The Journal of Systems and Software 85 (2012) 1400–1408

Contents lists available at SciVerse ScienceDirect

The Journal of Systems and Software

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

Comparison of scheduling schemes for on-demand IaaS requests

Tien Van Do[a,*], Csaba Rotter[b]

[a] Department of Telecommunications, Budapest University of Technology and Economics, H-1117, Magyar tudósok körútja 2., Budapest, Hungary
[b] Nokia Siemens Networks, Köztelek útca 6, Budapest, Hungary

ARTICLE INFO

Article history:
Received 3 June 2011
Received in revised form
10 November 2011
Accepted 7 January 2012
Available online 16 February 2012

Keywords:
Cloud computing
Scheduling
On-demand IaaS

ABSTRACT

Infrastructure-as-a-service (IaaS) is one of emerging powerful cloud computing services provided by IT industry at present. This paper considers the interaction aspects between on-demand requests and the allocation of virtual machines in a server farm operated by a specific infrastructure owner. We formulate an analytic performance model of the server farm taking into account the quality of service (QoS) guaranteed to users and the operational energy consumption in the server farm. We compare several scheduling algorithms from the aspect of the average energy consumption and heat emission of servers as well as the blocking probabilities of on-demand requests. Based on numerical results of a comparison of different allocation strategies, a saving on the energy consumption is possible in the operational range (where on-demand requests do not face unpleasant blocking probability) with the allocation of virtual machines to physical servers based on the priority.

1. Introduction and motivation

Cloud computing has initiated a paradigm shift in IT industry with the provision of new powerful services like infrastructure-as-a-service (IaaS), software as a service (SaaS) and platform as a service (PaaS) (Campbell and Jeronimo, 2006; Dikaiakos et al., 2009; Sotomayor et al., 2009; Takemura and Crawford, 2009; Moschakis and Karatza, 2010). Recent studies have shown that a large proportion of the largest companies worldwide uses cloud computing service to achieve new business goals and to provide more efficient services to their customers. Disaster recovery, avoidance of service outage and dynamic load balancing represent some of the most important areas of application of the rapidly evolving cloud computing concepts.

Various studies showed that cloud computing can reduce the infrastructure and IT management cost. The reason is that it substantially improves the utilization of the physical infrastructure (i.e., host machines) while it can provide the similar level of safety (compared to a solution where each application service provider obtains a separate physical server from the infrastructure owners) for application service providers (ASP) who order server capacity from infrastructure owners. Another advantage is that the infrastructure can provide different service packages concerning specific operating systems running on top of the same hardware in a flexible manner.

In order to provide services, three enabling capabilities should be directly or indirectly offered by cloud computing systems.

- computing capability is achieved with the use of advanced processors in an appropriate hardware environment.
- storage capability is established using storage architecture (note that hard disks come along with physical servers are rarely used or not used for storing clients' data).
- management capability includes functionality to efficiently manage (e.g., allocation of tasks and demands to processors) the cloud and necessary API (Application Programming Interface) for clients.

Based on these basic capabilities, operators can offer services like IaaS (where infrastructure or the actual hardware is provisioned to customers who are responsible to install operating systems and necessary softwares), PaaS (the computing infrastructure and the installed platforms/operating systems) and SaaS (PaaS and the necessary software are provisioned).

Among many problems (Berl et al., 2010) related to the management of services in data centers, the topic of virtual machine placement has intensively been researched in some earlier works (Hyser et al., 2007; Nguyen et al., 2009; Jayasinghe et al., 2011; Xu and Fortes, 2010; Tsakalozos et al., 2011; Moschakis and Karatza, 2010; Do, 2011b). Hyser et al. (2007) reported the design of a controller that automatically assigns virtual machines to physical

* Corresponding author.
E-mail address: do@hit.bme.hu (T.V. Do).

0164-1212/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jss.2012.01.019

hosts. Later, Nguyen et al. (2009) proposed a detailed architecture to control the dynamic placement of virtual machines. Jayasinghe et al. (2011) considered the structural constraint-aware virtual machine placement and presented an algorithm to improve performance and availability of services on IaaS clouds. Xu and Fortes (2010) formulated the VM placement issue as a multi-objective optimization problem and proposed an improved genetic algorithm with fuzzy multi-objective evaluation for efficiently searching a solution. Tsakalozos et al. (2011) considered the problem of instantiating entire virtual infrastructures in large non-homogeneous IaaS clouds. Moschakis and Karatza (2010) investigated the performance and the overall cost of two major gang scheduling algorithms for cloud computing with simulation.

However, none of these works considered the interaction aspects between on-demand requests and the allocation of virtual machines in a server farm operated by a specific infrastructure owner (an example of such a scenario is the IaaS service offering by Amazon's Elastic Compute Cloud, RackSpace, GoGrid, Joyent, and Terremark), taking into account various aspects such as the performance measures of blocking probability of heterogeneous requests, the energy consumption and the heat emission of physical servers.

This paper is on the modeling of the interaction between on-demand requests and the allocation of virtual machines in a server farm operated by its infrastructure owner. We investigate schemes for the allocation of virtual machines taking into account the energy consumption because the energy use of servers significantly contributes to the business cost (Berl et al., 2010). To reduce the energy consumption either the scheme of dynamic voltage and frequency scaling (DVFS) depending on the load can be applied in a physical machine, or physical servers can be switched to a low-energy consuming state when no virtual machines are allocated in that physical server (Lefurgy et al., 2003). However, the application of the DVFS scheme is not assumed in this paper because of the following reasons. First, a study (Sueur and Heiser, 2010) revealed that in the latest generations of processors an idle state has clearly more impact on energy savings in servers, and dynamic voltage and frequency scaling has only a minimal impact on the reduction of energy consumption. Second, operators provisioning IaaS have a limited scope to change the CPU frequency of processors since computing capabilities explicitly expressed in equivalent CPU frequency are normally offered to customers in practice. Therefore, the viable method to reduce the energy consumption is the increase of the utilization of server farms with switching off unused servers (Mazzucco and Dyachuk, 2012).

In this paper, we present a queueing model for the proposed operation policy incorporating energy-aware allocation schemes. Based on numerical results of a comparison of different allocation strategies, we can conclude that a saving on the energy consumption is possible in the operational range (where on-demand requests do not face unpleasant blocking) if an appropriate allocation scheme is applied.

The rest of this paper is organized as follows. In Section 2 we provide an overview of management issues related to the reduction of the energy consumption. In Section 3 we present our modeling approach of a server farm. The allocation strategies of the virtual servers are considered and modeled in terms of a queueing model. Numerical results of a comparison of these allocation strategies are discussed for different hardware configurations in Section 4. Finally, Section 5 provides some conclusions.

2. Management issues to provide IaaS by a server farm

2.1. Quality of service issues in the provisioning of IaaS

Virtualization constitutes the enabling technology to realize the provision of infrastructure-as-a-service (IaaS). It is interpreted as one type of service offered by cloud computing besides software as a service (SaaS) and platform as a service (Platform). To provide IaaS, infrastructure owners typically run their server farm consisting of physical servers (therefore, called hardware infrastructure owners) and apply an appropriate virtualization technology (such as Xen) and management software to deploy and offer virtual machines to dynamic clients. Generally, three different roles are identified: users/applications, application service providers and owners of hardware infrastructure (see Fig. 1). Applications and related services are provided by application service providers who require virtual machines from an infrastructure owner to run their application servers.

A Service Level Agreement (SLA) often includes a service commitment in terms of the availability of a service (i.e., the percentage of the expected uptime during the service year) that an infrastructure owner guarantees to the users in their contracts. We assume that operators provisioning IaaS promise to provide virtual machines with computing capabilities explicitly expressed in equivalent CPU frequency.

Another QoS measure that is not often specified in the service contract is given by the blocking probability of on-demand requests. Some traditional service operators such as telephone and mobile network operators keep the ratio of blocked calls less than 1%. The blocking probability is especially important in a competitive market environment. A high value of the blocking probability of on-demand requests leads to unsatisfied customers. It may cause a loss of profit of the specific operator because unsatisfied customers may likely send the blocked request and any further requests to another infrastructure owner. Therefore, the capacity of a server farm must be engineered to achieve a low blocking probability.

2.2. Scheduling on-demand requests

In this paper we focus on an energy-aware management procedure of a server farm. The first thought that comes to our mind is that physical servers should be switched off when there is no load (i.e., virtual servers) in the machine. However, the booting time may take several minutes. It may be too high to react to requests on demand. Another idea is that a computer-system can be switched into low energy-consuming states when there is no load. Scaling down CPU frequencies of physical hosts and voltage of CPUs can lead to savings of the energy consumption (Olsen and Morrow, 2003; de Langen and Juurlink, 2007; Pallipadi and Siddha, 2007; Schönherr et al., 2010). However, the reduction of the CPU frequencies of physical hosts and the voltage of CPUs may cause a longer run time of the applications compared to the normal case, which may not result in savings. Recently, Sueur and Heiser (2010) have reported that the idle state has more impact on energy savings in servers with the latest generations of processors and dynamic voltage and frequency scaling has a minimal impact on the reduction of energy consumption with the latest generation of processors.

In this paper we utilize the fact that a physical server can be switched into a standby state where the actual state of the machine is saved into memory with low-energy consumption (i.e., RAM remains powered) and the processor is switched off (no power consumption). The saved information allows the switching of the physical host from the standby state into the state of full functionality in a much shorter time than the booting time. Therefore, we propose an operation procedure that automatically switches hosts to the standby state when no virtual machines are allocated in a specific physical server and manages a physical host into the operating state of full functionality without human intervention when virtual servers are allocated. Hence, the energy consumption of a server farm largely depends on the number of physical hosts which are not in the standby state. Note that a machine can be easily switched to the standby station using the capability of

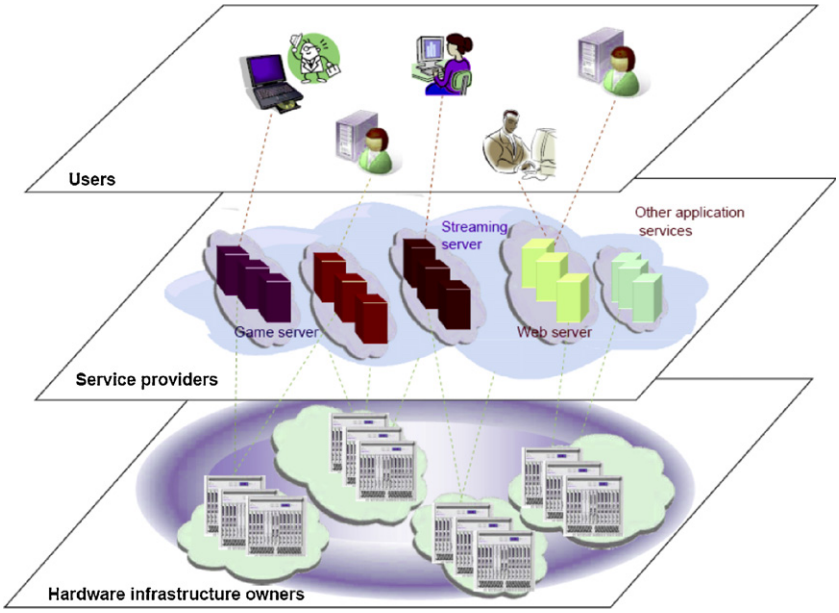


Fig. 1. Utility computing environment based on virtual machines.

the Advanced Configuration and Power Interface (ACPI) subsystem (http://www.acpi.info/spec.htm). The system wake-up can be initiated using the Wake-on-LAN capability.

Without compromising the QoS, an appropriate algorithm should be applied to allocate arriving requests for virtual machines to a physical host in a server farm because the random allocation of virtual machines to physical hosts is not an energy-aware solution. Regarding the allocation of virtual servers we investigate in this paper three different schemes (note that the schemes only consider servers that have enough free capacity taking into account the already allocated capacity upon the arrival of an on-demand request):

- The most free capacity (MF) scheme assigns the request for a virtual server to a physical server which has the largest free CPU resource.
- The least free capacity (LF) scheme chooses a physical server which has the smallest free CPU resource.
- Physical servers are sorted according to a certain criteria. We assume that the server with the highest index has the largest priority. Upon an arrival of a request, the priority (PR) scheme selects a server with the largest priority among the considered servers.

Both MF and LF choose the physical host with the largest index if there are several servers having a similar most/least free CPU capacity.

3. An analytic model

3.1. Assumptions and notations

We consider a server farm with L physical servers. Each physical host has the resource (C_l, M_l), $l = 1, \ldots, L$, where C_l represents the number of normalized CPU (Central Processing Unit) capacity units[1] and M_l denotes the amount of memory that is available for the allocation of virtual servers. The energy consumption and the heat emission is denoted by W_l and H_l for a physical server $l \in \{1, \ldots, L\}$, respectively. Assume that the total amount D of storage capacity is available.

In the server farm N types of virtual machines are offered to clients. A virtual machine of type $n \in \{1, \ldots, N\}$ is allocated c_n normalized CPU capacity units, m_n memory (e.g., measured in megabytes) and d_n disk capacity. We assume that $c_1 \le c_2 \le \cdots \le c_N$. Let λ_n, $n = 1, \ldots, N$, denote the arrival rate of clients who request virtual machines of type n with exponentially distributed holding times of mean $1/\mu_n$.

Let $X_{l,n}(t)$, $l = 1, \ldots, L$, $n = 1, \ldots, N$, be the number of type-n virtual machines which are allocated to clients on a physical server l at time instant t. We can write the equation concerning the limitation of the resources of a physical server $l \in \{1, \ldots, L\}$ as follows:

$$\begin{aligned} &\sum_{n=1}^{N} x_{l,n} c_n \le C_l, \\ &\sum_{n=1}^{N} x_{l,n} m_n \le M_l, \\ &\sum_{l=1}^{L} \sum_{n=1}^{N} x_{l,n} d_n \le D. \end{aligned} \tag{1}$$

3.2. The analytic model

The system is described by the continuous time Markov chain (CTMC) $\Upsilon = (\mathbf{X}_1(t), \ldots, \mathbf{X}_L(t))$, where $\mathbf{X}_l(t) = (X_{l,1}(t), \ldots, X_{l,N}(t))$. The state space Θ of CTMC Υ can be defined based on Eq. (1) by

$$\Theta = \{\mathbf{x} : 0 \le x_{l,n} \in \mathrm{IN} \text{ s.t.}(1) \text{ holdsfor } l = 1, \ldots, L; \ n = 1, \ldots, N\},$$

where the vectors $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_l, \ldots, \mathbf{x}_L)$ and $\mathbf{x}_l = (x_{l,1}, \ldots, x_{l,N})$, $l = 1, \ldots, L$, contain the specific values of $(\mathbf{X}_1(t), \ldots, \mathbf{X}_L(t))$ and $\mathbf{X}_l(t) = (X_{l,1}(t), \ldots, X_{l,N}(t))$, respectively.

The steady state probabilities are denoted by

$$p_{\mathbf{x}} = \lim_{t \to \infty} Pr(\mathbf{X}_l(t) = \mathbf{x}_1, \ldots, \mathbf{X}_L(t) = \mathbf{x}_L).$$

[1] E.g., in Amazon's Elastic Compute Cloud it is called an EC2 Compute Unit (see http://aws.amazon.com/ec2/instance-types).

Let us define the following vectors of integers

$$\mathbf{x}_{l,n}^{-} = \{x_{l,1}, \ldots, x_{l,n} - 1, \ldots, x_{l,N}\},$$
$$\mathbf{x}_{l,n}^{+} = \{x_{l,1}, \ldots, x_{l,n} + 1, \ldots, x_{l,N}\}, \quad l = 1, \ldots, L; \quad n = 1, \ldots, N.$$

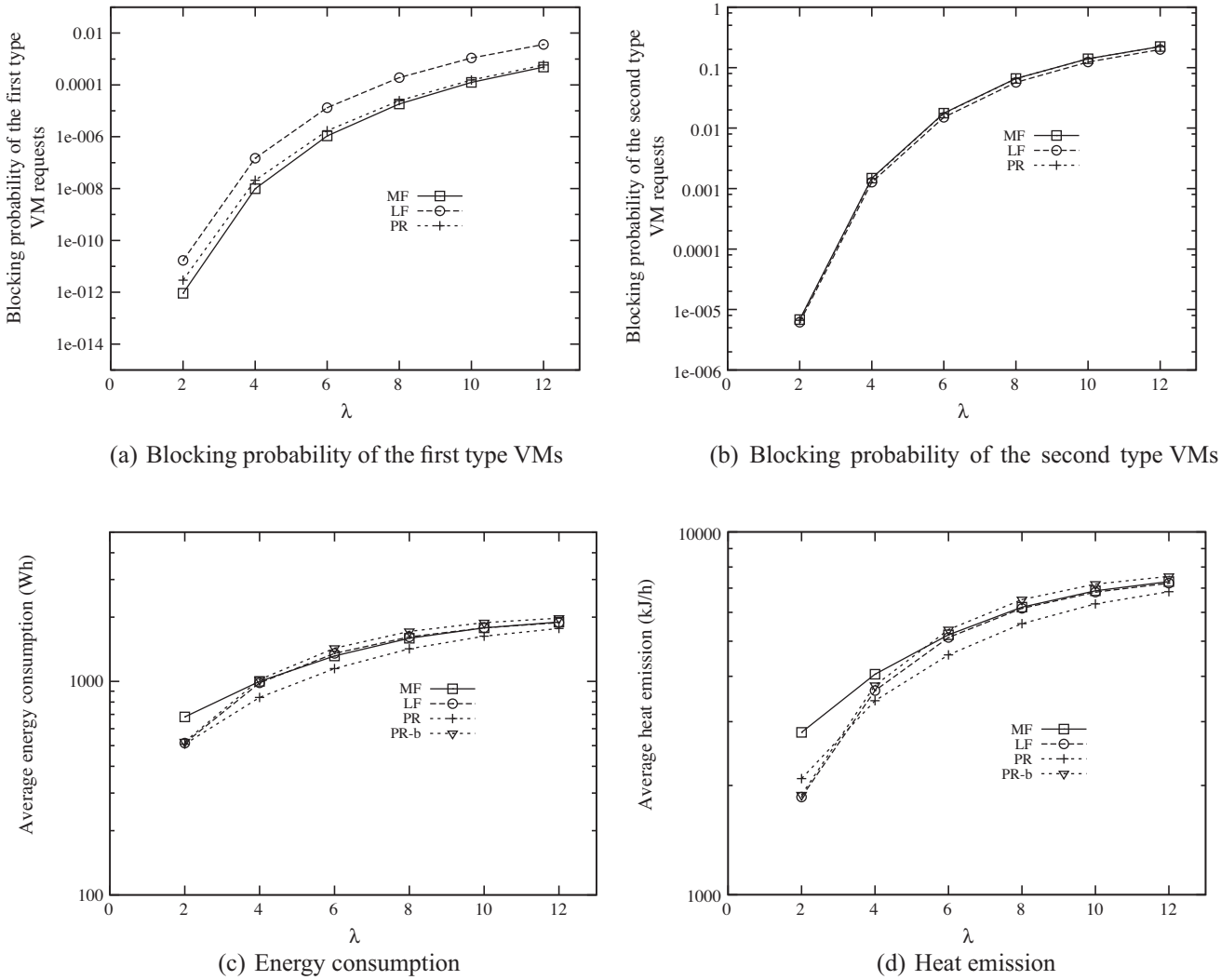
There are two types of transitions between the states of the Markov chain Υ: a first type of transitions due to the arrival of requests and Upon the arrival of a specific request, the decision to allocate the request to machine m depends on the specific allocation scheme, thus the allocation initiates a transition.

The arrival of a VM of type-n when the system in state $\mathbf{x} \in \Theta$ corresponds to the transition from state $(\mathbf{x}_1, \ldots, \mathbf{x}_l, \ldots, \mathbf{x}_L)$ to state $(\mathbf{x}_1, \ldots, \mathbf{x}_{l,n}^{+}, \ldots, \mathbf{x}_L)$, $l = 1, \ldots, L$, with rate λ_n

- if $(\mathbf{x}_1, \ldots, \mathbf{x}_{l,n}^{+}, \ldots, \mathbf{x}_L) \in \Theta$ and $C_l - \sum_{n=1}^{N} x_{l,n} c_n$ is the largest among $C_k - \sum_{n=1}^{N} x_{k,n} c_n$, $k = 1, \ldots, L$, when the most free capacity (MF) scheme is applied. Note that l is the largest index in cases there are several servers having the same largest free CPU capacity equal to $C_l - \sum_{n=1}^{N} x_{l,n} c_n$.
- if $(\mathbf{x}_1, \ldots, \mathbf{x}_{l,n}^{+}, \ldots, \mathbf{x}_L) \in \Theta$ and $C_l - \sum_{n=1}^{N} x_{l,n} c_n$ is the smallest among $C_k - \sum_{n=1}^{N} x_{k,n} c_n$, $k = 1, \ldots, L$, when the least free capacity (LF) scheme is applied. Note that l is the largest index in cases there are several servers having the same smallest free CPU capacity equal to $C_l - \sum_{n=1}^{N} x_{l,n} c_n$.
- if $(\mathbf{x}_1, \ldots, \mathbf{x}_{l,n}^{+}, \ldots, \mathbf{x}_L) \in \Theta$ and $(\mathbf{x}_1, \ldots, \mathbf{x}_{k,n}^{+}, \ldots, \mathbf{x}_L) \notin \Theta \quad \forall k = l + 1, \ldots, L$.

A customer assigned a VM of type-n, $1 \le n \le N$ in host l, $l \le l \le L$, relinquish the VM usage, which corresponds to the following transition from state $(\mathbf{x}_1, \ldots, \mathbf{x}_l, \ldots, \mathbf{x}_L) \in \Theta$ to state $(\mathbf{x}_1, \ldots, \mathbf{x}_{l,n}^{-}, \ldots, \mathbf{x}_L)$ with rate $x_{l,n}\mu_n$ if $(\mathbf{x}_1, \ldots, \mathbf{x}_{l,n}^{-}, \ldots, \mathbf{x}_L) \in \Theta$.

The transition rate matrix can be easily set up with the above rules if the state space Θ is lexicographically ordered and the states are numbered according the lexicographical order. Then, using the standard technique (Takacs, 1962) the steady state probabilities $p_{\mathbf{x}}$ can be obtained. Note that the stationary distribution always exists because the state space of the CTMC Υ is finite. In addition, advanced techniques (Fujita et al., 1997; Hermanns et al., 1999; Kwiatkowska et al., 2004, 2010; Wimmer et al., 2010) can also be applied to exploit the sparseness of the matrices and to cope with the problem of state space explosion. We can also use the probabilistic symbolic model checking PRISM tool (Kwiatkowska et al., 2004) to compute the stationary probabilities and performance measures.



(a) Blocking probability of the first type VMs

(b) Blocking probability of the second type VMs

(c) Energy consumption

(d) Heat emission

Fig. 2. A server farm with 6 physical servers, $\lambda_1 = 0.2\lambda$, $\lambda_2 = 0.8\lambda$. (a) Blocking probability of the first type VMs. (b) Blocking probability of the second type VMs. (c) Energy consumption. (d) Heat emission.

3.3. Performance measures

Several basic performance measures are defined as follows:

- the blocking probability of requests for virtual machines of type k is expressed as

$$B_k = \sum_{\mathbf{x} \in \Phi_k} p_{\mathbf{x}}, \tag{2}$$

where the subset Φ_k of Θ contains states such that no allocation of an arriving virtual machine of type k is possible (i.e., Eq. (1) is violated if we add one additional virtual machine of type k).

- the average number of busy physical servers is given by

$$E_p = \sum_{j=1}^{L} jPr\left(\sum_{l=1}^{L} \mathbf{1}_{>0}\left(\sum_{n=1}^{N} x_{l,n}\right) = j\right), \tag{3}$$

where $\mathbf{1}_{>0}$ is the indicator function

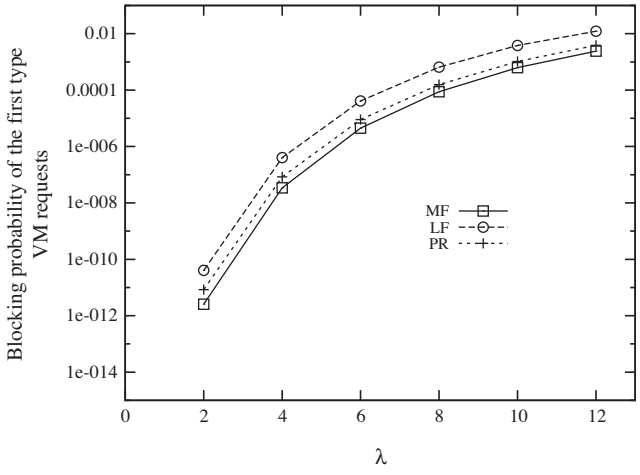
$$\mathbf{1}_{>0}(y) = \begin{cases} 1 & \text{if } y > 0 \\ 0 & \text{if } y = 0 \end{cases}.$$

- the average energy consumption (EG) and the heat emission (HE) are expressed by

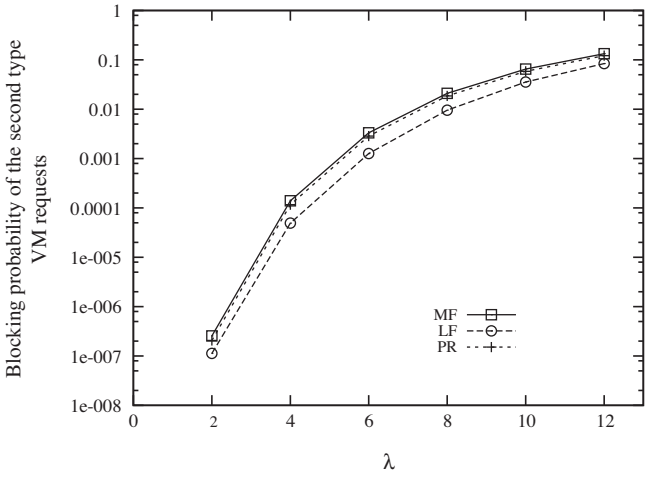
$$\mathrm{EG} = \sum_{j=1}^{L} W_j \sum_{\substack{\mathbf{x} \in \Theta \\ \sum_{k=1}^{N} x_{j,k} > 0}} p_{\mathbf{x}}, \tag{4}$$

$$\mathrm{HE} = \sum_{j=1}^{L} H_j \sum_{\substack{\mathbf{x} \in \Theta \\ \sum_{k=1}^{N} x_{j,k} > 0}} p_{\mathbf{x}}. \tag{5}$$

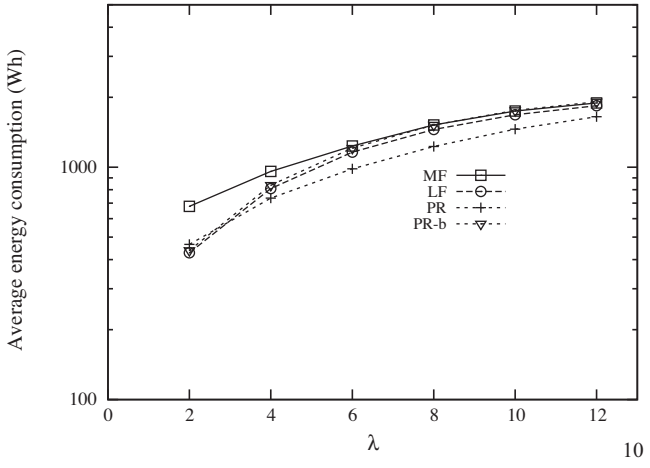
It worth emphasizing that the maximum heat emission (which can be easily obtained) is important to dimension the cooling and air-conditioning of the server rooms. The average energy consumption allows us to determine the overall energy cost arising from the operation a server farm.



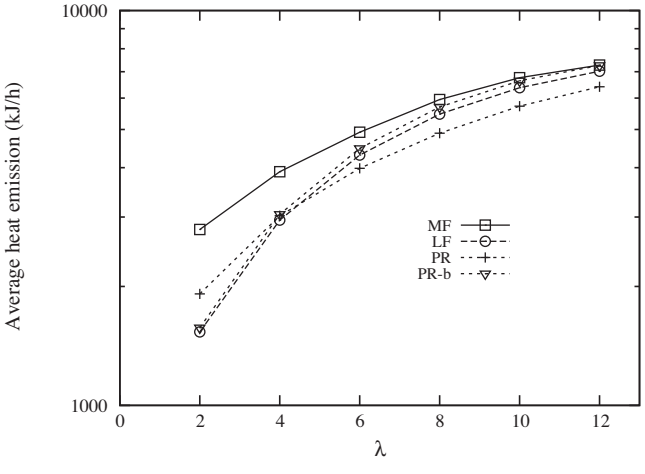
(a) Blocking probability of the first type VMs



(b) Blocking probability of the second type VMs



(c) Energy consumption



(d) Heat emission

Fig. 3. A server farm with 6 physical servers, $\lambda_1 = 0.5\lambda$, $\lambda_2 = 0.5\lambda$. (a) Blocking probability of the first type VMs. (b) Blocking probability of the second type VMs. (c) Energy consumption. (d) Heat emission.

4. Numerical results

4.1. Hardware configurations

To investigate and illustrate the capability of the proposed scheduling schemes to reduce the energy consumption, we perform a numerical study. In our example a server farm of six physical servers offers two types of virtual machines (each will get 2 Gbyte storage capacity):

- the first virtual machine type provides 2 GB memory and 1 Core with the equivalent CPU capacity of one core of 2.4 GHz Intel Xeon Processor X3430.
- the second virtual machine type receives the allocation of 4 GB memory and the CPU capacity equivalent to the CPU capacity of 2 Cores of 2.4 GHz Intel Xeon Processor X3430.

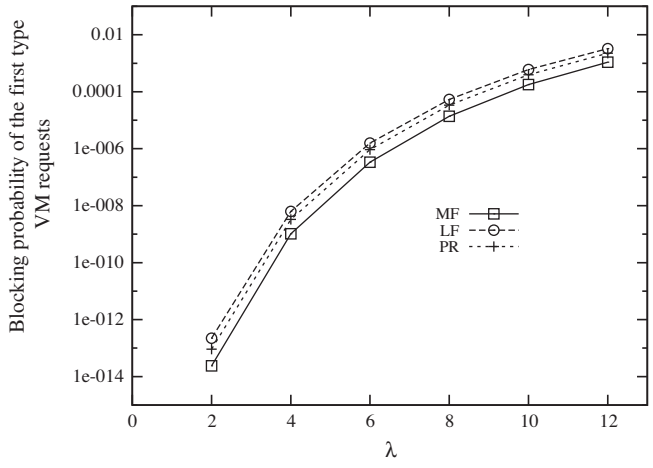
We assume that the server farm is built from the following types of low-cost commercial off-the-shelf (COTS) servers (the information is taken from the data sheets of commercial servers):

- A server of type 1 has one CPU slot configured with one 2.4 GHz Quad-Core Intel Xeon Processor 3430 and 16 Gbyte memory. The energy consumption of the first server type is 290 W h. The heat emission of the second type server is 1036.8 kJ/h.
- A server of type 2 equipped with a motherboard including two CPU slots. Each slot is configured with one 2.4 GHz Quad-Core Intel Xeon Processor E5620. The server is configured with 64 Gbyte memory. The energy consumption and the heat emission is 480 W h and 1976.4 kJ/h.
- A server of type 3 has one CPU slot configured with one Dual-Core Intel Processor (each core provides the equivalent capacity of one core of 2.4 GHz Intel Xeon Processor X3430) and 8 Gbyte memory. The energy consumption of the third server type is 120 W h, while the heat emission is 432.0 kJ/h.

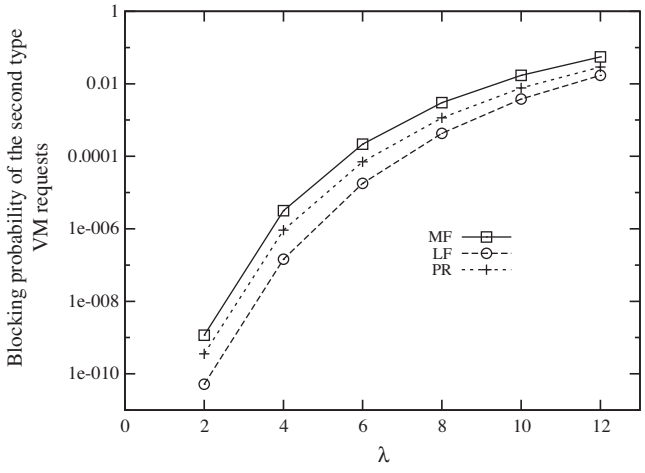
Furthermore, it is assumed that there an advanced storage architecture is applied in a server farm.

4.2. Numerical results and discussion

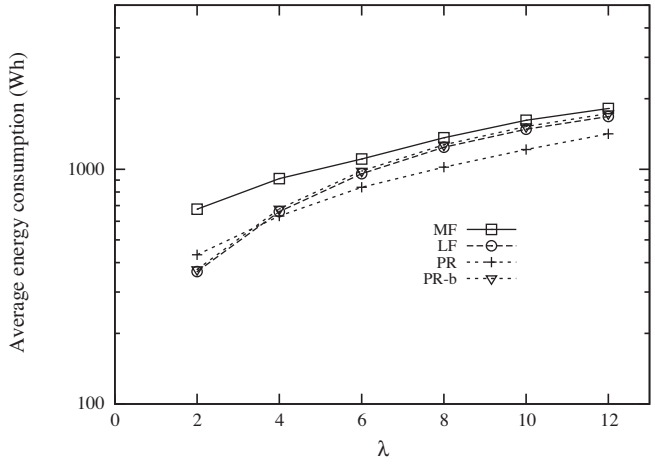
Several hardware configurations and allocation schemes are investigated in this section. When several virtual machines are



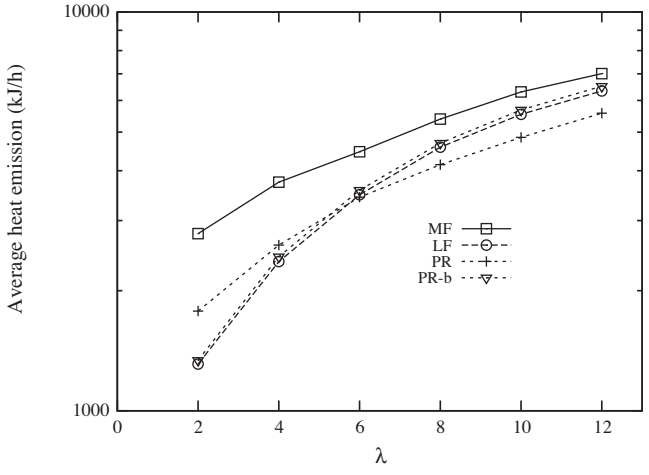
(a) Blocking probability of the first type VMs



(b) Blocking probability of the second type VMs



(c) Energy consumption



(d) Heat emission

Fig. 4. A server farm with 6 physical servers, $\lambda_1 = 0.8\lambda$, $\lambda_2 = 0.2\lambda$. (a) Blocking probability of the first type VMs. (b) Blocking probability of the second type VMs. (c) Energy consumption. (d) Heat emission.

hosted it is normally assumed that one core of servers is reserved for the host operating system or a hypervisor under which the guest systems are installed.

- In the first configuration we consider six physical servers (which can be put into one rack): server number 1, 2, 3 and 4 are of the first server type, and the last two belong to the second server type.
- In the second configuration server 1 and 2 are from the second server type, while server number 3, 4, 5 and 6 belong to the first server type.

Therefore, the total energy consumption is 2120 W h if all servers are switched on. Regarding the normalized values of parameters $\mu_1 = \mu_2 = 1$, $\lambda_1 = p_1\lambda$, $\lambda_2 = (1 - p_1)\lambda$, we plot the blocking probabilities, the average energy consumption and the average heat emission vs. the request rate λ in Figs. 2–4. Based on the specified QoS objectives and these numerical results, we can determine the appropriate range of the arrival rate (i.e., the level of λ) where a QoS requirement can be met. As expected arrivals for the second type virtual machines face higher blocking probabilities since it requires more resources than the first type of machines. It is worth mentioning that some techniques (Do, 2011a) from mobile cellular network operators can be applied to balance or equalize the blocking probabilities.

It is also observed that LF is the best allocation policy for small load. From the point of the view of energy consumption and heat emission PR has a very favorable characteristic in the range of medium load. Compared to the peak energy consumption of 2120 W h, the saving is noticeable. However, physical servers must be ordered in a good way, i.e. the server with the largest resources should have the largest priority. In Figs. 2–4, the plots denoted by PR-b concerning the average energy and the average heat emission of the second configuration[2] with the PR scheme confirms this statement.

In the third configuration we add two additional physical servers of type 3 (i.e., server number 1 and 2 are from the third type, server number 3, 4, 5 and 6 belong to the first server type and server 7 and 8 to the second server type). The results concerning the third configuration are plotted in Figs. 5 and 6. They confirm the observation as one can conclude from the first and the second configuration.

[2] The second configuration is built in a similar manner as the first configuration, but the physical servers are ordered in a different way.

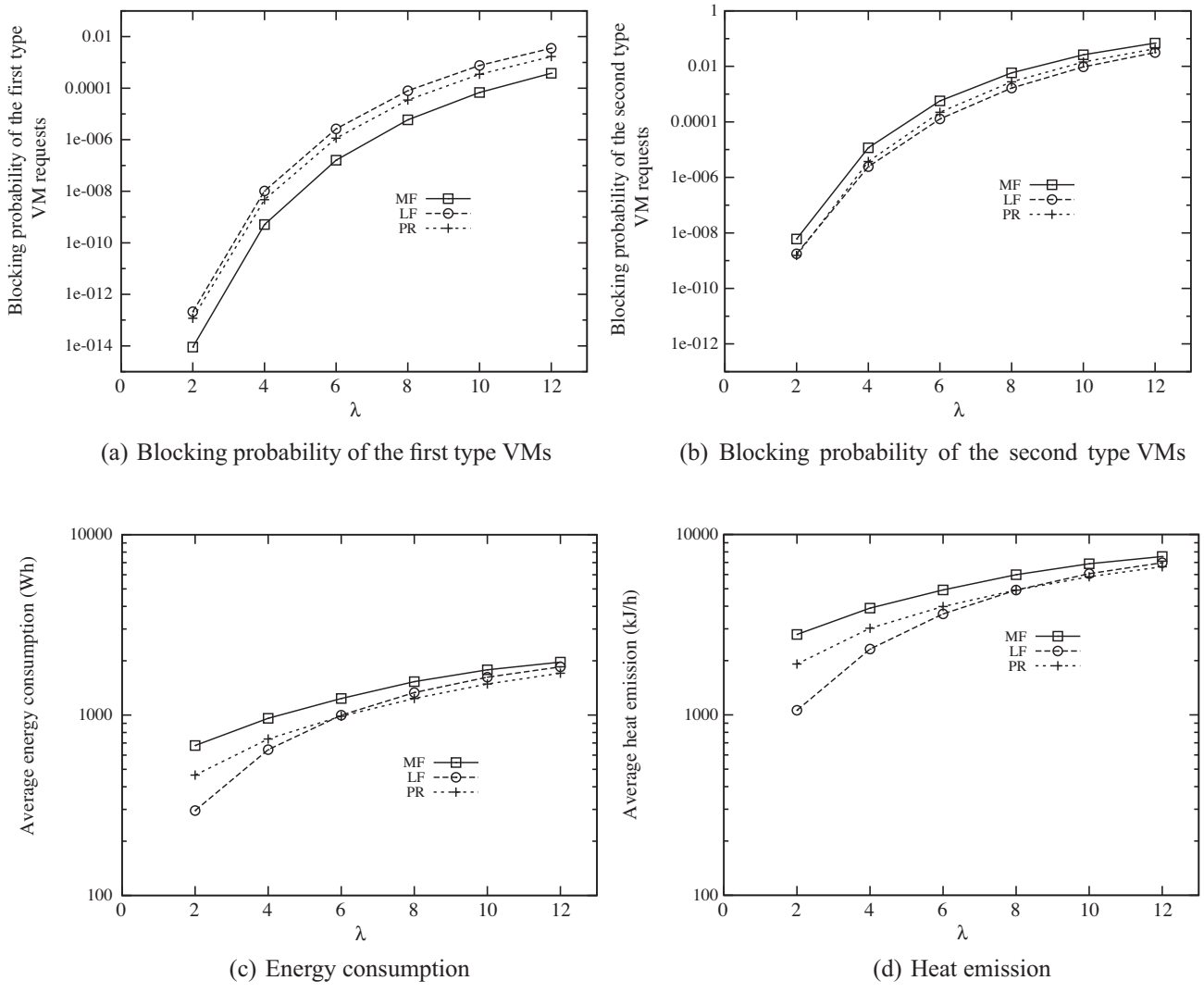


Fig. 5. A server farm with 8 physical servers (the third configuration), $\lambda_1 = 0.5\lambda$, $\lambda_2 = 0.5\lambda$. (a) Blocking probability of the first type VMs. (b) Blocking probability of the second type VMs. (c) Energy consumption. (d) Heat emission.

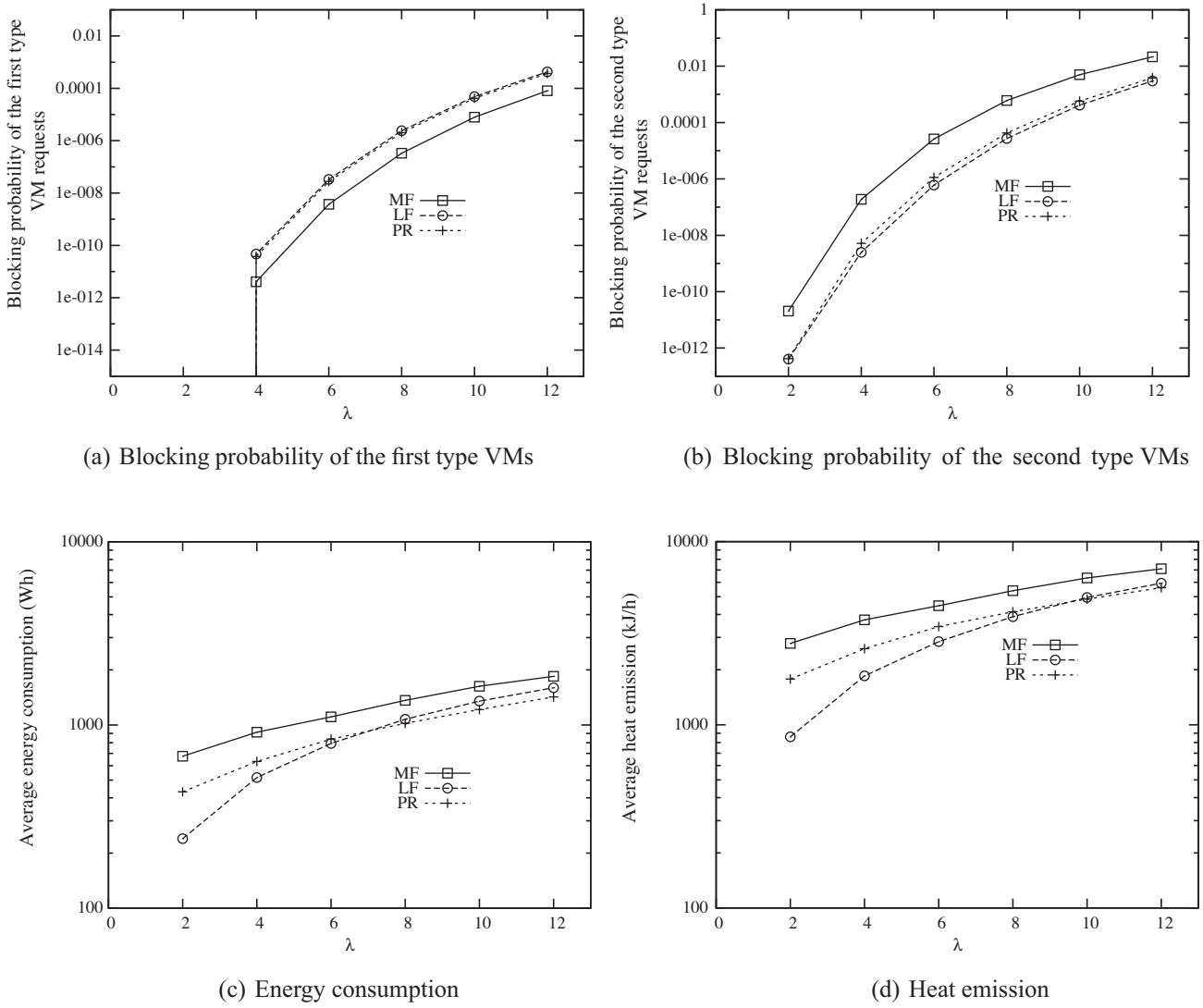


Fig. 6. A server farm with 8 physical servers (the third configuration), $\lambda_1 = 0.8\lambda$, $\lambda_2 = 0.2\lambda$. (a) Blocking probability of the first type VMs. (b) Blocking probability of the second type VMs. (c) Energy consumption. (d) Heat emission.

4.3. A method to cope with the computational complexity

The size of the state space of the proposed model depends on the number of physical servers, the parameters of physical servers and the number and parameters of virtual machine types. To deal with the state explosion problem that we may face in practical cases, we can proceed as follows.

Physical servers in a large server farm are normally installed into racks (Jayasinghe et al., 2011) that are fixed in the floor of the server rooms. The power and network cabling should be hierarchically organized according to racks. Therefore, it is reasonable to schedule load and manage physical servers in accordance with the hierarchical organization of racks. Besides the hierarchical organization of power and network cabling, the priority scheme has been shown to be acceptable from the aspect of the energy consumption. This means, the priority scheme can be analyzed by using the stochastic modeling techniques of overflow loss queues (Glabowski et al., 2007, 2008, 2010; Huang et al., 2009; Iversen, 2005) developed for telecommunication systems with multi-service overflow traffic. It is worth emphasizing that the stochastic modeling of overflow loss queues is applied to determine the revenue-maximizing number of servers to be allocated for two classes of customers in a server farm (Mazzucco and Dumas, 2011).

5. Conclusions

We have formulated an analytic performance model of a server farm to investigate realistic scenarios regarding the provision of infrastructure-as-a-service with on-demand requests for virtual servers. We can obtain important performance measures such as the blocking probabilities, the average energy consumption as well as the heat emission. Based on numerical results of a comparison of different allocation strategies, we can conclude that a saving on the energy consumption is possible in the operational range (where on-demand requests do not face unpleasant blocking) if an appropriate allocation scheme is applied. Furthermore, we have identified that the savings of the energy consumption can be achieved by the policies LF and PR depending on the load conditions of the server system.

Acknowledgments

This work is connected to the scientific program of the "Development of quality-oriented and harmonized R+D+I strategy and functional model at BME" project. This project is supported by the New Széchenyi Plan (Project ID: TÁMOP-4.2.1/B-09/1/KMR-2010-0002).

The authors are grateful for Prof. Udo R. Krieger (Faculty Information Systems and Applied Computer Sciences, Otto-Friedrich-Universität, Bamberg, Germany) for interesting discussions on the topic.

References

Berl, A., Gelenbe, E., di Girolamo, M., Giuliani, G., de Meer, H., Dang, M.Q., Pentikousis, K., 2010. Energy-efficient cloud computing. The Computer Journal 53 (7), 1045–1051.

Campbell, S., Jeronimo, M., 2006. Applied Virtualization Technology: Usage Models for IT Professionals and Software Developers. Intel Press.

de Langen, P., Juurlink, B.,2007. Trade-offs between voltage scaling and processor shutdown for low-energy embedded multiprocessors. In: SAMOS'07: Proceedings of the 7th International Conference on Embedded Computer Systems. Springer-Verlag, Berlin, Heidelberg, pp. 75–85.

Dikaiakos, M.M., Katsaros, D., Mehra, P., Pallis, G., Vakali, A., 2009. Cloud computing: distributed Internet computing for IT and scientific research. IEEE Internet Computing 13 (5), 10–13.

Do, T.V., 2011a. Solution for a retrial queueing problem in cellular networks with the fractional guard channel policy. Mathematical and Computer Modelling 53 (11–12), 2058–2065.

Do, T.V., 2011b. Comparison of allocation schemes for virtual machines in energy-aware server farms. Computer Journal 54 (11), 1790–1797.

Fujita, M., McGeer, P.C., Yang, J.C.-Y., 1997. Multi-terminal binary decision diagrams: an efficient datastructure for matrix representation. Formal Methods in System Design 10 (2–3), 149–169.

Glabowski, M., Kubasik, K., Stasiak, M., 2007. Modeling of systems with overflow multi-rate traffic. In: Third Advanced International Conference on Telecommunications (AICT 2007), May 13–19, 2007, Mauritius, p. 6.

Glabowski, M., Kubasik, K., Stasiak, M., 2008. Modeling of systems with overflow multi-rate traffic. Telecommunication Systems 37 (1–3), 85–96.

Glabowski, M., Kaliszan, A., Stasiak, M., 2010. Modeling product-form state-dependent systems with bpp traffic. Performance Evaluation 67 (3), 174–197.

Hermanns, H., Meyer-Kayser, J., Siegle, M., 1999. Multi terminal binary decision diagrams to represent and analyse continuous time Markov chains. In: Plateau, B., Stewart, W., Silva, M. (Eds.), Proceedings of 3rd International Workshop on Numerical Solution of Markov Chains (NSMC'99). Prensas Universitarias de Zaragoza, pp. 188–207.

Huang, Q., Ko, K.-T., Iversen, V.B.,2009. Performance modeling for heterogeneous wireless networks with multiservice overflow traffic. In: Proceedings of the 28th IEEE Conference on Global Telecommunications, GLOBECOM'09. IEEE Press, Piscataway, NJ, USA, pp. 3676–3682.

Hyser, R.G.C., McKee, B., Watson, B.J., 2007. Autonomic virtual machine placement in the data center. Technical Report HPL-2007-189, HP.

Iversen, V.B., 2005. Teletraffic Engineering. ITU-D Study Group 2 Question 16/2, Geneva.

Jayasinghe, D., Pu, C., Eilam, T., Steinder, M., Whally, I., Snible, E., 2011. Improving performance and availability of services hosted on iaas clouds with structural constraint-aware virtual machine placement. In: 2011 IEEE International Conference on Services Computing (SCC), July, pp. 72–79.

Kwiatkowska, M., Norman, G., Parker, D., 2004. Probabilistic symbolic model checking with PRISM: a hybrid approach. International Journal on Software Tools for Technology Transfer (STTT) 6 (2), 128–142.

Kwiatkowska, M., Norman, G., Parker, D., 2010. Advances and challenges of probabilistic model checking. In: Proceedings of 48th Annual Allerton Conference on Communication, Control and Computing. IEEE, Allerton, IL, pp. 1691–1698.

Lefurgy, C., Rajamani, K., Rawson, F., Felter, W., Kistler, M., Keller, T.W., 2003. Energy management for commercial servers. Computer 36 (December), 39–48.

Mazzucco, M., Dumas, M., 2011. Reserved or on-demand instances? A revenue maximization model for cloud providers. In: IEEE International Conference on Cloud Computing, CLOUD 2011, Washington, DC, USA, 4–9 July, pp. 428–435.

Mazzucco, M., Dyachuk, D., 2012. Balancing electricity bill and performance in server farms with setup costs. Future Generation Computer Systems 28 (2), 415–426.

Moschakis, I., Karatza, H., 2010. Evaluation of gang scheduling performance and cost in a cloud computing system. The Journal of Supercomputing, 1–18, doi:10.1007/s11227-010-0481-4.

Nguyen, V.H., Dang Tran, F., Menaud, J.-M.,2009. Autonomic virtual resource management for service hosting platforms. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD '09. IEEE Computer Society, Washington, DC, USA, pp. 1–8.

Olsen, C.M., Morrow, L.A.,2003. Multi-processor computer system having low power consumption. In: PACS'02: Proceedings of the 2nd International Conference on Power-aware Computer Systems. Springer-Verlag, Berlin, Heidelberg, pp. 53–67.

Pallipadi, V., Siddha, S.B., 2007. Processor power management features and process scheduler: do we need to tie them together? In: LinuxConf Europe, pp. 1–8.

Schönherr, J.H., Richling, J., Werner, M., Mühl, G.,2010. Event-driven processor power management. In: e-Energy '10: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking. ACM, New York, NY, USA, pp. 61–70.

Sotomayor, B., Montero, R.S., Llorente, I.M., Foster, I., 2009. Virtual infrastructure management in private and hybrid clouds. IEEE Internet Computing 13 (5), 14–22.

Sueur, E., Heiser, G., 2010. Dynamic voltage and frequency scaling: the laws of diminishing returns. In: Proceedings of the 2010 Workshop on Power Aware Computing and Systems (HotPower'10), Vancouver, Canada, October, pp. 1–5.

Takacs, L., 1962. Introduction to the Theory of Queues. Oxford University Press, New York.

Takemura, C., Crawford, L.S., 2009. The Book of Xen. No Starch Press.

Tsakalozos, K., Roussopoulos, M., Delis, A., 2011. VM placement in non-homogeneous IaaS-clouds. In: Proceedings of the 9th International Conference on Service Oriented Computing (ICSOC'11), Paphos, Cyprus, December.

Wimmer, R., Derisavi, S., Hermanns, H., 2010. Symbolic partition refinement with automatic balancing of time and space. Performance Evaluation 67 (9), 816–836.

Xu, J., Fortes, J., 2010. Multi-objective virtual machine placement in virtualized data center environments. In: Green Computing and Communications (GreenCom), 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom), December, pp. 179–188.

Tien Van Do received the M.Sc. and Ph.D. degrees in telecommunications engineering from the Technical University of Budapest, Hungary, in 1991 and 1996, respectively. He is an associate professor in the Department of Telecommunications of the Technical University of Budapest, and a leader of Analysis, Design and Development of ICT Systems Laboratory. He habilitated at BME, and received the DSc from the Hungarian Academy of Sciences in 2011. He has participated and lead work packages in the COPERNICUS-ATMIN 1463, the FP4 ACTS AC310 ELISA, FP5 HELINET, FP6 CAPANINA projects funded by EC (where he acted as a work package leader). He led various projects on network planning and software implementations that results are directly used for industry such ATM & IP network planning software for Hungarian Telekom, GGSN tester for Nokia, performance testing program for the performance testing of the NOKIA's IMS product, automatic software testing framework for Nokia Siemens Networks. His research interests are queuing theory, telecommunication networks, cloud computing, performance evaluation and planning of ICT Systems.

Csaba Rotter research manager at Nokia Siemens Networks, takes part in different cloud related topics, like telco grade cloud computing, telco PaaS and different proof of concept network elements deployment in private clouds projects. In previous positions at Nokia Siemens Networks and Nokia his main activities were test automation concept security test automation and test automation development for large telecommunication systems. Beforehand, he was taking part in daily testing activities, planning and execution at Nokia - both client and network side. Csaba received his M.Sc. in applied electronics in Technical University of Oradea and later MSC in IT management in Central European University.