Ad Hoc Networks 25 (2015) 497–504

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

Automatic energy efficiency management of data center resources by load-dependent server activation and sleep modes

Paul J. Kuehn [a,*], Maggie Ezzat Mashaly [b]

[a] Institute of Communication Networks and Computer Engineering, University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany
[b] Networks Department, German University in Cairo (GUC), Egypt

ARTICLE INFO

Article history:
Received 26 June 2014
Received in revised form 11 November 2014
Accepted 11 November 2014
Available online 20 November 2014

Keywords:
Data center server consolidation
Resource management
Energy efficiency
Service Level Agreement (SLA)
FSM-controlled queuing model
Performance evaluation

ABSTRACT

The operation of large Data Centers (DC) with thousands of servers is very costly in terms of energy consumption and cooling requirements. Currently, major efforts can be observed for server virtualization and consolidation to approach a proportionality between computation amount and energy consumption. In this contribution, a generalized model is presented which allows an automatic server consolidation by a load-dependent control of server activations using multi-parallel hysteresis thresholds, cold and hot server standby, and Dynamic Voltage and Frequency Scaling (DVFS). For the energy-efficiency and performance analysis, a multi-server queuing model is defined which is controlled by a Finite State Machine (FSM). The parameters of the queuing model are defined such that Service Level Agreements (SLA, e.g. as mean or percentiles of response times) are guaranteed except for overload conditions. The queuing model can be exactly analyzed under Markovian process assumptions from which all relevant quality of service (QoS) and energy efficiency (EE) metrics are derived. Numerical results are provided which demonstrate the applicability of the proposed model for the DC management, in particular to theoretically quantify the tradeoff between the conflicting aims of EE and QoS.

1. Introduction

High-speed fixed, mobile and wireless networks, web-based, social and multi-media applications are the driving forces of the currently ongoing paradigm shift from a communication-centric to an information-centric internet. Traditional IT infrastructures are challenged by cloud networks with distributed data centers which make full use of multi-core processing and huge storage capacities allowing fast access to data, search and business process applications. The huge energy demand of these data centers contributes significantly to the energy consumption and the "carbon foot print" and caused world-wide efforts to the "Green ICT" movement.

Energy efficiency can be improved on quite different levels: device level by a steady miniaturization of transistor elements, on the circuit/chip level by low-power circuit design, on the network level through efficient use of bandwidth by modulation and coding, switching and routing protocols, on the application level by energy-aware user behavior, and on the systems level by system operation management. This paper addresses energy efficiency management, specifically through modeling and quantitative analysis by stochastic queuing theory. In this approach, energy-consuming resources are modeled as "servers" (e.g., a processing device which executes a task) and "buffers". The energy consumption of a processor is usually simplified by a constant representing the average consumption during processing. Variable energy consumption

* Corresponding author.
E-mail address: paul.j.kuehn@ikr.uni-stuttgart.de (P.J. Kuehn).

http://dx.doi.org/10.1016/j.adhoc.2014.11.013
1570-8705/

through, e.g., dynamic pipeline operations or caching, belong to a lower level closer to the hardware and require more knowledge on the underlying application program and will not be considered in this contribution. The dynamic behavior of a whole data center or of specific server groups is modeled by "stochastic arrival processes" of processing tasks and task execution times are modeled by "stochastic service times". Tasks which cannot be executed immediately are buffered in a "queue".

Actions for energy efficient operations on the systems level are deactivations at low-load situations and activations at high-load situations, sleep mode (non-operative mode, e.g. by lowering the power supply to avoid booting in case of reactivation), or slow-mode operation by clock frequency throttling; the latter two operations are known as Dynamic Voltage and Frequency Scaling (DVFS). Control actions on multiple servers aim at resource management through adaptive assignment of tasks to server groups by virtualization methods, server consolidation by activity monitoring and deactivation of servers, by load sharing for tasks among different processors (scheduling), or by load balancing through shifting tasks to other virtual machines of the same or of remote data centers (task or process migration).

Energy efficiency has become a hot research topic in the recent years reflected by high publication activities. Most contributions address architecture, measurement and management issues, c.f. [1–4]. Another group of papers approach the problem by modeling and queuing theory, see, e.g. [5], where the data center is modeled by a multi-server queuing system. Self-adaptive server consolidations have been suggested and modeled by the authors on the basis of hysteresis mechanisms by FSM-controlled queuing systems [6–8] and applied to load balancing [13]. The current paper is a revised and extended version of a conference paper [14]; it presents a more detailed modeling approach for multi-server queuing systems which are controlled by a Finite State Machine allowing automatic adaption to the load level and a detailed consideration of specific control schemes and overhead involved with dynamic server activations.

The rest of the paper is structured as follows: In Section 2, the structure and the parameters of the queuing model are presented together with the design criteria for the intended operation mode of the FSM which controls the adaptive algorithms for power saving. In Section 3, a short review is given over existing literature on queuing models with hysteresis control. For the generalized model, we have developed a new recursive solution algorithm which allows for an effective calculation of the probabilities of state for an arbitrarily large number of servers and the most characteristic performance and energy consumption values. Finally, Section 4 provides a numerical case study and discusses the parametric influences on the performance.

2. Queuing model

According to the stated features of the DC queuing model, we have to construct the State Transition Diagram (STD) of the FSM in a systematic way. For this, the following characteristics have to be satisfied by the FSM:

(1) Multiple hysteresis thresholds to avoid frequent oscillations between activations and deactivations of server resources to serve stochastically varying service requests and for an automatic self-adaptation to highly volatile load variations.
(2) Throttling of new server activations upon short load bursts by buffering of the requests up to scalable upper thresholds.
(3) Serving of task requests with the maximum service rate of the activated servers to keep delays as small as possible as long as (4) is not affected.
(4) Threshold parameters of the hystereses have to be set such that a prescribed Service Level Agreement (SLA) is guaranteed except for overload situations. Overload situations are defined when all servers are already activated and new requests could not be served under the given SLA.
(5) Throttling of server deactivations when the queue of waiting task requests falls below of a scalable lower threshold (implemented by the DFS principle).
(6) Consideration of two different deactivation modes:
    (6.1) If a server becomes idle, it will be set in a sleep mode with lower power consumption from which it can be reactivated quickly ("warmup") without booting ("hot stand-by mode", HSB).
    (6.2) If a server becomes idle, it will be completely deactivated (switched-off) and has to be booted again if a new server activation is required ("cold stand-by mode", CSB).
(7) Sleeping or deactivated servers are activated again at the instant of a task request arrival and under a predefined threshold for buffered requests.
    (7.1) In case of a sleeping server, activation takes a short warmup time.
    (7.2) In case of a deactivated (switched-off) server, activation requires a longer activation time for booting. Activated or reactivated servers start servicing buffered requests immediately after booting or warmup according to the FIFO (First-In, First-Out) queue discipline.

STDs for multiple serial and parallel hystereses without activation overhead and without DVFS were reported in [6,7] and extended to activation overhead in [8] by the authors. This paper extends these results with respect to DVFS, cold and hot stand-by.

Fig. 1 shows a generic queuing model with dynamic activations/deactivations of servers acc. to [6–8] for generally distributed task requests (Arrival Process Type G, arrival rate λ tasks/s), generally distributed task service times (Service Process Type G, maximum service rate μ per activated server), buffer with capacity s, the Finite State Machine (FSM), a server group with n servers, and a scheduler. The state of the queuing system is indicated by the vector (X, Z), where X denotes the current number of busy (i.e., service executing) servers and Z of waiting task requests. The variables X and Z are reported to the FSM

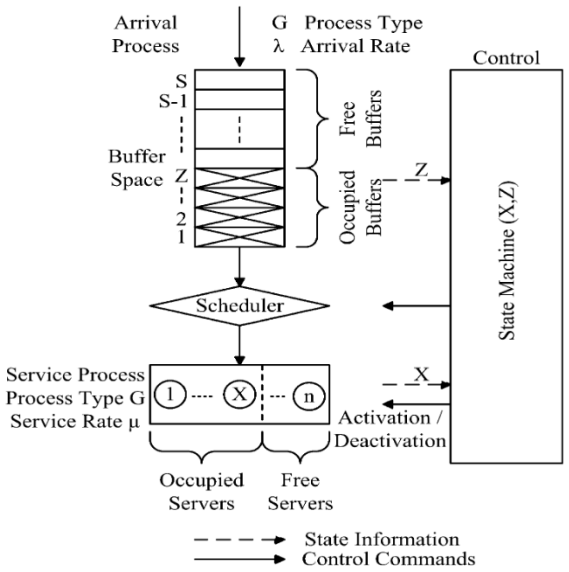

Fig. 1. General queuing model for DC servers, buffers and control.

which controls the operation on the servers as activation, deactivation/sleeping, and service commands which are executed by the scheduler. X and Z are considered as random variables.

The model of Fig. 1 is implemented for simulation where arbitrary arrival and service processes can be assumed. For the analytical performance evaluation we will restrict ourselves to Markovian arrival and service processes, i.e., negative-exponentially distributed interarrival, service, warm-up or booting times, respectively.

In Fig. 2, the STD is presented for the case of Markovian arrival and service processes (quasi 2-dimensional Markov Chain).

The many features of the DC model stated above lead to a more complex STD of the FSM which is captured by two sorts of states distinguished by shading. Consider first the lower-right section of states (circles) which are white (not shaded). This is the model for zero overhead for server activations/deactivations, i.e., when the DC is operated in an ideal CSB mode without sleeping. (*Note:* In that case a new arrival at the border states (0,0), $(x, w^{(x)})$, $x = 1,2,\ldots,n-1$, where $w_0 = w^{(0)} = 0$, $w_1 = w^{(1)}$ leads to an immediate server activation by a horizontal transition arrow into state (1,0), $(x+1,\ w^{(x)})$, $x = 1,2,\ldots,n-1$), respectively, not shown in Fig. 2, c.f. the earlier paper [7].)

The threshold values $w^{(x)}$ can be derived from given SLA, indicated either by mean response time or even by a percentile of the response time distribution function. If the mean response t_{W0} of an arriving request which has to wait is chosen as SLA, $w^{(x)}$ follows from the following worst-case observation: If the new arrival meets state $(x, w^{(x)} - 1)$ it has to wait in average not longer than $w^{(x)}/x\mu$ because of FIFO queue discipline which results in $w^{(x)} = t_{W0}x\mu$, $x = 1,2,\ldots,n$. In this case, the integer-valued threshold values $w^{(x)}$ for server activations are increased stepwise with x by $w = t_{w0}\mu$.

This model meets already the characteristics (1)–(4) as stated above. Characteristics (5)–(7) are met by the extension of Fig. 2 by the shaded states. The highlighted transition arrows indicate the events of a server starting processing or becoming inactive (switched-off or going to sleep).

The extended STD of Fig. 2 by the shaded states reflects the further features (5)–(7) of the requirements stated above. The circles indicate the actual state; the blank circles (x, z) indicate currently x processing servers and z waiting task requests, where all other servers are switched off or they are sleeping. The shaded states indicate all those states where one or several servers are presently either in the activation phase ("booting") for CSB mode or in the warmup phase for HSB mode of operations. The state variable x for the shaded states implies that up to $(n - x)$ servers are in booting or in warmup phase whose averages differ only in the value $1/\alpha$. For example, the shaded state $(1, w_1 + 1)$ means that still one server is processing, but one server is in the activation phase (i.e., booting or warmup). The activation transition is, accordingly, annotated by "A" on the corresponding transition from state $(1, w_1)$ to $(1, w_1 + 1)$. (*Note:* To avoid a 3-dimensional state description we have used shading which indicates currently ongoing server activations, i.e. either in warmup after sleeping or in booting after being switched off.) The number of servers which are currently in the process of activation follow implicitly from the STD as servers are triggered for activation at the same system state levels $(x + z)$ of the hystereses as in case without activation overhead. The actual number x_A of servers being in a booting or warmup phase for a shaded state (x, z) is $x_A = \left(\frac{z - w^{(x)}}{w}\right)$. Transitions marked by "A" indicate the activation of an idle or sleeping server. Transitions marked by "D" indicate the deactivation of a busy server or a server during its activation phase.

State transitions are represented by annotated arrows. The annotations indicate the transition rates λ (for a new arrival of a task request) and $\mu_{x,z}$ (for a server termination with the aggregated termination rate of all x busy servers). In all cases without DFS $\mu_{x,z} = x * \mu$. In the case DFS, the server termination rates can be modified

$$\mu_{x,z} = x * \mu \quad \text{for } z > z^*$$
$$\mu_{x,z} = x * \mu^* \quad \text{for } z \leqslant z^*$$

where z^* indicates the queue state up to which the server speed is reduced by DFS, $x = 1,2,\ldots,n$, and $\mu^* < \mu$ the reduced service rate. (*Note:* Simple settings can be $z^* = 0$ or 1.) Note, that this model is based on a state-dependent DFS, while other implementations reported in the literature are based on time-dependent DFS.

Server activations in the ideal system with zero overhead are triggered at states $(x, w^{(x)})$, where $w^{(x)} = w_1 + w_2 + \cdots + w_x$, $x = 1,2,\ldots,n-1$, indicate the queue thresholds for triggering the next server activation (or reactivation), while w_x denotes the threshold increase for buffering new arriving task requests in the state of x busy servers (boundary values $w_0 = 0$, $w_n = s$). The activation thresholds for non-zero activation overhead are based on the identical system state level $(x + z)$ as in case of zero overhead.

State transitions caused by the end of a warmup phase /booting phase of a deactivated server are indicated by

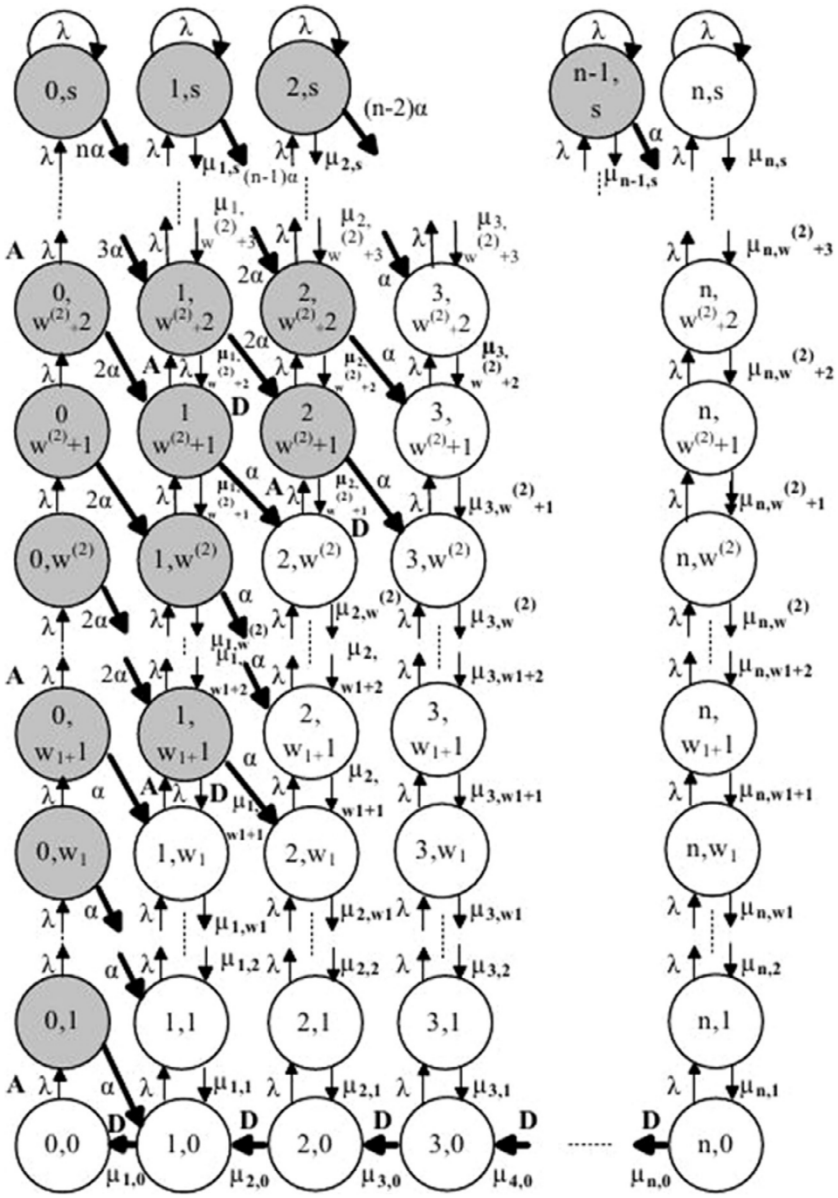

Fig. 2. State transition diagram for multiple parallel hystereses, activated and deactivated/sleeping servers and dynamic activation/deactivation of servers.

$y * \alpha$, where y indicates the number of servers being simultaneously in the phase of warmup/booting. The difference between warmup and booting lies only in the choice of α: Small values of α indicate the (longer) booting phase, while larger values of α indicate the (shorter) warmup phase. We assume that both warmup and booting phases are negative-exponentially distributed with mean $1/\alpha$. In this paper, we consider deactivations either as switching-off (CSB mode) or as turning to sleep (HSB mode).

3. Performance analysis

The queuing model defined in Section 2 will be analyzed mathematically under Markovian process assumptions. For general arrival and service processes no exact method exists; this case is analyzed by event-by-event simulations.

3.1. Theoretical background

Markovian queueing models with hysteresis thresholds have repeatedly appeared in the literature. For a single hysteresis, Tran-Gia [9] derived a closed-form solution by use of macro-state aggregation. Generalized models have been treated by two seminal contributions by Ibe and Keilson [10] and by Liu and Golubchik [11] based on the method of Green's function applied to half-lattice Markov Chains with ergodic compensation rates or by stochastic

complementation, respectively. Both arrive at closed-form solutions for the probabilities of state. The authors of [11] extended their analysis to the case of non-negligible server activation times [12]. All these solutions [10–12] are computationally difficult and have been applied only to very small numbers of servers which is not adequate for our current applications.

Our Markovian chain models will be analyzed by a new recursive algorithm. The recursive property of Markov graphs as the one in Fig. 2 has not been discovered before and allows a fast computation of the probabilities of state for arbitrary numbers of servers. It has been applied to various models with serial and parallel hystereses [6,8,13] and can be extended to the more general model of Fig. 2 as well.

3.2. Recursive algorithm

The Markov chain of Fig. 2 can be solved recursively under stationarity according to the following algorithm:

Step 0: Assume $p(0,s)$, e.g. $p(0,s) = 1$.
Step 1: Balance equations for states $(0,s),\ldots,(0,1)$ yield $p(0,z)$, $z = s-1,\ldots,0$, as function of $p(0,s)$.
Step 2: Assume $p(x,s)$ as a parameter, beginning with $x = 1$.
Step 3: Balance equations for states $(x,s),\ldots,(x,1)$ yield $p(x,z)$, $z = s-1,\ldots,0$, as function of $p(0,s)$ and $p(x,s)$.
Step 4: Balance equation for state $(x,0)$ yields $p(x,s)$ and successively all $p(x,z)$, $z = s-1,\ldots,0$, as function of $p(0,s)$.
Step 5: Repeat Steps 2, 3 and 4 for $x = 2,\ldots,n$.
Step 6: Normalization: $\sum_{x=0}^{n}\sum_{z=0}^{s} p(x,z) = 1$ yields $p(0,s)$.

$$p(x,z) := p(x,z) * p(0,s) \quad \text{for } x = 0,\ldots,n;\ z = 0\ldots s$$

Note: The numerical values of the state probabilities may stretch over a huge range, dependent on the set of system parameters, especially with respect to λ and s. This may cause numerical problems and can be solved by proper truncation of ranges with extremely low values.

3.3. Performance values

From the probabilities of state, the most characteristic performance values can be derived:

- State distribution of active (busy) servers

$$P(x) = \sum_{z=0}^{s} p(x,z) \tag{1}$$

- Average number of busy servers

$$Y_S = \sum_{x=1}^{n} x \cdot P(x) \tag{2a}$$

- Average number of servers in activation phase

$$Y_A = \sum_{x=0}^{n-1} \sum_{z=w^{(x)}+1}^{s} x_A \cdot p(x,z) \quad \text{where}$$
$$x_A = \left(\frac{z - w^{(x)}}{w}\right) \tag{2b}$$

- State distribution of buffered task requests

$$Q(z) = \sum_{x=0}^{n} p(x,z) \tag{3}$$

- Mean queue length of buffered requests

$$L = \sum_{z=0}^{s} z \cdot Q(z) \tag{4}$$

- Probability of loss (blocking)

$$B = \sum_{x=0}^{n} p(x,s) \tag{5}$$

- Probability of delay upon arrival

$$W = 1 - B \tag{6}$$

- Mean waiting time of arriving requests (Little's Law)

$$E[T_W] = L/\lambda \tag{7a}$$

- Mean waiting time of buffered requests

$$E[T_W | T_W > 0] = L/\lambda W \tag{7b}$$

- Activation rate of deactivated/sleeping servers

$$R_A = \lambda \cdot \sum_{x=0}^{n-1} \sum_{i=x}^{n-1} p(x, w^{(i)} + i - x) \tag{8}$$

- Power consumption by servers in CSB mode

$$P_{CSB} = P_0 \left[Y_S - \sum_{x=1}^{n} \sum_{z=0}^{z} x p(x,z) \cdot (\mu - \mu)/\mu \right] + P_{A,CSB} \tag{9a}$$

where P_0 is the power consumption of one active server at full speed, μ the service rate of a server at full speed, μ^* the reduced service rate by DFS, CSB the cold stand-by, and $P_{A,CSB}$ is the acc. to Eq. (9c).

- Power consumption by servers in HSB mode

$$P_{HSB} = P_0 \left[Y_S - \sum_{x=1}^{n} \sum_{z=0}^{z^*} x \cdot p(x,z) \cdot (\mu - \mu^*)/\mu \right] + P_{A,HSB} + (n - Y_S - Y_A) P_0^* \tag{9b}$$

where P_0^* is the power consumption of one sleeping server, HSB the hot stand-by, and $P_{A,HSB}$ is the acc. to Eq. (9c).

- Power consumption for activating sleeping/switched-off servers

$$P_{A,HSB} = Y_{A,HSB} \cdot P_0, \quad P_{A,CSB} = Y_{A,CSB} \cdot P_0 \tag{9c}$$

Note to formulas (9a)–(9c):

(9a) is based on a value of α for CSB.
(9b) is based on a value of α for HSB.
(9c) formula applies to both operation modes based on the corresponding values for α.

There are various possibilities to define "power efficiency". Note, that the highest gain in power saving is achieved for low load situations compared to the operation

without power-saving mechanisms. In this paper, power efficiency η will be defined as a fraction of the amount of power saved referred to the power which is necessary if no power-saving method is applied.

- Power-saving efficiency

$$\eta_{CSB} = (nP_0 - P_{CSB})/nP_0 \tag{10a}$$

$$\eta_{HSP} = (nP_0 - P_{HSB})/nP_0 \tag{10b}$$

4. Applications and discussion

The queuing model defined in Section 2 will be analyzed mathematically under Markovian process assumptions. For general arrival and service processes no exact method exists; this case can be analyzed by event-by-event simulations.

Below, several numerical results on the performance and the power efficiency are illustrated for an example case for the specific model parameters:

$n = 100$ servers,
$s = n \cdot w$ buffer capacity,
$w_x = w$, incremental queue thresholds for server activations, $x = 1, 2, \ldots, n-1$,
$\lambda = 0, \ldots, 100$ arrival rate (av. number of arrivals per unit time),
$\mu = 1$, i.e., mean service time $= 1/\mu = 1$ (unit of time),
$\alpha/\mu = 0.25, 0.50, 1.0, \infty$,
normalized termination rates for a new server activation,
$\mu^*/\mu = 0.1$ and 1.0 reduced service rate by DFS at level $z_x^* = 2$, $x = 1, 2, \ldots, n$,
P_0^*/P_0 sleep power ratio,
μ^*/μ service ratio for DVS.

Fig. 3 illustrates the probability accumulation effect caused by buffering of arriving requests acc. to the parallel hystereses, where the mass of state probabilities is concentrated around $x = \lambda/\mu$. This effect is illustrated for 3 cases $x = 0$ (all servers are idle) $x = 50$ and $x = 100$ (all servers are processing) of the distribution of the probabilities of state (x) acc. to Eq. (1) versus the arrival rate λ. For each case one can see that the shape of the curves becomes steeper with increasing w (through more buffering) before a new server is activated. Fig. 4 shows the related effect on the server activation rate R_A which decreases significantly with increasing values of w. This results in less frequent overhead phases for server activations and less energy consumption.

The decrease of server activations results, however, in a performance degradation, see next Fig. 5, where performance is illustrated by means of the average delay (waiting times) of buffered requests. (*Note:* For any finite activation time, all arriving requests are buffered except the ones which are lost due to buffer overflow.) The average delays generally increase with decreasing values of α; this effect is specifically dominating for small loads; in this case, almost all arrivals require a server activation for booting (CSB) or for warm-up (HSB). The wide range of λ with almost constant or even decreasing delays is due to the relatively decreasing activation overheads between about 20–90% of loading. The expected asymptotic increase of delays when approaching the capacity limit $\lambda/\mu = n$ cannot happen due to the finite buffer capacity s (delay-loss system behavior). The relatively wide load range with almost constant delays is a specific feature of the control strategy and opens a high flexibility for server consolidation while a performance threshold (required by Service Level Agreements, SLA) can still be guaranteed.

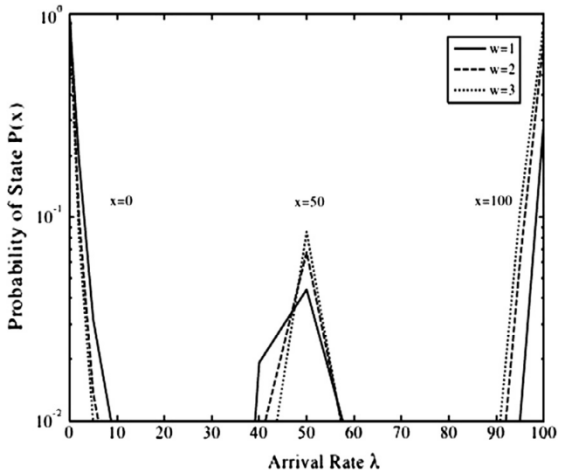

Fig. 3. Probabilities of state $P(x)$ versus arrival rate λ for $x = 1, 50$ and 100. Parameters: threshold w, busy servers x, $\alpha/\mu = 1$ (CSB).

Fig. 6 illustrates the effect of Dynamic Frequency Scaling (DFS) for the operation mode CSB on the average waiting time. There is only little influence under the current parameter setting $z^* = 2$: for the large load range, the effect is very small as throttling of the clock frequency affects only the processes in service for low queue lengths which, in turn of this, defeats server deactivation. For very small

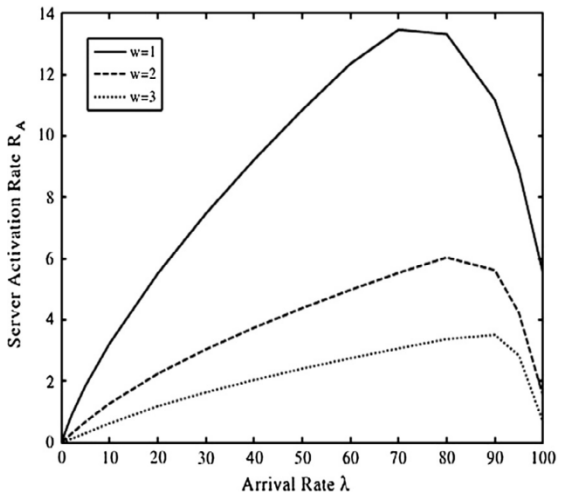

Fig. 4. Server activation rate R_A versus arrival rate λ. Parameters: threshold w, $\alpha/\mu = 1$ (CSB).

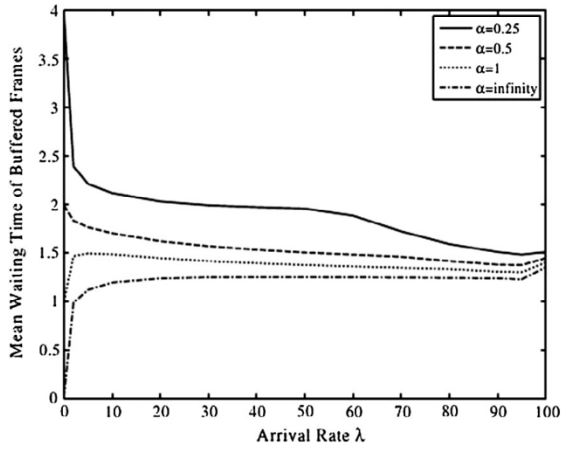

Fig. 5. Mean waiting time $E[T_W|T_W > 0]$ of buffered requests versus arrival rate λ. Parameters: server activation rate α/μ, threshold $w = 2$.

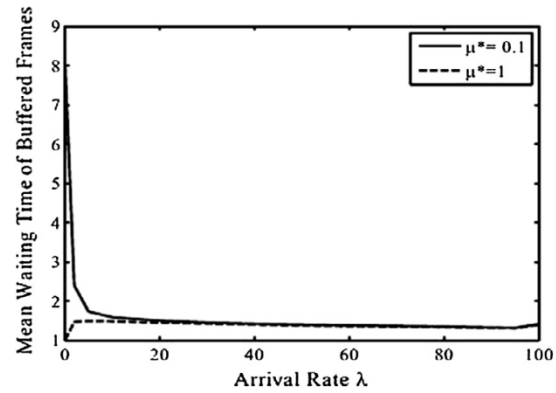

Fig. 6. Mean waiting time $E[T_W|T_W > 0]$ of buffered requests versus arrival rate λ – influence of DFS. Parameters: server activation rate $\alpha/\mu = 1.0$, CSB without/with DFS, $w = 2$, $\mu^*/\mu = 0.1$ and 1 for all x, $z^* = 2$.

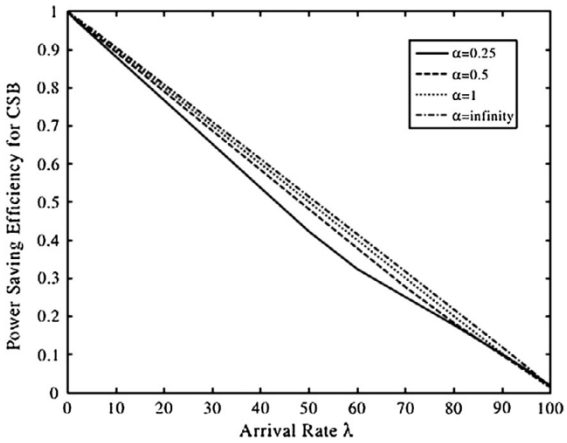

Fig. 7a. Power-saving efficiency η versus arrival rate λ. Parameters: server activation rate α/μ, operation mode CSB, threshold $w = 2$.

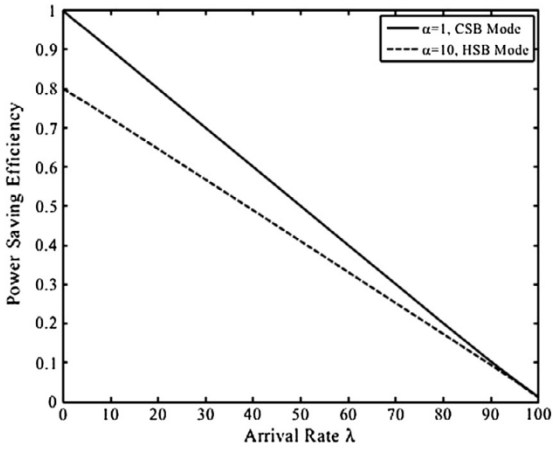

Fig. 7b. Power-saving efficiency η versus arrival rate λ – comparison of CSB and HSB. Parameters: $w = 2$, $\alpha^*/\mu = 1$ (CSB), 10 (HSB), $P^*/P_0 = 0.2$.

loads, almost all arrivals suffer from low server speed and delays increase significantly.

Fig. 7a illustrates the power-saving efficiency η for CSB dependent on the arrival rate λ and on various activation overhead times indicated by the parameter α ranging from $\alpha = 0.25$ (i.e., $1/\alpha = 4$, large activation time) to $\alpha \to \infty$ (i.e., $1/\alpha \to 0$, zero activation time). Generally, the power-saving efficiency increases slightly with decreasing values of α.

In Fig. 7b, CSB and HSB are compared. Both modes differ in α ($\alpha = 1$ for CSB and $\alpha = 10$ for HSB) and in the sleep time power consumption for HSB with power ratio $P_0^*/P_0 = 0.2$. The HSB mode is generally less energy efficient (but results in slightly shorter delays (see Fig. 5).

5. Conclusions

In this paper, a generalized and versatile multi-server queuing model for Data Center Servers with automatic activation/deactivation of servers, activation overhead under cold and hot stand-by operation, and Dynamic Voltage and Frequency Scaling (DVFS) has been proposed. The queuing model is controlled by a Finite State Machine (FSM) model. The model allows for the study of trade-offs between power efficiency and performance which are reciprocal to each other and subject to the optimization of DC resources and SLA requirements. A new and efficient recursive analysis algorithm allows for parametric studies. The exact analysis of this model is only possible under Markovian assumptions for arrival and service processes. (First simulations have shown that the principal results of the parametric influences are maintained under generalized traffic process assumptions.) The proposed model has a variety of parameters which can be adapted to a wide range of real DC parameters, such as the number of servers, power-saving operation modes, booting or warmup overhead times, and load-dependent activation/deactivation thresholds. Some first example studies have been made to illustrate the principal application of the model in a conference contribution [8]. In another two companion papers, the model has been applied to load balancing in cloud

networks [13], and to percentiles of the response time for virtualized DC server groups [15]. Further studies and experiments in a cloud laboratory at the GUC are currently in progress.

References

[1] J. Sekhar, G. Jeba, S. Durga, A survey on energy-efficient server consolidation through VM life migration, Int. J. Adv. Eng. Technol. (IJAET) 5 (1) (2012) 515–525.

[2] A. Beloglazov, R. Buyya, Y.Ch. Lee, A. Zomaya, A taxonomy and survey of energy-efficient data centers and cloud computing systems, in: M. Zollkowitz (Ed.), Advances in Computers, Elsevier, San Francisco, 2011 (51p).

[3] H. Zhang, Research on the influence of cloud computing on the virtual operation performance management, in: 7th Int. Conf. on Computer Science & Education (ICCSE), Melbourne, 2012, pp. 235–238.

[4] K. Ye et al., Virtual machine based energy-efficient data center architecture for cloud computing: a performance perspective, in: Proceedings of 2010 IEEE/ACM Int. Conf. on Green Computing and Communications, 2010, pp. 171–178.

[5] H. Khazaei, J. Misic, V.B. Misic, Performance analysis of cloud computing centers using M/G/m/m+r queuing systems, IEEE Trans. Parallel Distrib. Syst. 23 (5) (2012) 936–943.

[6] P.J. Kühn, M. Mashaly, Modeling and performance evaluation of self-adapting algorithms for the optimization of power-saving operation modes, in: Proc. 1st. Europ. Teletraffic Seminar (ETS), Poznan, Poland, February 14–16, 2011.

[7] P.J. Kühn, Systematic classification of self-adapting algorithms for power-saving operation modes of ICT systems, in: Proc. 2nd ACM Conf. on Energy-Efficient Computing and Networking" (e-Energy 2011), New York, NY, USA, May 30–June 1, 2011.

[8] P.J. Kühn, M. Mashaly, Performance of self-adapting power-saving algorithms for ICT systems, in: IFIP/IEEE Symposium on Integrated Network and Service Management (IM 2013), Ghent, May 27–30, 2013.

[9] P. Tran-Gia, Overload Problems in Stored-Program Controlled Switching Systems – Modeling and Analysis, Doctoral Dissertation, Univ. of Siegen, Germany, 1982 (in German).

[10] O.C. Ibe, J. Keilson, Multi-server threshold queues with hysteresis, J. Perform. Eval. 21 (1995) 185–213.

[11] J.C.S. Lui, L. Golubchik, Stochastic complement analysis of multi-server threshold queues with hysteresis, J. Perform. Eval. 35 (1999) 19–48.

[12] C.-F. Chou, L. Golubchik, J.C.S. Lui, Multi-class multi-server threshold-based systems: a study of non-instantaneous server activation, IEEE Trans. Parallel Distrib. Syst. 18 (1) (2007) 96–110.

[13] M. Mashaly, P.J. Kühn, Load-balancing in cloud-based content delivery networks using adaptive server activation/deactivation, in: IEEE Conf. ICET 2012, Cairo, Egypt, October 10–11, 2012.

[14] P.J. Kühn, M. Mashaly, Dynamic load balancing in cloud networks – resource management, modeling and performance, in: Proc. 2nd Conf. on Energy-Efficient Data Centers (E2DC 2013), Berkeley, CA, May 21, 2013.

[15] M. Mashaly, P.J. Kühn, Modeling and analysis of virtualized multi-service cloud data centers with automatic server consolidation and prescribed service level agreements, in: 23rd. Int. Conf. on Computer Theory and Applications (ICCTA 2013), Alexandria, Egypt, October 29–31, 2013.