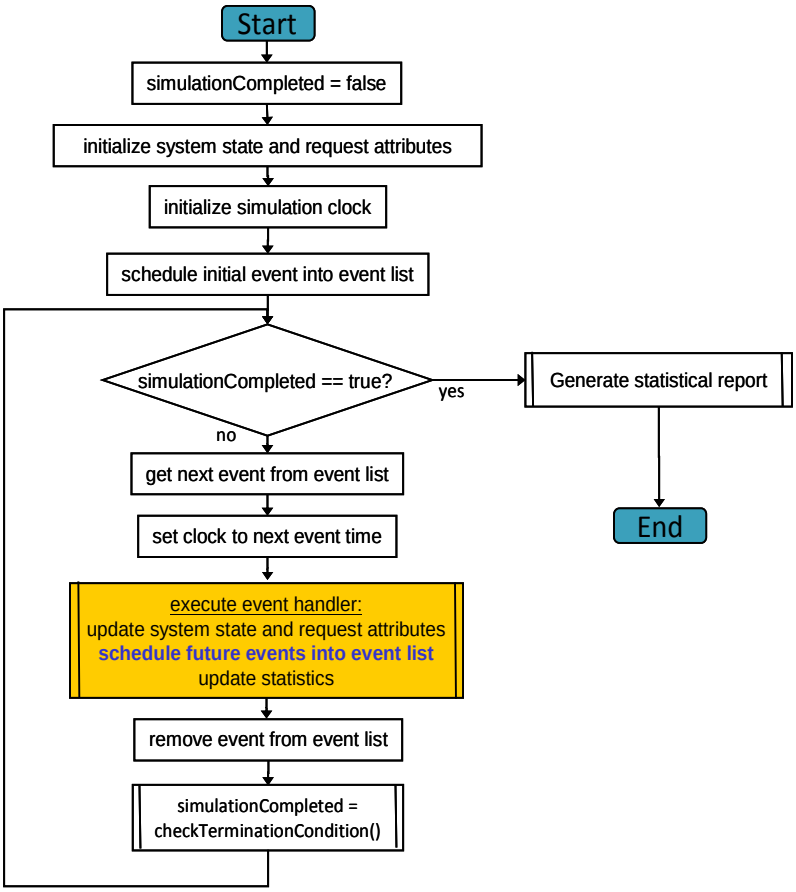


Fig. 3. The core algorithm of a discrete-event simulation engine.

- Last but not least the enormous size of contemporary cloud data centers requires highly scalable simulation technologies.

Besides addressing these challenges, a key point is to find a useful abstraction level for modeling the system and workload under investigation. This is largely determined by the goals of the modeling effort, but must ensure modeling of typical cloud level objects like compute nodes, switches, routers, hypervisors, virtual machines, load balancers, firewalls with an appropriate level of detail[8].

In fact there are quite a few simulation frameworks for clouds available, see [13] - [26]. For definiteness, we will describe the approach used in [13] and [14] to address the challenges mentioned above in more detail, but most of this chapter is independent of the concrete simulation framework under consideration.

[8] I.e. cloud simulations can not simply reuse the simulation approaches applied to model processor cores or crossbar switches to support hardware design.

## MODELING AND SIMULATION OF CLOUDS

Although clouds are in general highly complex systems, they consist of a rather small set of fundamentally different hardware and software building blocks, see Fig. 4. The key modeling challenges here are to

- identify the appropriate building blocks
- model these building blocks on an appropriate abstraction level
- enable their modular and scalable combination to build complex clouds
- support implementation of request workflows on various levels of detail

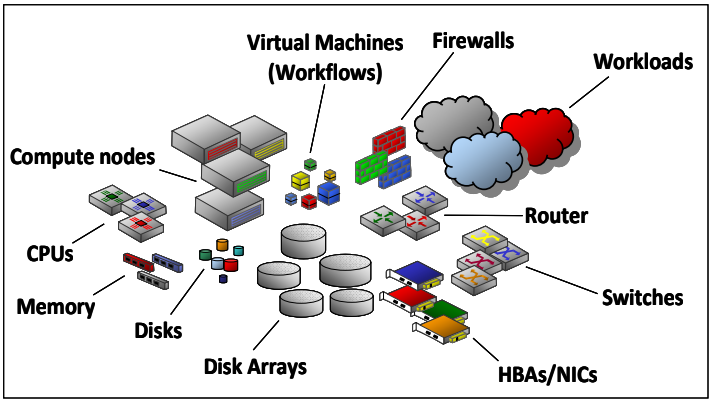


Fig. 4. Samples of modeled cloud hardware and software components

We distinguish between active resources representing hardware resources like processor cores, disks and switches where requests spend time for execution and passive resources representing software artifacts like database connections or thread pools. Requests do not spend time for execution at these passive resources, but frequently need to own one or many of them to access an active resource.

As a typical example for a model of an active (hardware) resource, we take a closer look at the model of a network device, see Fig. 5. It consists of basic building blocks ("Lego bricks") providing bandwidth to model the ports and a crossbar switch, another "Lego brick" providing processor cycles and a module implementing the software workflow details including the creation of routing tables at initialization of the simulation. Using different parameterizations of the components, this simulation module can be used to model a wide variety of network devices like switches, router and firewalls. A request enters the network device at a port, traverses the crossbar, spends some time at a processor core, looks up the appropriate port connected to the next target in the routing table of the workflow module and finally leaves the device via this port[9].

[9] This low-level request workflow within a network device is in general initiated be a high-level workflow request implementing e.g. a cloud level software application.

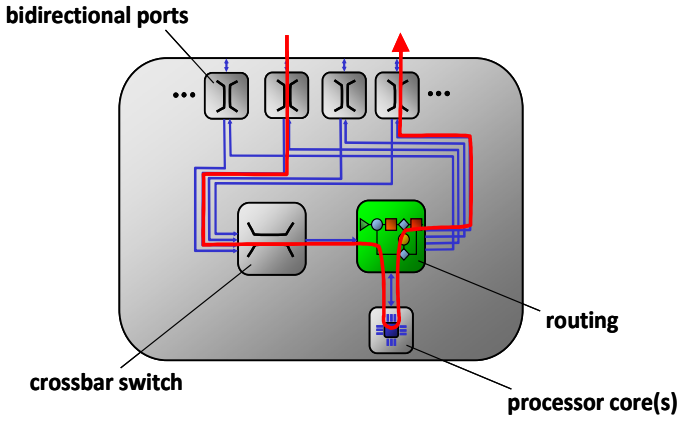


Fig. 5. A general network device The red line indicates the flow of a request.

A typical scenario for the application of passive resources is the simulation of so called critical regions, see Fig. 6. Such regions allow only one concurrent request to proceed and are used e.g. to ensure consistency of data updates. In this case a request has to queue for a token provided by a token pool to access the critical region and spend some time for execution within it. After leaving the critical region, the request releases the token again and other requests waiting may try to obtain it according to a specified arbitration policy.

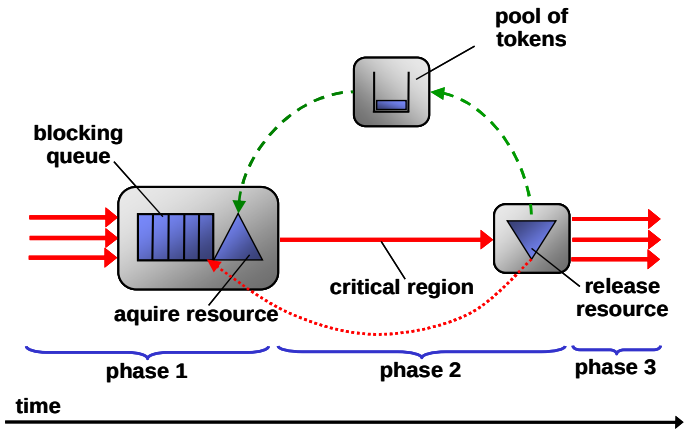


Fig. 6. Modeling a critical region allowing only one request (thread) concurrently in flight

Models of large-scale clouds are created by combining more basic modules, e.g. processor cores, switches and disks to create a compute node, compute nodes and switches to create a rack and several interconnected racks and storage units to create a data center[10]. Finally, these can be combined to build a cloud consisting of a number of world-wide distributed data centers, see Fig. 7. The key design concepts of a simulation framework supporting this approach are modularity and the ability to replicate ("copy-and-paste") objects at any level of complexity.

[10] It is important to note, that modules may be replaced by more fine-grain of coarse-grain models at any level of this building-block approach.

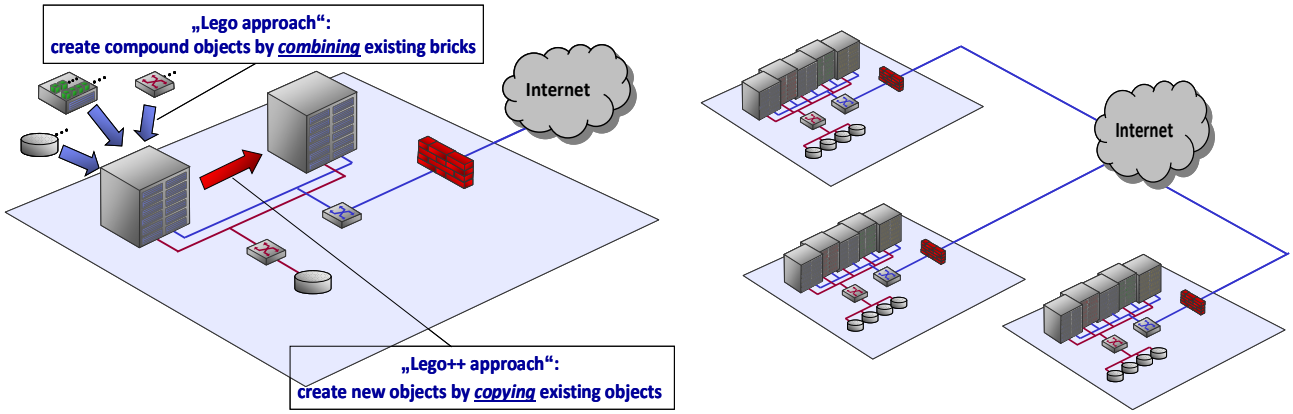


Fig. 7. Complex compound modules can be created by combining more basic modules, e.g. a server rack by combining compute nodes with various VMs, disk arrays and network components. Using these racks, a data center can then be created by a copy-and-paste approach. Finally, a cloud consisting of a number of world-wide distributed data centers can also be created by applying the copy-and-paste approach, this time applied on data center level.

For a simple taxonomy of cloud requests, see Fig. 8. Furthermore, we can associate various levels with requests. Requests at a higher level are then initiating request workflows at a lower level. The highest level request workflows are in general associated with applications at cloud level and implemented in the context of virtual machines (VMs), the lowest level request workflows with accessing hardware components like disk media. This allows the implementation of new cloud level application workflows without the need to take details of low-level device related workflows into account and vice-versa.

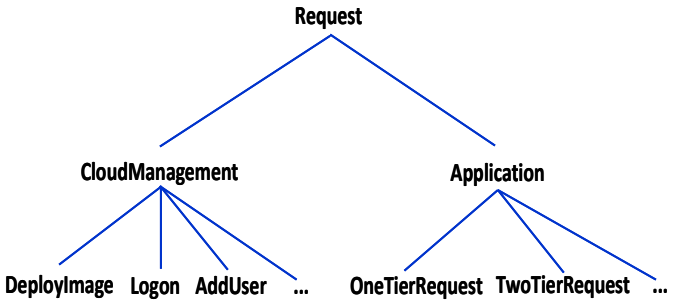


Fig. 8. Request taxonomy with the request of type “Request” at its root

In the workload generator module (the internet in the context of clouds, see Fig. 7), all functionality related to generating, initializing and posting requests of various types against the cloud is implemented. Furthermore, it may also be used to collect all request related statistics[11].

[11] Device related statistics like utilization and queue lengths are in general collected at the device simulation modules.

We will now focus on some key cloud modeling aspects that are quite independent of the concrete resources and requests under consideration. Because all cloud resources are limited and demand may well be bigger than resources available in the cloud, arbitration (respectively scheduling) of resources to requests is essential. A simple FCFS arbitration is frequently used in [14], but it is straightforward to implement more complex schemes. Another key feature when modeling clouds is segmentation and reassembly of requests, especially of requests modeling the transfer of data over the network. Such segmentation allows a request to be in flight concurrently at several devices resulting in an increased throughput. Therefore it is essential to model segmentation even if some compromises concerning granularity have to be made[12] given the finite time and resources provided to execute simulations.

## CLOUD MODELING AND SIMULATION CHALLENGES

Modeling and simulation of cloud performance is associated with some specific challenges caused by their highly volatile nature and scale.

- Architectural and workload details of clouds are changing rapidly. This is caused e.g. by new processes of delivering software like continuous delivery [27]. Besides requiring fast updates of the simulation models themselves, this makes their precise parameterization difficult. The problem can be alleviated by focusing on the prediction of *relative* values of performance metrics against a known baseline, e.g. modeling the *increase* of request throughput when updating the cloud.

- For calibrating models, high quality measurement data with a detailed specification of workloads and cloud infrastructure components are of great value. Unfortunately, such data is rare[13] and this is significantly impacting model accuracy.

- Due to constraints in time and available resources, measurement data is in general available only for small or medium sized clouds. Using this data for parameterization and calibration, there is a considerable risk to overlook resources of importance for clouds at large scale. Not taking these resources into account will most likely result in significant modeling errors.

- To enable non-expert users to execute cloud simulations, a limited set of modeling scenarios should be made available as a service accessible via a web user interface. The challenge here is to identify the most useful scenarios and a small set of associated key parameters to be exposed to a non-expert user.

[12] E.g. it is not feasible to model a MTU size of 1500 bytes in large cloud networks due to the excessive amount of events associated with processing such fine granular data segmentation.
[13] This is especially the case for end-to-end measurements on large scale clouds due to constraints in time and resources.

## SIMULATION PROJECT CASE STUDY: OPENSTACK IMAGE DEPLOYMENT

We will use the simulation of image deployment in OpenStack [28] managed clouds as an example of a concrete cloud simulation project and describe its phases below.

1. **Specification of objectives**
   In the ideal case, the objectives of a simulation project are specified and agreed on with the stakeholders at the beginning of a simulation project. In our case, objective is to study the impact of concurrency on throughput and response time of image deployment requests for various cloud architectures. Furthermore, we want to learn about various device utilizations and identify bottlenecks.

2. **Design and Specification**
   In this phase, the appropriate cloud architectures to be simulated are specified as well as key workload characteristics. In our case e.g. network topologies, levels of concurrency, images to be deployed and overall cloud sizes.

3. **Implementation**
   The specified OpenStack image deployment workflow [29] is implemented for various cloud architectures, e.g. for the one outlined in Fig. 9.

4. **Parameterization**
   Key workload and infrastructure related parameters like service times and resource consumptions have to be extracted from available measurements and documentations. When no data is available, reasonable values based on past experience have to be used. Various tradeoffs have to be made (e.g. concerning size of data packets) to balance between execution time and simulation accuracy.

5. **Calibration**
   Available measurements for concurrent image deployments are used for calibration. To factor out various unknown details of this data, we use relative numbers comparing performance metrics for concurrent image deployments versus single image deployment numbers for each scenario, see Fig. 10.

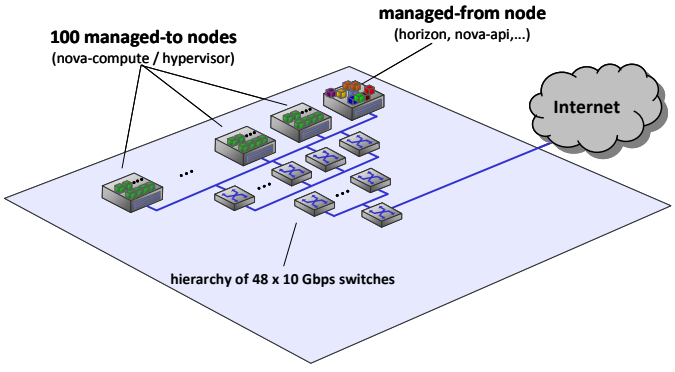


Fig. 9. An OpenStack managed cloud with all OpenStack management components on one compute node, but with separate nodes for the managed-to system

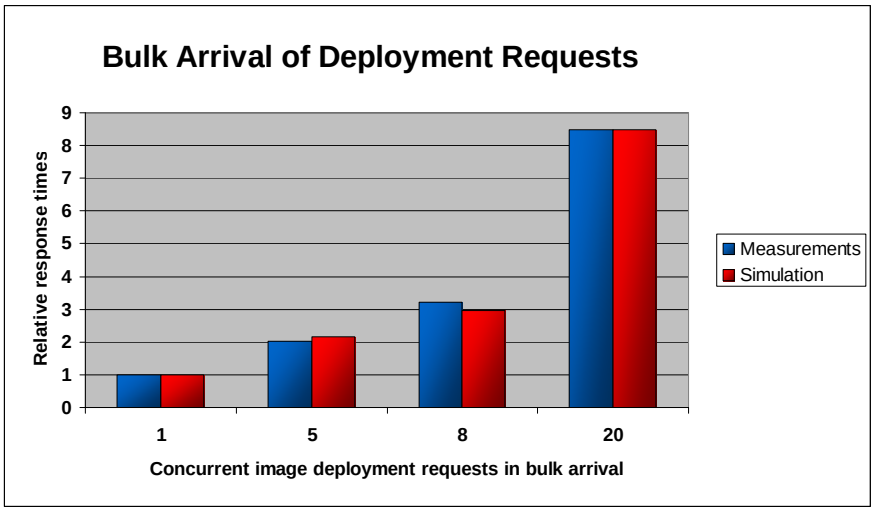


Fig. 10. A number of scenarios with a bulk arrival of image deployment requests is used for calibration of the simulation. Note that we use relative response times versus the response time for one request here.

## 6. Execution

The key performance metric characterizing the execution of discrete-event simulations is the event throughput. We observe a throughput of approximately 1.2 million events per second on one contemporary compute node using the OMNEST simulation software. This number as well as the size of the cloud[14] that can be simulated is largely determined by the type and amount of available memory[15] [16]. The execution time depends critically on the required accuracy: to ensure a deviation of less than a few percent for most simulation results, the number of executed requests should be at least two orders of magnitude greater than the number of concurrent requests in the cloud.

## 7. Result Analysis and Visualization

Visualization seems to be mandatory to convey simulation results. For some simple examples here see figures 11 – 15. Figures 11 and 12 show image deployment throughput respectively response times demonstrating saturation at approximately 4 (single compute node) respectively 28 (many compute nodes) concurrent requests. Figure 13 shows the queue length for locks at the hypervisor, e.g. for a software resource. Figure 14 shows an increased utilization of an external disk interface at the OpenStack image repository for many compute nodes. This indicates that in the case of distributing image deployment requests to many compute nodes, the bottleneck moves from a software to a hardware resource.

[14] We think of the size of a cloud in terms of its infrastructure and the concurrent number of requests in flight.

[15] At the current level of detail, a single compute node with 96 GB main memory is sufficient to simulate various image deployment scenarios on cloud data centers with approximately 1000 compute nodes and associated storage and network devices. Larger clouds would require either a higher level of abstraction or parallel simulations on a cluster.

[16] This is because discrete-event simulations need to update data structures representing system state with high frequency resulting in a high rate of IO operations.

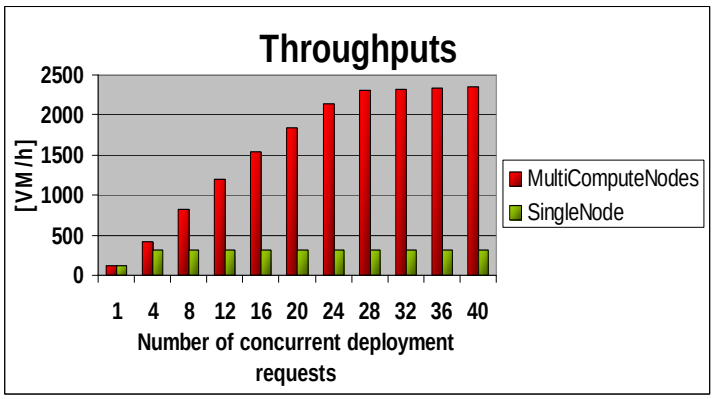


Fig. 11. Image deployment throughputs for various level of concurrency for a cloud architecture with a single compute node only and with many compute nodes.

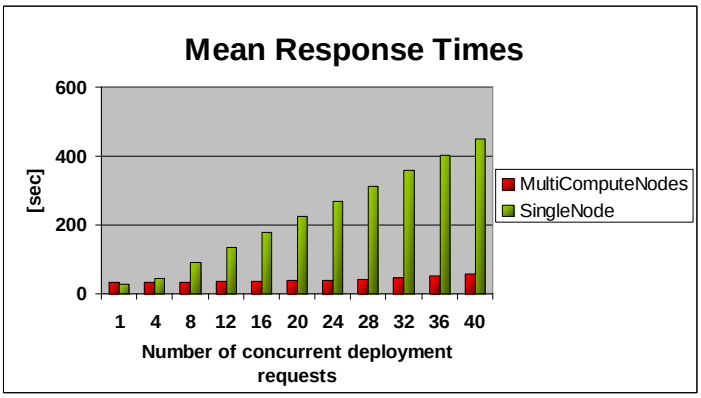


Fig. 12. Image deployment response times for various level of concurrency for a cloud architecture with a single compute node only and with many compute nodes.

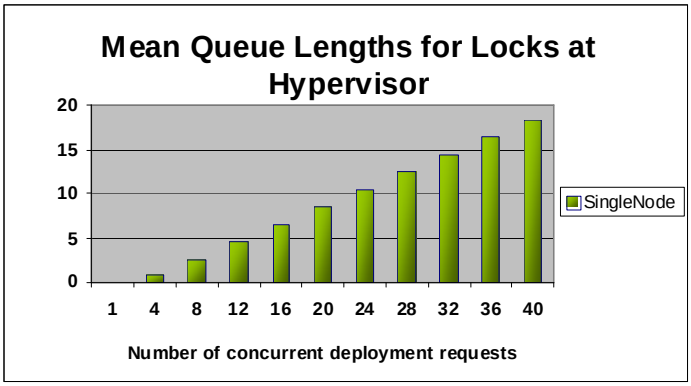


Fig. 13. Queue length for image deployment for various level of concurrency for the single node cloud architecture. No significant queueing occurs here in the node architecture with many compute nodes.

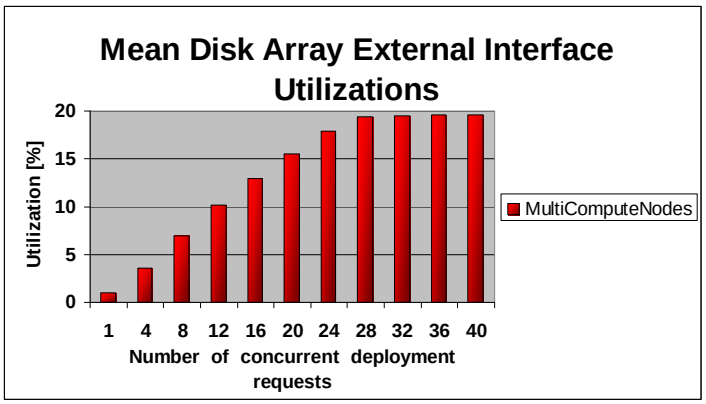


Fig. 14. Utilization of the external interface at the OpenStack image repository indicating an icreased contention for bandwidth here. In the single node case the utilization here is negligible. Comparing this with the results in figure 13 shows, how the bottleneck moves from software to hardware in case of multiple compute nodes.

## CONCLUSION

Cloud modeling and simulations can be of great value to support the design of workload optimized clouds, especially because of the prohibitive costs and time associated with the creation of large-scale test clouds. The challenges here are to address cloud agility and scalability, treat all hardware and software components of a cloud as first class citizens and make cloud modeling and simulation technologies easily accessible for non-experts. This requires quite a different modeling approach than in traditional simulation domains like microprocessor design or networking. Progress has been made, but more research and innovations are required to enable a more wide-spread use of these highly valuable technologies.

## REFERENCES


[1] Cloud Computing, http://en.wikipedia.org/wiki/c, http://en.wikipedia.org/wiki/Cloud_computing loud_computing.

[2] Above the Clouds: A Berkeley View of Cloud Computing, http://www.eecs.berkeley.edu/Pubs/TechRpts/ 2009/EECS-2009-28.pdf .

[3] G. Bloch, S. Greiner, H. de Meer, K.S. Trivedi. Queueing Networks and Markov Chains, Second Edition. Wiley-Interscience, 2006.

[4] L. Kleinrock. Queueing Systems, Volume 1: Theory. John Wiley, 1975.

[5] L. Kleinrock. Queueing Systems, Volume 2: Computer Applications. John Wiley, 1976.

[6] N.J. Gunther. Analyzing Computer System Performance with Perl::PDQ, 2nd edition. Springer Verlag, 2011.

[7] A. Erlang. “Solution of some problems in the theory of probabilities and telephone conversations.” Nyt Tidsskrift for Matematik B, 20:33-40, 1917.

[8] A.M. Law, W.D. Kelton. Simulation Modeling and Analysis, Third Edition. McGraw-Hill, 2000.

[9] J. Banks, J.S. Carson II, B.L. Nelson, D.M. Nicol. Discrete-Event System Simulation, Fourth Edition. Prentice Hall, 2005.

[10] M.O. Steinhauser. Computer Simulation in Physics and Engineering. Walter de Gruyter 2013.

[11] S. Pasricha, N. Dutt. On-Chip Communication Architectures, Elsevier Inc. (2008).

[12] D. Bertsekas, R. Gallager. Data Networks, Second Edition. Pearson Education (1992).

[13] P. Altevogt, W. Denzel, T. Kiss. Proc. of the 2011 Winter Simulation Conference (WSC'11). S. Jain, R.R. Creasey, J. Himmelspach, K.P. White, and M. Fu, eds.

[14] P.Altevogt, W.Denzel, T.Kiss. The IBM Performance Simulation Framework for Cloud. IBM Research Technical Paper http://domino.research.ibm.com/library/cyberdig.nsf/papers/041264DDC6C63D8485257BC800504EA2

[15] Wei Z., Yong P., Feng X., Zhonghua D., “Modeling and Simulation of Cloud Computing: A Review”, IEEE Asia Pacific Cloud Computing Congress (APCloudCC), 2012, pp.20-24.

[16] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and Experience, Vol.41, No.1, pp.23-50, 2011.

[17] Wickremasinghe, B., Calheiros, R.N. , Buyya, R., "CloudAnalyst: A CloudSim-Based Visual Modeller for Analysing Cloud Computing Environments and Applications", IEEE 24th International Conference on Advanced Information Networking and Application, 2010, pp. 446 – 452.

[18] S. K. Garg and R. Buyya, "NetworkCloudSim: modeling parallel applications in cloud simulations," 4th IEEE International Conference on Utility and Cloud Computing, pp.105-113, 2011.

[19] R. N. Calheiros, M .A. S. Netto, C. A. F. De Rose, and R. Buyya, "EMUSIM: an integrated emulation and simulation environment for modeling, evaluation, and validation of performance of cloud computing applications," Software-Practice and Experience, 00: 1-18, 2012.

[20] S. Ostermann, K. Plankensteiner, R. Prodan, and T. Fahringer, "GroudSim: an event-based simulation framework for computational grids and clouds," CoreGRID/ERCIM Workshop on Grids and Clouds. Springer Computer Science Editorial, Ischia, 2010.

[21] A. Nunez, J. L. Vazquez-Poletti, A. C. Caminero, G. G. Castane et al., "iCanCloud: a flexible and scalable cloud infrastructure simulator," Journal of Grid Computing, Vol.10, No.1, pp.185-209, 2012.

[22] M. Tighe, G. Keller, M. Bauer, and H. Lutfiyya, "DCSim: a data centre simulation tool for evaluating dynamic virtualized resource management," The 6th International DMTF Academic Alliance Workshop on Systems and Virtualization Management: Standard and the Cloud, 2012.

[23] D. Kliazovich, P. Bouvry, and S. U. Khan, "GreenCloud: a packet-level simulator of energy-aware cloud computing cata centers," Journal of Supercomputing, special issue on Green Networks, 2011.

[24] I. Sriram, "SPECI, a Simulation Tool Exploring Cloud-Scale Data Centers," CloudCom'09, LNCS 5931, pp.381-392, 2009.

[25] W. Zhang, X. Huang, N. Chen, W. Wang, H. Zhong, "PaaS-Oriented Performance Modeling for Cloud Computing", IEEE 36th Annua Computer Software and Applications Conference (COMPSAC), 2012.

[26] H. Khazaei, J. Misic, V. B. Misic, "A Fine-Grained Performance Model of Cloud Computing Centers", IEEE Transactions on Parallel and Distributed Systems , Vol. X, No. Y, 201Z.

[27] P. Swartout. Continous Delivery and DevOps: A Quickstart Guide. Packt Publishing 2012.

[28] OpenStack, http://www.openstack.org.

[29] OpenStack Request Flow, Mirantis Inc 2012, http://www.slideshare.net/mirantis/openstack-cloud-request-flow

[30] OMNEST Simulation Software, http://www.omnest.com/