Load Balancing in Distributed Cloud Data Center Configurations - Performance and Energy Efficiency

Paul J. Kueh
University of Stuttgart
70569 Stuttgart
Germany
paul.j.kuehn@ikr.uni-stuttgart.d

Maggie Ezzat Mashaly
German University in Cairo
New Cairo City
Egypt
maggie.ezzat@guc.edu.eg

ABSTRACT

In this contribution two cloud server clusters are considered which process virtualized user service requests defined as Virtual Machines (VM) operated under Hypervisor control. Load Balancing (LB) is applied to avoid temporary overloads and to enforce negotiated Service Level Agreements (SLA) defined by means and percentiles of processing delays. Two novel LB strategies are defined through which the two server clusters perform job processing cooperatively through mutual job overflows by a "Local Server System First" (LSSF) and through a "Shortest Response Time First" (SRTF) strategy, respectively. The cooperation operation is performed by VM migration at the instant of VM scheduling by the Hypervisor. Both LB models are defined by queuing systems which are analyzed by the method of Markov-Chains. Energy efficiency has been analyzed by the authors through server consolidation, server sleep modes, and through Dynamic Voltage and Frequency Scaling (DVFS), c.f. [7-11]. In this contribution another method is studied which is based on a flexible VM migration to a virtualized common server cluster by which the total number of servers can be reduced making use of the effect of the economy of scale by server aggregation.

COPYRIGHT

e-Energy ´17, May 16-19, 2017, Shatin, Hong Kong
© Copyright 2017 ACM
ISBN 978-1-4503-5036-5/17/05…$15.00
DOI: http://dx.doi.org/10.1145/3077839.3084027

CCS CONCEPTS

• Information systems~Data centers • Theory of computation~Scheduling algorithms • Computing methodologies~Planning and scheduling • Computer systems organization~Cloud computing • Hardware~Enterprise level and data centers power issues • Software and its engineering~Scheduling • Software and its engineering~Cloud computing

KEYWORDS

Cloud Data Centers, VM Migration, Service Level Agreements, Task Modeling & Scheduling & Queuing, Performance Analysis.

1 INTRODUCTION

Cloud Data Centers are increasingly integrated in our ICT infrastructures for storing of application or network configuration data, efficient searching, processing of mass data (“Big Data”), as well as for providing application processing functions (Software-as-a-Service, SaaS), The number of processing resources can be reduced considerably through the method of virtualization. Energy can be saved either by server consolidation if the current load does not require all activated servers of the configured local server systems, or by job (Virtual Machine, VM) migration when a new job can be transferred to another server system of the local DC or to a server system of a foreign DC when all servers of the local DC are currently occupied and if the negotiated Service Level Agreement (SLA) can be met. The performance of networked service systems can be improved by dynamic load balancing through transferring service requests arriving during a peak or overload period to another server system with currently available capacity. Job migrations can be decided either at the instant of arrival or during processing (“Life Migration”) (the latter one will not be considered in this paper). Dynamic load balancing aims at a better use of the resource configuration or for reasons to meet real-time service conditions with respect to the response time. Load balancing is usually achieved through static server system configurations, but this is not adaptable to quick load variations.

Virtualization allows for a more economic use of servers, but it may cause severe server "hot spots" of power consumption through the dynamic workload offered to the DC which may overload the DC cooling systems and, thus, lead to system downtimes and a worsened DC Power Usage Effectiveness (PUE) [6]. For this reason a novel dynamic (automatic) server consolidation method has been suggested in [1-5] by which the frequency of server activations/deactivations is reduced considerably and which contributes to a much smoother server usage dynamics and smaller hot spot risk, while still meeting the real-time SLAs. This method can be additionally combined

In this paper we develop two generic models for dynamic load balancing as a basis of performance analysis and studies of the efficiency of such load balancing algorithms. Server systems can be modeled by multidimensional Markov Chains under stationary conditions resulting in various types of overflow or VM assignment queuing models. Generally, models with overflow are more difficult to analyze due to the fact that the overflow changes the stochastic property of requests, e.g., by becoming non-renewal when inter-arrival times become dependent on each other. Service systems with overflow without buffering have been intensively studied for telecommunication switching systems and networks operating as loss systems, With the introduction of packet-oriented networks and web server farms queuing models with overflow traffic and with buffering have become of interest, Overflow, however, is usually only directed in one direction and not mutual as for load balancing in the current contribution. Service systems where an arriving request is assigned to a particular server system (scheduling) have appeared in the theoretic queuing literature repeatedly Applications of these models are typically to job dispatching methods in Web farms without server consolidation. Most of these models are difficult to solve exactly even under pure Markovian conditions. with the Load Balancing strategies suggested in this paper.

Queuing analyses can principally be applied to study effects of Energy-Efficiency as the power consumptions of actively processing ("busy") servers and actually ("idle") servers in the sleep states depend on the utilization factors of these resources. In this paper we suggest an optimized energy-efficient operation through job migrations to a virtually combined single server group; by this method the number of totally required servers can be reduced through the effects of "Economy of Scale", still by meeting the identical load and SLA conditions. We will show how the energy-efficiency can be improved considerably depending only the number of servers which can be saved by server aggregation and on the power consumption ratio between servers in the busy and in the sleep state.

The remaining part of the paper is structured as follows: In Section 2 two basic models for two server clusters are introduced which operate under different load balancing strategies. As mentioned before, the mathematical analyses for load balancing can also be used for the estimation of energy efficiency which can in principle be derived from the utilizations of servers in the various server states. In Section 3 a completely different approach for the estimation of energy-efficiency is presented which is based on a key property of multi-server service systems known as the "Economy of Scale" or "Bundling Gain". In Section 4. two examples of the proposed models for load balancing and energy efficiency will be provided before the paper is concludes with the main achievements.

2 MODELING OF COOPERATIVE SERVER SYSTEMS

A cloud data center is configured by multiple multi-server systems, where each server represents a multi- core processor. In the following, two generic models will be introduced for two multi-processor server systems (SS) with a limited buffer capacity to control the SLA which is defined by the response time of accepted jobs. The two SSs operate cooperatively so that they can accept jobs of the other SS by job migration through mutual overflow (LSSF), or by assigning arriving jobs instantaneously to that SS which guarantees the shortest response time (SRTF), respectively.

2.1 Local Server System First (LSSF)

The generic model for the LSSF strategy is shown in Figure 1. The operation of this model is as follows:

* Each SS serves its arrivals as long as an idle server is available. If all servers are occupied, arriving requests are assigned to the local queue and buffered there. Buffered jobs will be served in the strict FIFO mode. Features of the model are.

* If all buffer places within the queue of the local SS are occupied, an arriving request may overflow (or be migrated) to the complementary SS to be served there if the given SLA can be met.

* The buffer capacities s_1 and s_2 and migration thresholds x_1^* and x_2^* for the overflow to a complementary DC are chosen such that the SLAs of accepted arrivals will always be met.

* SLAs are defined by a threshold x_i^* for the mean response time $tW_{,T}$ of a request which has to wait and which is assigned to queue i, i = 1,2. More specifically, the SLA can also be defined to meet a prescribed percentile m̧ for the maximum delay an arrival will have to suffer.

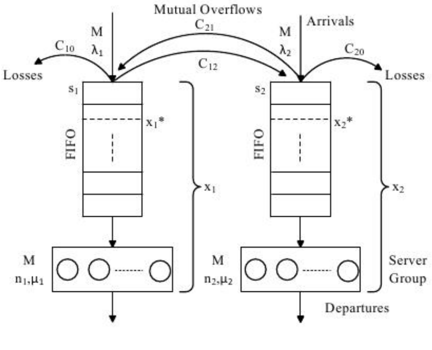

Figure 1: System Model for two Server Systems with Mutual Overflow.

The model parameters are defined as follows for the server group i, i = 1,2:

n_i number of servers
s_i buffer capacity,
$\bullet_i$ arrival rate of new requests,
$\circ_i$ service rate for a job i ($\circ_i$ = 1/h_i, h_i mean service time)
M Markovian traffic type for inter-arrival and service times
x_i system state at Server System i (number of jobs)
x_{ii*} state threshold for acceptance of migrated arrivals
t_M average time for a job migration
t_{WT} SLA threshold for mean response times
t_{Th}: SLA threshold for response time percentile:
C_{i0} conditions for rejection (loss) of an arrival at SS_i
C_{ij} conditions for migration of a new arrival at SS_i to SS_j,

Note: The C-values represent logical conditions and can be expressed accordingly by system state variables and are suppressed here for reasons of space.

For strategy LSSF a new arriving job will always be buffered in the queue of the local SS as long as the local queue is not completely filled. If the buffer is filled up and if there is space in the complementary local or foreign SS queue the arriving job overflows to the complementary queue if the state at the instant of arrival is such that the SLA cannot be met, the new request will not be accepted("lost").

The queue capacities si are dimensioned according to the condition that the mean response time tWi in each SS does not exceed a threshold tW,T in the worst case, i.e., for Markovian service times

$$s_i 1/n \ \circ \mathrm{i} \leq \mathrm{t}_{WT} \tag{1}$$

The state thresholds xi* are dimensioned such that the overhead time for a process migration from one (overloaded) SS to the complementary SS is considered such that SLA is still met for a migration from SS_i to SS_j, i.e., if

$$t_M + (x_j^* - n_j + 1)/n_j.\circ \mathrm{j} \leq t_{WT}$$

$$x_j^* \leq (t_{WT} - t_M) \cdot n_j.\circ \mathrm{j} + n_j - 1, \qquad j = 1,2 \tag{2}$$

For real-time services with a tighter SLA, the dimensioning of the acceptance/migration level can be derived from the complementary distribution function (DF) of accepted jobs W_i (> t) / W_i acc. To

$$W_i(> t_{Th}t)\,/\,W_i = P\{T_{Wi} > t_{Th}\,|\,T_{Wi} > 0\} < \text{ɱ} \tag{3}$$

where T_{Wi} denotes the random variable of the waiting time of an arriving job at SS_i, t_{Th} the response time threshold for an arriving and accepted job and ɱ the delay percentile, i.e., that the response time T_{Wi} of an arriving and accepted job exceeds the threshold t_{Th} only with a prescribed probability ɱ. The acceptance level x_i^* of an i-job at DCi follows from the worst-case arrival state $x_i^* > n_i$ at SS_i from (3); in that case, the response time distribution is an Erlangian DF of degree $k_i = x_i^* - {}_{ni} + 1$, i.e., the sum of k_i exponential phases each with mean h_i/n_i the arriving job has to wait until service begins. If the i-job arrives at state $x_i > x_i^*$, it might be migrated to the other data center SS_j, if the current state there is $x_j < x_j^*$. The condition for the value of x_j^* follows from the worst-case response time for the migrated job which is composed of the random job migration time TM and the waiting time at SS_j; the latter one follows again an Erlangian DF of degree $k_j = x_i^* - n_j + 1$, i.e., from k_j exponential phases each with average h_j/n_j. As the DF of T_M is generally not known, a proper assumption has to be made; in the most optimistic case TM could be assumed to be constant t_M; in that case, the response time DF is a shifted Erlangian DF. If the conditions at both DCs cannot be met, the new job must be rejected.

2.2 Shortest Response Time First (SRTF)

A special case of this strategy has already gained some research interest, better known under the name “Join the Shortest Queue" (JSQ), see references [15-20]. By this strategy, several single or multi--queue server systems with identical service time distributions and identical server numbers are considered. An arriving job is assigned to the currently shortest queue. The attraction of this policy is its guarantee of the absolutely shortest delay and its intrinsic strategy to balance the queue lengths instantaneously. We will consider a generalized strategy of JSQ as an alternative to the LSSF policy above under a modified condition acc. to the propositions of this paper, i.e., to guarantee given SLA's by scheduling the arriving job to that SS which provides the Shortest Response Time (SRTF) and allowing heterogeneous server groups with different numbers of servers, different buffer capacities, and even different server speeds. Figure 2 shows the server system arrangement.

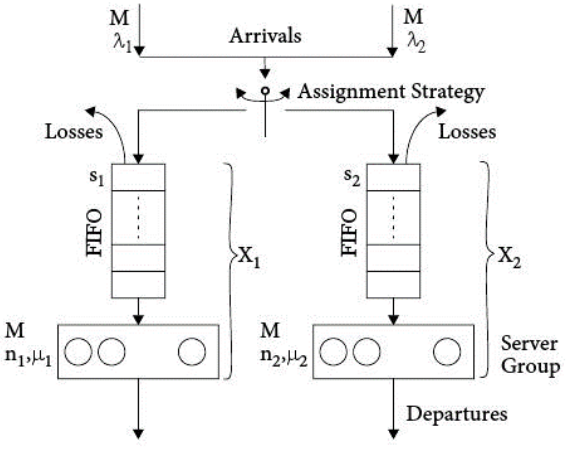

Figure 2: System Model for two Server Systems under SRTF.

The operation of the queuing model is as follows:

* An arriving job (VM), is assigned to that server system with the smallest number of busy servers, if at least one server is still idle in each SS. In principle, the arriving job could be assigned to any SS as the response time is zero in that case, but the assignment to that SS with the smaller number of busy servers enforces a better load balancing.

* If only one SS has fully occupied servers, the arriving job is assigned to the complementary SS.

* If all servers of both SSs are fully occupied the arriving job is assigned to that SS which provides the shortest mean response time, i.e., acc. to the following condition:

$$min\{(x_1 - n_1 + 1)h_1/n_1 , (x_2 - n_2 + 1)h_2/n_2\}$$

* If both queues are completely filled, the arriving job will be lost. The SRTF strategy provides the best load balancing, but requires a higher overhead as each arriving job has to be scheduled according to the condition above. However, this strategy outperforms all other scheduling strategies.

Let us consider the special case of two homogeneous SSs with identical number n of servers, identical service rates and identical number s of buffers for comparison. With each arriving job the assignment balances the load instantaneously and enforces that the probability mass of the state probabilities concentrates close to the main diagonal of the two-dimensional state space (x_1,x_2), see Section 2.3. This comes close to a virtual single service system of 2n servers and 2s buffers with FIFO queuing discipline. With the larger number of servers the SRTF profits from the effect of "economy of scale" and with the combined buffer space it results into the smallest probability of loss.

2.3 Mathematical Analyses

The mathematical analysis will be limited to Markovian assumptions, i.e., to negative-exponentially distributed inter-arrival and service times. The system operation will be represented by the method of State-Transition-Diagrams (STD) where states are represented by nodes (vertices) and transitions are represented by arcs (edges) of a directed graph. This representation holds generally, independent of the specific traffic assumptions. Transitions are annotated by transition rates; these annotations hold only for the case of Markovian traffic assumptions where the STD acts as representation of a Markov Chain.

In case of Markovian traffic assumption for the inter-arrival and service times, respectively, the systems are described by a 2-dimensional state (x_1,x_2), where $x_i < n_i + s_i$, i= 1,2. The state transition diagrams (STD) are given in Figure 3 and Figure 4 for the two cases of LSSF and SRTF, respectively. Both server groups are heterogeneous in general; for the example analyses we have chosen both SSs with identical numbers and identical processing speeds as well as identical QoS for simplicity. For the case of SRTF the homogeneous server system models result in effect into the JSQ policy. The circles in Figure 3 and Figure 4 represent the joint system states with x_1 in the upper and x_2 in the lower part. The transitions between the states are represented by directed arcs which are annotated by the upward and downward transition rates of the corresponding two-dimensional Markov Chains. Differences between the two models are shortly commented as follows:

LSSF: The migration levels for the LSSF are indicated by the dashed lines with the corresponding thresholds x_1^* and x_2^* for job migration from SS2 to SS1 or SS1 to SS2, respectively. State transitions which end at the same state represent those events where an arriving job cannot be accepted (i.e., when the SLA conditions are not fulfilled) and indicate, thus, that the arrived job is rejected or lost. The transition rates for the state equations are completely indicated in Figure 3.

SRTF: The transition rates are more complex and are partly represented by functions •(Q1); •(Q2) follows analogously:

$$\lambda(Q\,1)=\begin{cases} \lambda_1+\lambda_2 & \text{if } (x_1\text{-n}_1\text{+}1)h_1/\text{n}_1 < (x_2\text{-n}_2\text{+}1)h_2/\text{n}_2 \\ \lambda_1 & \text{if } (x_1\text{-n}_1\text{+}1)h_1/\text{n}_1 = (x_2\text{-n}_2\text{+}1)h_2/\text{n}_2 \\ 0 & \text{if } (x_1\text{-n}_1\text{+}1)h_1/\text{n}_1 > (x_2\text{-n}_2\text{+}1)h_2/\text{n}_2 \end{cases}$$

There are no closed-form solutions known for these problems, i.e.,they have to be solved by a numerical solution of the balance equations for the probabilities of state p(x_1,x_2). From the stationary state probabilities p(x_1,x_2), the most important performance metrics are derived straightforward for VM_i, i = 1,2:

Y_i Carried traffic (server group occupancy)
L_i Average queue lengths
B_i Probability of loss
W_i Probability of delay
w_i Mean waiting time (all arrivals)
t_{Wi} Mean waiting time (delayed arrivals)
M_i Migration probability to the complementary DC
$W_i(>t)$ Complementary delay distribution function.

3 ENERGY EFFICIENCY

For a principle comparison we will consider only pure delay models. In Figure 5a we have two separated, identical non-cooperative server systems with n_1 servers and job arrival rate •; in Figure 5b we have one server system with an aggregated server group of n_2 servers, but with the doubled job arrival rate 2•. Model 5b stands for the (idealized) result of job migrations to a virtual single server system queue. The mean service time is identically h for both models. In both models we will consider unlimited buffer space, i.e., $s_1$and s_2 are infinitely large.

The carried traffic Y (i.e., the average number of occupied servers) is identical to the offered traffic A = ✦h, the average number of idle (or sleeping) servers is ✦h, the average number of idle (or sleeping) servers is (n - A). The power consumptions will be denoted by PP for a processor actively processing a job and by PS for a server being in the sleeping state. The power consumptions for both models 1 (Figure 5a) and 2 (Figure 5b) can be expressed as follows:

$$\begin{aligned} P_1 &= 2[A_1 P_P + (n_1 - A_1)P_s] \\ P_2 &= A_2 P_P + (n_2 - A_2)P_s \; where \; A_2 = 2A_1 \end{aligned} \qquad (4)$$

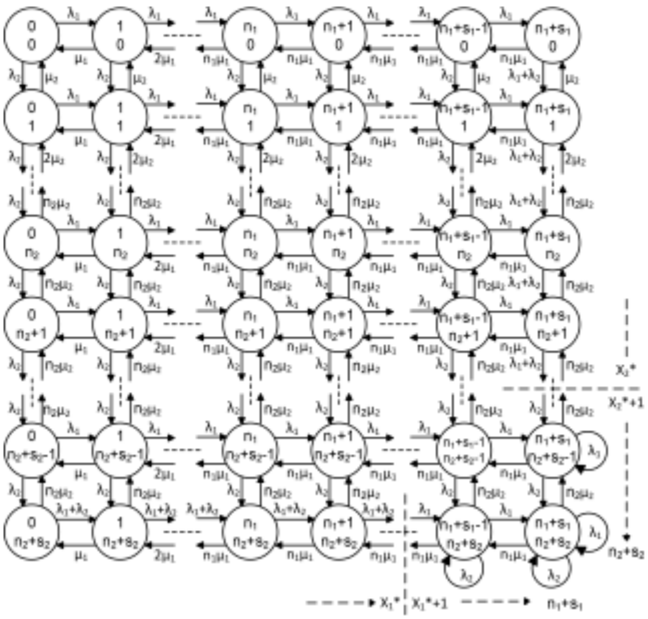

Figure 3: State Transition Diagram for LSSF.

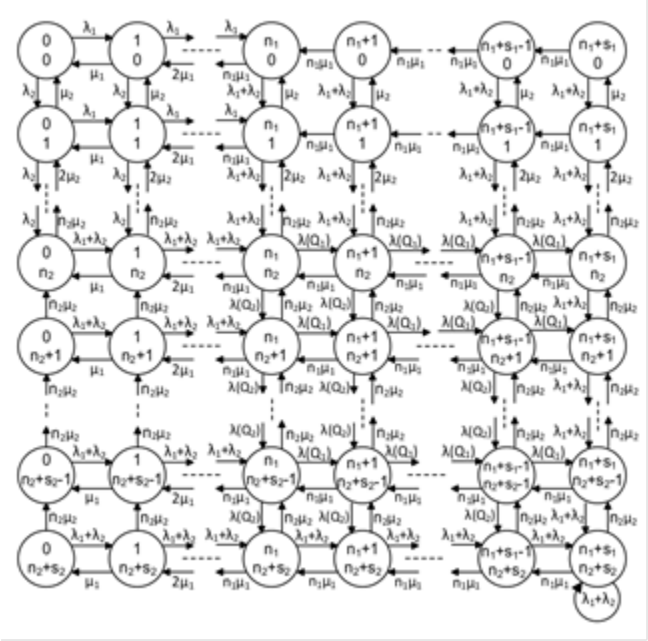

Figure 4: State Transition Diagram for SRTF.

To make sure that both models meet the identical SLA the number of servers n_2 for the aggregated model has to be chosen accordingly. As we prefer the mean waiting time of waiting jobs as a Quality of Experience for a SLA metric we result at the condition $t_{W1} = t_{W2}$, where $t_W = E[T_W|T_W > 0]$. Using Markovian traffic assumptions we can make use of the extremely simple result for the M/M/n queue; $t_W = h/(n\text{-}A)$. From this relation we find the result for the number n_2 of virtualized servers from

$$t_{W1} = \frac{h}{n_1 - A_1}, t_{W2} = \frac{h}{n_2 - 2A_1}, \qquad (5)$$

$$n_2 = n_1 - A_1$$

For processors with active and sleep states we know actual values of power consumptions from [24] for the Intel Pentium M 1.6 GHz processor at different p-states ranging from P0 (25 W) to P6 (6W) for different voltages and clock frequencies (Dynamic Voltage and Frequency Scaling, DVFS) which we have also considered in our automatic server consolidation modeling [4]. The energy-efficiency h for the comparison of the two models 1 and 2 can be expressed by the power gained through the effect of "Economy of Scale" by the aggregated server group related to the Model 1 without aggregation:

$$\eta = \frac{P_1 - P_2}{P_1} = 1 - \frac{2A_r + (n_2 - 2A)}{2A_r + (n_1 - A)}, \qquad (6)$$

$$where\ r = P_P/P_S$$

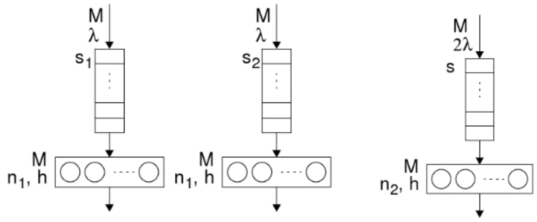

Figure 5: a, b: Two Non-cooperative Server Systems (left) and one Virtualized Common Server System (right).

4 EXAMPLE RESULTS

4.1 Load Balancing

Two use cases are presented in Figs 6a,b for the two LB strategies LSSF and SRTF for two identical SSs with n = 10 servers, s = 30 buffers without/with migration overhaed delay and SLA threshold tW/h = 3 (Figure 6a) and two local SSs under the idealized SRTF strategy with n = 20, s = 60, tM = 0 and 2 (Figure 6b).

4.2 Energy Efficiency

Taking the best power ratio P0/P6 25 W/6W = 4.167 of [12] between the active state "Processing" and the "Sleep" state for our example, Table 1 shows the relative power gain in % for the four server cases of n_1 and three load cases of A_1/n_1.

Table 1: Relative Power Gains for Virtualized Servers

n₁	**4**		**10**		**20**		**30**	
	n_2		n_2		n_2		n_2	
$A/n_1 = 0.1$	5	38	11	34	22	34	33	34
$A/n_1 = 0.5$	6	9.7	15	9.7	30	9.7	45	9.7
$A/n_1 = 0.8$	8	0	18	2.9	36	2.8	54	2.8

Discussion: The "Economy of Scale" works well for server numbers in the range below 50; the higher the bundle size n, the smaller becomes the gain, which is well-known from classical traffic theory. With increasing traffic load the efficiency decreases, i.e., the best gain is for small and medium loads (which is the experience from Cloud Data Center servers).

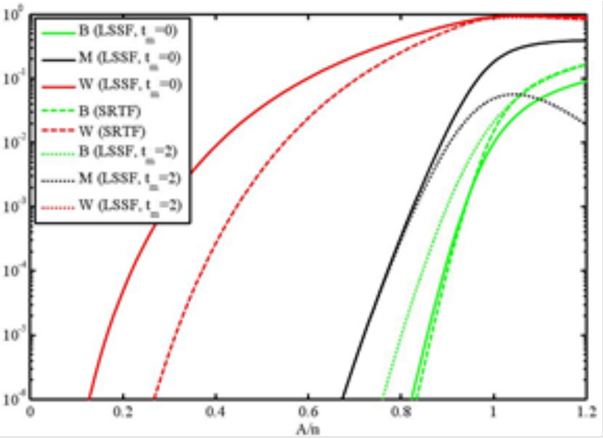

Figure 6a: Migration (M) and Delay (W) Probabilities vs. Load A/n. One Local, 1 Foreign SS for LSSF with/without Migration Overhead Time.

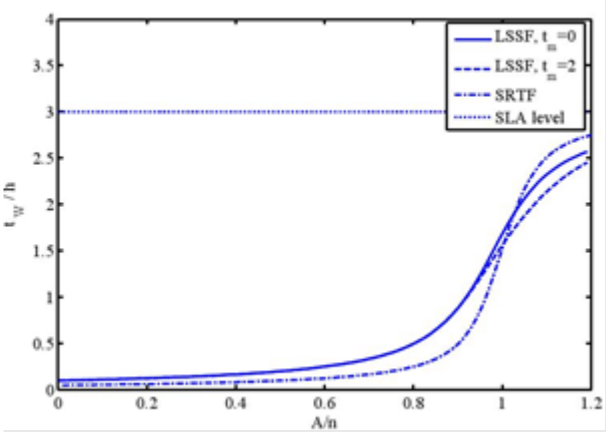

Figure 6b. Mean Response Time of Delayed Jobs vs. Load. SRTF: Dotted-Dashed Curve.

5 CONCLUSION

In this contribution two pre-configured server systems (SS) with finite buffer capacities for arriving jobs in a Data Center (DC) are modeled by finite-buffer queuing systems with two different load-balancing strategies to improve the performance in cases of temporary overload caused by stochastically varying job execution times. In the LSSF model jobs are migrated in overload situations to a companion SS in the local DC or in a foreign DC under the condition to meet the SLA using the principle of mutual overflow, even under consideration of a finite migration time for the VM transfer. The results are compared to the migration strategy SRTF which outperforms LSSF. These models can be enhanced by application of the method of server consolidation by a novel hysteresis mechanism which saves energy by reduction of the frequency of server activations, again under the same restriction that the SLAs, are met. Load balancing can also be performed by job migrations to a larger common (virtual) SS to make use of bundling gain effects. by which considerably large efficiency gains can be achieved in medium sized multi-processor ranges..

REFERENCES

[1] P.J. Kuehn, “Systematic Classification of Self-Adapting Algorithms for Power-Saving Operation Modes of ICT Systems”, Proc. 2Nd ACM Conf. on Energy-Efficient Computing and Networking (e-Energy 2011), New York, N.Y., USA, May 30- June 1, 2011. http://edas.info/N9577

[2] P.J. Kuehn, M. Mashaly, “Performance of Self-Adapting Power-Saving Algorithms for ICT Systems”, IFIP/IEEE Symp. On Integrated Network and Service Management” (IM 2013), Ghent, Belgium, May 27 – 30, 2013, IEEE X-plore.

[3] M. Mashaly, P.J. Kuehn, “Modeling and Analysis of Virtualized Multi-Service Cloud Data Centers with Automatic Server Consolidation and Prescribed Service Level Agreements", 41st IEEE Conference on Local Computer Networks, LCN 2016, Dubai, pp. 9-16, IEEE Xplore.

[4] M. Mashaly, P.J. Kuehn, “Modeling and Analysis of Virtualized Multi-Service Cloud Data Centers with Automatic Server Consolidation and Prescribed Service Level Agreements", 41st IEEE Conference on Local Computer Networks, LCN 2016, Dubai, pp. 9-16, IEEE Xplore.

[5] P.J.Kuehn "Energy Efficiency and Performance of Cloud Data Centers - Which Role Can Modeling Play ? Inv. Contribution to E^2DC 2016, Waterloo, Canada. ACM Digital Library.

[7] Hwa-Chun Lin, C.S. Raghavendra, “An Approximate Analysis of the Join the Shortest Queue (JSQ) Policy”, IEEE Trans. On Parallel and Distributed Systems, Vol. 7, No.2, pp 301 – 307.

[8] V. Gupta, M.Harchol-Balter, K. Sigman, W. Whitt, Analysis of Join- the-Shortest-Queue Routing for Web Server Farms”, J. Performance Evaluation Architecture, Vol. 64, Issue 9 – 12, Oct. 2007, pp. 1062 – 1081

[9] S.Abimannan, K. Durai, A.V.Jejahumar, S. Krishnaveni, "Join-The- Shortest Queue Policy in Web Server Farms", Global Journal of Computer Science and Technology, Vol. 10, June 2010, pp. 39 - 45.

[10] I.J.B.F. Adan, O.J. Boxma, J.A.C. Resing, "Queuing Models with Multiple Waiting Lines", Queuing Systems No. 37, 2001, pp. 65 - 98.

[11] M. Harchiol-Balter, "Scheduling in Server Farms", Talk Slides, Carnegie-Mellon University, harchol@cs.cmu.edu

[12] H.H.Kramer, V.Petrucci, A.Subramanian,E.Uchoa,"A Common Generation Approach for Power-Aware Optimization of Virtualized Heterogeneous Server Clusters", J. of Computers and Industrial Engineering, vol. 63, issue 3, pp. 652 - 662, Nov. 2012.