Copyright Notice

Institute of Communication Networks and Computer Engineering
University of Stuttgart
Pfaffenwaldring 47, D-70569 Stuttgart, Germany
Phone: ++49-711-685-68026, Fax: ++49-711-685-67983
Email: mail@ikr.uni-stuttgart.de, http://www.ikr.uni-stuttgart.de

Modeling and Analysis of Virtualized Multi-Service Cloud Data Centers with Automatic Server Consolidation and Prescribed Service Level Agreements

Maggie Mashaly
Information Engineering & Technology Department
German University in Cairo
Cairo, Egypt
maggie.ezzat@guc.edu.eg

Paul J. Kuehn
Institute of Communication Networks and Computer Engineering
University of Stuttgart
Stuttgart, Germany
paul.j.kuehn@ikr.uni-stuttgart.de

***Abstract*—Cloud Data Centers (CDC) are developing rapidly and will have a major impact on IT infrastructures in the future for reasons of their low ramp-up costs and service delivery/support capabilities for the users. In this paper CDCs with multi-service application classes are considered which are operated under an automatic server consolidation based on parallel hysteresis methods for server activations/deactivations which have been reported on our previous work. Each class is subjected to an individual SLA, e.g., for the average service delay for non-real-time services or for delay percentiles for services with strict response time constraints, and probabilities for service rejection (loss) or migration. The CDC is modeled by a multi-class server cluster (SC) system, each of them represented by a multi-server queuing system which is controlled by a Finite State machine (FSM) for each class of cloud services. The SC systems are analyzed exactly under Markovian assumptions to receive averages and percentiles of response times and probabilities of loss or migration. The method is novel as it minimizes the energy consumption for servers by an automatic server consolidation strategy while guaranteeing the negotiated SLAs. The method is based on a worst case boundary consideration for the delays of arriving service requests and can be useful to understand the parametric influences and to assess the energy saving gains for multi-tier CDCs.**

***Keywords*—Modeling; Cloud Data Centers; Virtualization; Server Consolidation; Service Level Agreements; Performance Analysis; Multi-Objective Optimization**

I. INTRODUCTION

The availability of high-capacity Data Centers (DC) and the high-speed internet have a major effect on the IT infrastructures of enterprises as well as of cloud service providers. Major information-centric services are web-access, peer-to-peer file sharing, web-based business processes, storage and distribution of contents and multi-media communications. Data Centers are interconnected through the internet and form a "cloud" of resources delivering application services ("Software-as-a-Service", SaaS), providing an application interface ("Platform-as-a-Service", PaaS), or allowing for a user-configurable IT infrastructure ("Infrastructure-as-a-Service", IaaS). These possibilities are attractive due to the low ramp-up capital expenditure (Capex), low operational costs (Opex), and scalability of applications for the users. Virtualization concepts allow for a flexible and economic use of the DC equipment to meet the main aims of energy reduction ("Greening") and Service Level Agreements (SLA) between the user and the service provider. The analysis is based on modeling and mathematical performance evaluation as well as on computer simulations.

These developments are reflected by enormous research and development activities in the recent years addressing architectural, operational, resource management, energy-reduction, performance, experimental tests, and economic aspects, see [1-5], reflected by numerous conferences as, e.g., the conference series E²DC addressing energy-efficiency [6]. Studies on cloud DC performance are mostly based on experimental benchmarks or on simulations either by standard simulation tools or by specifically developed cloud simulation tool systems, see, e.g. [7,8]. Theoretical studies are based on modeling and performance evaluation using queuing theoretic approaches see [9-12]. An actual overview on these activities has been provided by a recent invited paper [13].

In this paper; Cloud DCs (CDC) are modeled by stochastic service systems (also known as queuing systems) which represent the main computational resources ("servers") and different service classes with class-specific SLAs for the evaluation of average or percentiles of response times, (delays between service request and service processing instants) as well as for blocking probabilities. The model is defined such that exact analytic performance evaluations are feasible under Markovian (i.e., memory-less) traffic assumptions, but can be extended to rather general cloud traffic models, scheduling strategies, and SLAs using stochastic event-by-event simulations. The specific and original features of the proposed method are the simultaneous aims of:

1. Regarding prescribed SLAs for each service class based on averages or percentiles of service response times as well as for service losses/migrations.

DOI 10.1109/LCNW.2016.16

2. Automatic power-saving server consolidation strategy based on parallel hystereses controlled by a Finite State Machine (FSM) whose concept was developed by the authors in previous contributions.

We will focus primarily on processing IT resources as the main energy-consuming devices. Memory and I/O resources are less power-intensive; their influence can in principle be considered by a compute time-proportional factor derived from experiments. Multi-core processors are common features of modern processor architectures allowing for parallel computation of process tasks. Under the assumption that multi-cores are used for parallel processing of individual jobs only, our modeling covers this case through a novel method of task graph reduction by which a multi-core can be modeled as a virtual single processor [14]. The current modeling approach allows for a rather quick estimation of the principal effects of energy-saving resource management by an automatic server consolidation under the objective of meeting prescribed SLAs.

In Section II the general model of a CDC is introduced. Section III addresses the parameterization of one Service Cluster (SC) of the CDC and its mathematical analysis shortly. In Section IV the multi-objective optimization of the model parameters is presented and applied to two case studies before the results are summarized with an outlook on work-in-progress in the concluding Section V.

II. Modeling of Multi-Class CLoud data centers

Modeling aims at an abstraction of a physical system, its functional operations, and its workload, and is, thus, a valid representation of that part of the real world under study.

Figures 1 and 2 represent our approach to model the main structural and functional properties of a CDC and a SC, respectively. In Figure 1 the principal structure of a multi-class CDC is sketched consisting of:

- The Hypervisor operating system, responsible for virtualization management providing an abstract view on the CDC as a cluster of server systems for processing of jobs ("Jobs" are also called Virtual Machines (VM) in connection with virtualized CDCs). Examples for Hypervisors are the Xen Hypervisor [15] or the VMware ESXi server [16].
- The Cluster Controller (CC), responsible for the mapping of the VMs to physical resources, i.e., servers and memory space, as well as for resource management functions as allocation of VMs to physical resources, load balancing, process migrations, power management, etc., under given load and SLA conditions. Typical Cluster Control equipment are the EMC[2] Distributed Resource Scheduler (DRS) and Distributed Power Management (DPM) [2].

The DC model consists of an arbitrary number *N* of SCs represented as queuing models for specifically defined cloud service classes (groups), each represented by the number of servers n_i service request buffers s_i and a class-specific Finite State Machine FSM_i controller, $i = 0,1,2,\ldots,N$. Jobs are assigned to their service class by the CC. The algorithm for server consolidation is implemented by the FSM. Job arrivals are represented by a general stochastic arrival process G_{Ai} with arrival rate λ_i jobs/s; the job execution times follow a general stochastic service process G_{Si} with service rate $\mu_i = 1/h_i$ where h_i is the mean service time. The FSM_i receives the actual state variables (x_i, z_i), where x_i denotes the number of currently activated process-executing servers and z_i denotes the currently buffered jobs to be executed by the scheduling controller CTL_i through commands c_i. Buffered jobs are served in strict order of arrival (First-In, First-Out; FIFO).

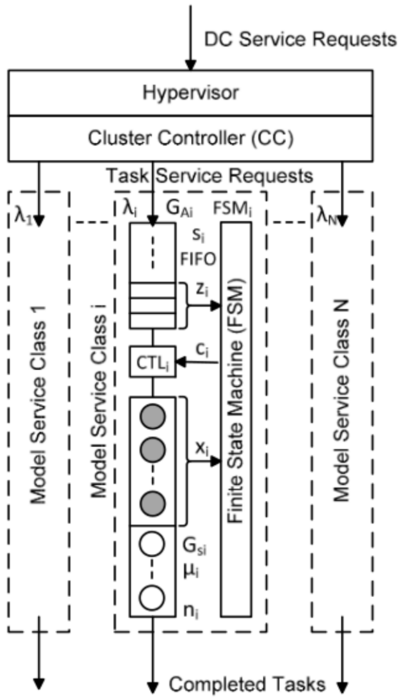

Figure 1. Multi-Class CDC Model

Figure 2 represents the function of the FSM_i expressed by a two-dimensional State Transition Diagram STD_i for the class i - system states (x_i, z_i). (Note: For reasons of presentation simplicity, the index i has been suppressed in Figure 2 and the remaining part of this section). This novel STD [17-19] has been constructed using multiple parallel hystereses of width z^{max} and steps z_j in order to reduce frequent server activations/deactivations by buffering service requests up to a threshold z^{max} before the next server becomes activated, while server deactivations take place only when no service request is waiting to become served, i.e., for $z = 0$. The bold-faced

transitions within the STD indicate server activations and deactivations, respectively.

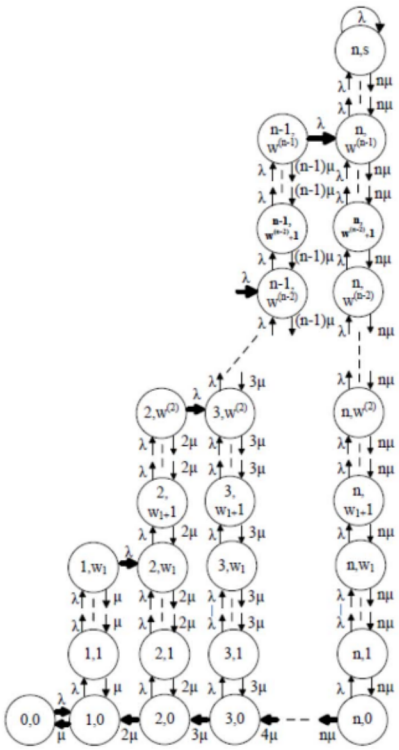

Figure 2. State Transition Diagram with Multiple Parallel Hystereses

The STD is flexibly adaptable to specific properties:

1. State-dependent multiple thresholds to avoid frequent oscillations between activations/ deactivations of resources to serve stochastically varying service requests and for an automatic self-adaptation to highly volatile load
2. Throttling of new server activations upon short bursts of arriving requests by buffering of these requests up to a scalable upper threshold
3. Serving of job requests with the maximum service rate of the activated servers to keep delays as small as possible.
4. For [illegible] and all states [illegible] net on arrival with [illegible], the average delay of an arriving job is bounded by

   [illegible] (1)

   in case of negative exponentially distributed service times.

Feature (4) is specifically remarkable with respect to service level agreements based on average delays of tasks. The worst-case average delay occurs when a new task arrives at a state [illegible], as all arriving jobs are served from the queue in strict order of arrival (FIFO), the new job has to wait for exactly [illegible] server terminations to start its service. In Section III we will extend this mean-value bound to a refined SLA, as well as the percentile of the response time which is particularly relevant for real-time-sensitive applications.

Notes:

- If the new job arrives at x = 0, its waiting time is 0; if it arrives at state [illegible], an immediate activation of an idle server occurs and the job waits on average no longer than [illegible] In other words; successively arriving jobs cannot be served prior to the considered job (because of FIFO), but may affect new server activations which increase the service rate and, therefore, decrease the average waiting time, respectively. Equation (1) can be extended with respect to generally distributed service times of type [illegible] (see Section III of this paper).
- The STD has been extended in order to include finite server activation overheads for server booting ("Cold stand-by" mode) or server sleep modes with lower power consumption and lower activation times ("Hot stand-by" mode) as well as to Dynamic Voltage and Frequency Scaling (DVFS) by state dependent service rates modeling the dynamically controlled server speeds [19]. In this paper this extension is not applied, which is in principle possible.

In this paper we will study the CDC under the assumption that the N service classes are served independently of each other; this means that the number of actually assigned servers [illegible] for class *i* tasks are predefined by the Cluster Controller and do not support each other in case of server bottlenecks, i.e., this is a solution with Static Load Balancing, i.e., the CC configures the assignment of servers to server clusters for a longer operating phase acc. to a management policy based on fixed SLAs for an economically planned target load level.

Alternatively, the CDC could also be organized by application of Dynamic Load Balancing methods, i.e., arriving jobs can be assigned individually to a specific server cluster either to avoid a job blocking ("loss") if the local buffer is already full, or based on the shortest response time at another server group, i.e., by co-operation between the server groups through job migrations to another (currently under loaded) server cluster either at the arrival instant prior to processing, or even during processing ("Life Migration"). This can be achieved through various co-operation principles:

(1) By mutual overflows (strategy "Local Server System First" (LSSF) through an overflow in case of buffer blocking), or

(2) On the basis of response times ("Shortest Response Time First", (SRTF)), where the job migration requires an additional overhead for buffer relocation, process transfer, possible delays through waiting phases and scheduling.

The two strategies LSSF and SRTF for Dynamic Load Balancing are analyzed exactly in a forthcoming paper [20].

III. Model Configuration and Analysis

In this section the model parameters will be configured first according to the required SLAs. After the model configuration, the model is analyzed exactly under Markovian traffic assumptions.

A. Model Configuration

As outlined before, applications are classified with respect to different service classes, such as individual tenants, user groups or services of individual types, traffic loads, or SLAs whose service demands are defined by means of the VM concept. For the execution, VMs are assigned to physical resources which execute the VM service tasks.

For class service the CC provides the main configuration parameters , xi =1, 2,…, . The performance of the model will be studied for class *i* for the traffic parameters and . The hysteresis width parameters have to be derived from the requirements. In this paper we will distinguish only between two main service characteristics:

Case 1: Non-real-time services (NRT)
Case 2: Real-time services (RT)

Case 1: Non-Real-Time Services (NRT)

For NRT services the SLA can be expressed by the average response (or waiting) time for a delayed task request. The worst-case mean delay is suffered by an arriving i-job meeting state (xi, 1) and when no further successive tasks arrive during its waiting time, which may cause another server activation and thus, a service speedup.

The mean delay is subjected under these assumptions to

(2)

from which follows

(3a)

(3b)

(3c)

Note that if the bound is integer valued, all hysteresis steps are identical for .

Case 2: Real-Time Services (RT)

For RT services much stronger conditions are required which can mathematically be expressed by response time percentiles derived from the complementary response time distribution function W (>t)/W (>0) of an arriving task which has to wait and the response time threshold

(4)

For negative exponentially distributed service times, the worst-case response (or waiting) time distribution function (DF) is of the type of an Erlang DF of order , namely when the state is met on the arrival of a new task. This leads to the condition

(5)

The hysteresis widths are found successively for x_i = 1, 2, …, by testing condition (5) for orders j=1,2,… as long as condition (5) is still fulfilled. The step sizes of the hysteresis width follow again from (3b).

Notes:

1. The method is based on the PASTA-Theorem ('Poisson Arrivals see Time Averages') and on Renewal Theory (see, e.g., [21]); it holds exactly only under Markovian conditions (exponential arrivals). As long as x_i servers are busy simultaneously, the worst-case response time delay follows as a sum of random phases:
   - 1st phase between arrival instant and the first server termination instant, which follows from Renewal Theory as the minimum of all residual service times of the currently ongoing services; for exponentially distributed service times this phase is exponentially distributed again.
   - phases between successive server termination instants which are all negative exponentially distributed.
2. For generally distributed service times (type G) these phases are not exactly known even for a queuing system of type M/G/n, but can be approximately derived by the method of stationary renewal intervals assuming independence between the server occupations, which has been used for the superposition of renewal point processes [22].

B. Model Analysis

Under the model assumptions the queuing models for each class of services can be analyzed separately for STDs of the type in Figure 2 under Markovian conditions. Contrary to theoretical transformation approaches [23, 24] he authors have developed an iterative recursion algorithm to solve for the stationary state probabilities exactly [19]. The novel algorithm allows the analysis of realistic cases of a large number of servers and has been published recently for DC models with one class of services and homogeneous servers [15]. The model was also extended to include server activation overhead, hot and cold stand-by and

Dynamic Voltage and Frequency Scaling (DVFS) [16], but will not be repeated here.

	Offered traffic
	Traffic load
	Average number of occupied servers
	Average queue length
	Activation /deactivation rate of servers
B_i	Probability of loss or migration for arrivals which don't meet the SLA
	Probability of delay
	Mean response time (delay)
	Compl. response time DF of delayed tasks
	Class specific and total power-saving efficiency, expressed by the fraction of power saved by consolidation referred to permanent power consumption

IV. DC Capacity Engineering and Case Studies

The principle of engineering the capacity of a DC Server Cluster for NRT and RT services and two case studies are considered to show numerically how the proposed method works and how it can be interpreted:

SC Capacity Engineering: Resource Sizing for NRT and RT
Case Study 1: Economy of Scale effect under given SLAs
Case Study 2: Load balancing between service classes

A. SC Capacity Engineering: Resource Sizing for NRT and RT

The SC performance behaves differently for the response time and for the loss of jobs dependent on the offered traffic load. If both criteria have to be met as SLAs simultaneously, that will turn the problem into a Multi-Objective Optimization Problem. In this paper we will solve this problem by fixing the hysteresis parameters at first according to delay criteria of the SLA. After having fixed these parameters, we will calculate the average response times and the loss probabilities B dependent on the traffic load .. from which conditions for the offered load A and for the bundle size n are derived. To meet a certain target value for the offered traffic A the number of servers can then be found.

Applying the analysis method for one SC according to the FSM - based operating algorithm [18-19] the principal results for the two SLA performance criteria for the average response time and for the loss probability B are illustrated in the Figures 3a, b for the case of NRT services for a given number of servers n dependent on the offered traffic load factor . In both Figures the SLA - prescribed upper performance bounds and as well as the decreasing behavior of delay and loss B for increasing n are also indicated. The intersections between the performance curves and the boundary values define the maximum allowable load factors and . We get the allowable load factor from

(6)

The mathematical problem of the resulting multi-objective optimization problem in order to find the required number of servers for a prescribed offered traffic and corresponding load factor can be found by an iterative algorithm as the functions for and B cannot be inverted with respect to .. mathematically. A given target triple of prescribed quantities can only be reached by adapting the number of servers n accordingly. The iterative algorithm starts with an optimistic number n = floor (A) and initial load factor thresholds . Then we increase n stepwise by 1 and determine the thresholds until .. crosses the first time the value of ..

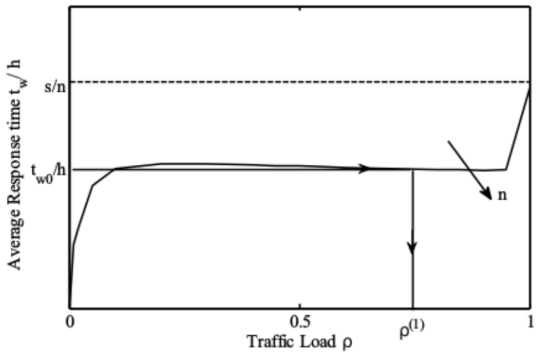

Figure 3a. Average Response Time vs. Traffic Load Factor ρ

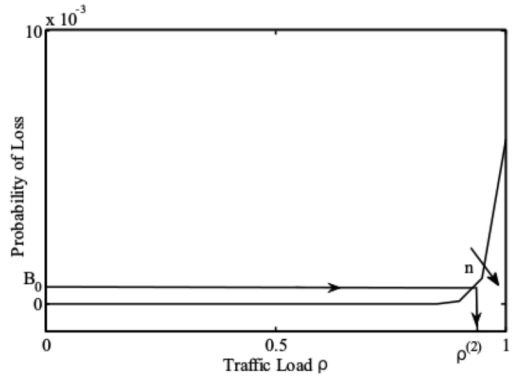

Figure 3b. Probability of Loss B vs. Traffic Load Factor ρ

Capacity Engineering for RT services follows in principle the same way when the upper average delay criteria $\bar{t}_{X,i}$ is complemented by the response time percentiles γ_i acc. to Eq. (5); this set of conditions satisfies the response time SLAs for an arriving job for each possible number k_i of occupied servers at their arrival instant by changed (i.e., smaller) hysteresis widths $\Delta_i^{(k)}$ and, thus, for all arrival cases. Then, we can proceed to optimize the number of servers *n* for the offered load ρ_i as in case of NRT services.

B. Case Study 1: Economy of Scale effects under given SLAs

The "Economy of Scale" (or bundling gain) is a well-known effect in teletraffic theory and describes the economic gain expressed by the increase of server utilization Y/n with increasing server group (bundle) size *n* for a given constant performance level. The most popular case is the classical loss system, expressed by the fraction of the carried traffic per server *Y/n* at a given probability of loss *B*, i.e. $Y/n = A(1-B)/n$ derived from the Erlang-B formula [21].

The present case is more sophisticated as the number of servers is automatically controlled by the server consolidation strategy of parallel hystereses and by consideration of the SLA parameters as bounds on the average values $\bar{t}_{X,i}$ or percentiles γ_i of the delay distribution

$$P(T_{D,i} > t_{D,i}) \le \gamma_i$$

for the delay threshold $t_{D,i}$, respectively, and the resulting probabilities of loss (or migration) B_i for all those arriving jobs which cannot be served under these SLA restrictions. The effect of loss/migration is a consequence out of the fact that only those arriving tasks will be served which will meet the SLA delay requirements.

To show the corresponding economy of scale effect, Figure 4a provides the server utilization Y_i/n_i as a function of the server group size n_i for a fixed average delay bound:

$$\bar{t}_{X,i} = \alpha\, h_i$$

for α=1 and different load parameters ρ_i. In Figure 4b the corresponding probabilities of loss (or migration) B_i are plotted vs. the server group size n_i.

The power saving efficiency η_i is defined as the fraction of saved power by server consolidation and the full power required without server consolidation ('always-on' case). It amounts to $\eta_i = 1 - Y_i/n_i$ and can directly be read-off from Figure 4a.

<u>Discussion:</u>

Figure 4a underlines the general wisdom of increasing server utilization gain Y_i/n_i with increasing bundle size n_i for a given average delay SLA. However, in contrast to the experience from pure loss systems, the full gain is already reached after a few servers, which allows for small group sizes in case of a high variety of SLA requirements. The probability of loss (or migration) decays with the bundle size, but increases, naturally, with the load. If a simultaneous SLA exists with respect to delay and loss, bundle size n_i which had been found through the algorithm above, can also be verified from Figure 4b to meet both requirements accordingly.

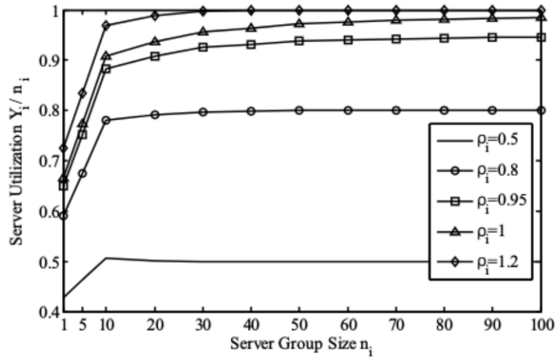

Figure 4a. Server Utilization Y_i/n_i vs. Server group size n_i for constant average delay bounds. Parameter: Offered traffic ρ_i

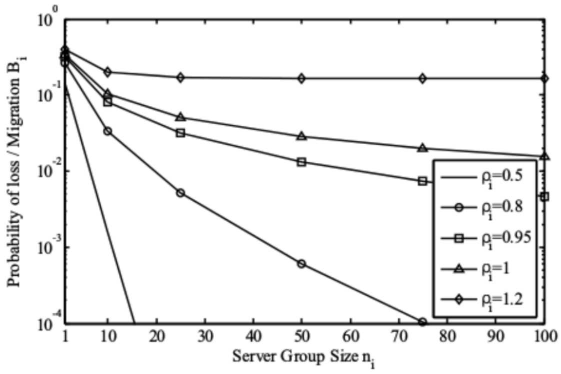

Figure 4b. Probability of Loss/Migration B_i vs. Server group size n_i for constant average delay bounds. Parameter: Offered traffic ρ_i

Figures 5a, b provide the corresponding results for the percentile bound of delays

$$P(T_{D,i} > t_{D,i}) \le \gamma_i$$

for

$$t_{D,i}/h_i = \beta = 5 \text{ and } \gamma_i = 0.01$$

The percentile SLA criterion results generally in smaller and heterogeneous hysteresis widths $\Delta_i^{(k)}$ which increase with the number of occupied servers k_i; the latter effect results out of the bundling gain which allows for an increasing buffering for increasing values of k_i and from the fact that with increasing queue lengths, the Erlangian tail components of the delay distribution function become less variant with increasing order.

If SLAs exist simultaneously for both delay and loss/migration, Figures 4b or 5b can be used to derive a lower bound for the

required number of servers n_i; the delay criterion is automatically met by the algorithm; to meet a prescribed loss/migration level B_i, the lower bound n_i can also be read off from Figures 4b or 5b, respectively.

C. Case Study 2: Load Balancing between Service Classes

The initial assignment of processing resources by the CC is based on fuzzy estimates which can change during execution and should be subjected to revision by the DC runtime management. Re-assignment decisions can be of either type:

1. to re-assign the number of servers to service classes, or
2. to migrate jobs of tenants between classes or CDCs.

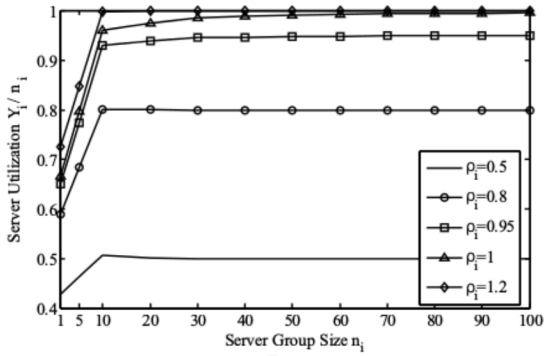

Figure 5a. Server Utilization Y_i/n_i vs. Server group size n_i for delay percentile bounds. Parameter: Offered traffic Y_i

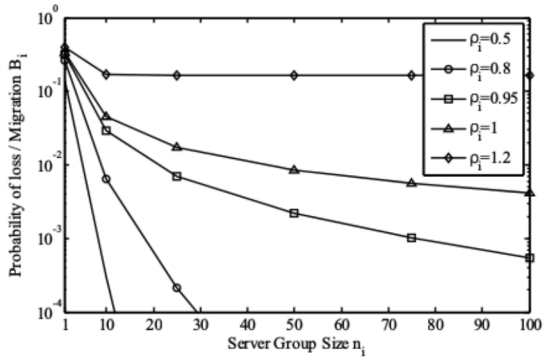

Figure 5b. Probability of Loss/Migration Y_i vs. Server group size n_i for delay percentile bounds. Parameter: Offered traffic Y_i

In the first case idling servers are blocked and can be re-scheduled after their assignment to the new service class. In the second case waiting jobs (including their memory assignments) are moved to another SC or even DC, to become scheduled if there is capacity available (job migration). Load Balancing was not the main topic of this paper and will be addressed in a forthcoming paper [20]. Here, we will just use this case study to demonstrate what can be achieved by dynamic load balancing.

For this example two service clusters1 and 2 are considered whose fixed (static) server assignment was $n_1 = n_2 = 10$ servers. Service cluster 1 is subjected to a current overload of ρ_1 = 0.95 while service cluster 2 runs at load ρ_2 = 0.8 and is able to accept additional load through job migration from cluster 1. As the percentile delay SLA is still maintained through our dimensioning of the hysteresis for automatic server consolidation, the difference lies in the SLA violation for the loss probability in class 1: As can be read-off from Figure 5b the current loss probabilities for classes 1 and 2 are about 0.2 and 0.007, respectively. Re-distributing the load to a balanced value of ρ_i = 0.875 for both clusters shows that the SLA of $B_i \leq 0.05$ can be reached for both.

V. Conclusion and Outlook

Based on previous work by the authors on modeling of hystereses-based server consolidation algorithms for automatic activation/deactivation of DC servers, a methodology has been developed in this paper for the performance evaluation of multi-tier CDCs with an arbitrary number of different service clusters (classes) and cluster-specific SLA values for averages or percentiles of response times and bounded quantities for service blocking (loss) or job migrations to other server clusters within or between CDCs. The theoretical analysis is based on Markovian traffic assumptions under which the system parameters are determined for prescribed SLA delay boundaries; the performance evaluation is executed for arbitrarily large server group sizes by a fast recursive numerical algorithm which had been reported in previous publications [18, 19].

The suggested method allows the parametric analysis for the probabilities of state, probabilities of delay and loss/migration, server utilization, queue lengths, average delays, distribution of delays and the power-saving efficiency by an automatic server consolidation algorithm based on a novel multiple hysteresis server consolidation mechanism. The method of capacity engineering for the resulting optimized system parameters of the Server Cluster queuing models has been derived and outlined in Section IV A. Two case studies were reported in Sections IV B, C which show the principal effects of economy of scale and the applicability to load balancing as a reaction to unbalanced load or overload to maintain agreed SLA requirements for the arriving jobs.

The paper shows that quite complex systems can be analyzed by the method of modeling and performance evaluations, from which a principal understanding of the parametric influences is supported, and multi-objective optimizations of the system operation can be achieved. Modeling requires, however, certain simplifications which have to be validated by either simulations of more realistic system models through system simulators or by experimental benchmarks and measurements. A short overview

on modeling methodologies has also been provided, for a more detailed review on the state-of-the-art it is referred to [13].

Our ongoing current work is directed towards the validation of our system models by simulation studies using standard simulation tools as OMNeT and the cloud simulation Framework CloudSim. For Markovian traffic assumptions our analyses methods are exact and need not be verified by simulations. Simulations, however, are adequate for performance studies with more general arrival and service time distribution functions to find out parametric sensitivities with respect to stochastic process variations and for more detailed models which are beyond the capabilities of theoretical performance evaluations; simulation studies are less adequate for extensive parametric studies compared to analytical model evaluations.

Cooperating SCs or even DCs are currently under study for two novel models for static and dynamic load balancing strategies LSSF (by the principle of job migration through mutual buffer overflow) and SRTF (by the principle of job migration through SC assignment by CC scheduling prior to job processing or even during ongoing job processing ("Life Migration") to be reported in a forthcoming paper [20].

Finally, lab experiments have been taken based on a small Cloud Lab at the GUC in Cairo which has been sponsored by the Cairo Competence Center of the company EMC² where experiments and measurements are performed within student theses and research projects.

REFERENCES

[1] B.P. Rimal; C. Eunmi; I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems," *INC, IMS and IDC, 2009. NCM '09. Fifth International Joint Conference*, pp. 44-51, 25-27 Aug. 2009.

[2] A. Gilati et. al., "VMware Distributed Resource Management: Design, Implementation, & Lessons Learned", VMware. Inc., White Paper, 2012.

[3] S. Kumar Garg, S.K. Gopalaiyengar, R. Buyya, "SLA-based Resource Provisioning for Hetrogeneous Workloads in a Virtualized Data Center", ICA3PP'11 Proceedings of the 11th international conference on algorithms and architectures for parallel processing - Volume Part I, Lecture Notes on Computer Science, Springer-Verlag Berlin, Heidelberg, 2011, pp. 371-384.

[4] A. Berl, et. al., "Energy-Efficient Cloud Computing", The Computer Journal, vol. 53, no.7, 2010, pp.1045-1051.

[5] RA. Gandhi, M. Harchol-Balter, R. Raghunathan, "Autoscale: Dynamic, Robust Capacity Management for Multi-Tier Data Centers", ACM Transactions on Computer Systems, vol. 30, no.4, 2012, pp. 14:1-14:26.

[6] Energy-Efficient Data Centers (E²DC): Conference Series, co-located with ACM Conferences e-Energy, Madrid/Spain (2012), Berkeley/Cal.(2013), Cambridge/UK (2014), Bangalore/Ind., (2015),Waterloo/Can. (2016). Publ. by Springer; since 2016 by ACM Digital Library.

[7] R. N. Calheiros, et. al., "CloudSim: a Toolkit for modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Journal Software Practice & Experience, Volume 41, Issue 1, January 2011, pp. 23-50

[8] C. Becker Westphall, et.al., "Green Clouds through Servers, Virtual Machines and Network Infrastructure Management", Chapter 6, 32 nd Brazilian Symposium on Computer Networks and Distributed Systems,[9]SBC - SBCR, 2014.

[9] H. Khazaei; J. Misic; V.B. Misic, "Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems," *IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No.5,pp. 936 - 943, May 212.*

[10] N.A. Brown Mary, K. Saravanan, "Performance Factors of Cloud Data Centers Using [(M/G/1) : (∞/GDModel)] Queuing Systems", International Journal of Grid Computing and Applications (IJGCA), vol. 4, no. 1, March 2013, pp. 1-9.

[11] A. Gandhi, V. Gupta, M. Harchol-Balter, M. A. Kozuch, "Optimality Analysis of Energy-Performance Trade-off for Server Farm Management", Journal of Performance Evaluation" vol. 67, issue 11, November, 2010, pp. 1155-1171.

[12] J. Idziorek, "Discrete Event Simulation Model for Analysis of Horizontal Scaling in the Cloud Computing Model," Proc. of the 2010 Winter Simulation Conference (WSC), pp.3004-3014, 5-8 Dec. 2010.

[13] P.J. Kuehn, "Energy Efficiency and Performance of Cloud Data Centers-Which Role Can Modeling Play ?", E²DC Conference, Waterloo/Canada June 2016 (Invited Paper), published in ACM Digital Library, 2016.

[14] P.J. Kuehn,"Performance and Energy Efficiency of Parallel Processing in Data Center Environments, E²DC , Cambridge 2014. Springer Lecture Notes in Computer Science LNCS No. 8945, pp.17 - 33.

[15] P. Barham, et.al., "Xen and the Art of Virtualization", Proc. of the 19th ACM Symp. on Operating Systems Principles, NY, 2003, pp. 164-177.

[16] VMware ESXi Server. Link: www.vmware.com

[17] P.J. Kuehn, "Systematic Classification of Self-adapting Algorithms for ‚Power-Saving Operation Modes of ICT Systems", Proceedings of the 2nd ACM International Conference on e-Energy: Energy-Efficient Computing and Networking, New York, 2011, pp. 51-54.

[18] P.J. Kuehn, M. Mashaly, "Performance of Self-Adapting Power Saving Algortihms for ICT Systems", Proceedings of IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2013), Ghent, 2013.

[19] P.J.Kuehn, M. Ezzat Mashaly, "Automatic Energy Efficiency Management of Data Center Resources by Load-Dependent Server Activation and Sleep Modes",:J. Ad Hoc Networks 25 (2015), pp. 497 - 504.

[20] P.J.Kuehn, M.Ezzat Mashaly, "Dynamic Load Balancing for Energy Efficiency and Quality of Service for Virtualized Cloud Data Centers", Forthcoming Paper.

[21] H. Kobayashi, B.L. Mark,"System Modeling and Analysis - Foundations of System Performance Evaluation", Pearson Education Inc., Upper Saddle River,N.J., 2009.

[22] P.J. Kuehn, " Approximate Analysis of General Queuing Networks by Decomposition", IEEE Trans. on Communications, vol. COM-27, no. 1, 1979, pp. 113-126.