Acta Technica 61, (No. 4A/2016, 63–74

Research on the data processing and rotation mode transfer of cloud computing based on the virtual machine (VM)

Lijun Mao[1]

Abstract. In order to improve the efficiency and security in the data processing and rotation mode transfer of cloud computing, the visual machine method is proposed in this paper. The paper achieves this through workload consolidation onto a set of servers and powering off servers that become idle after consolidation. The main idea is to reduce power wastage by idle servers that do not have any workload. This paper presents an efficient method of server consolidation through virtual machine migration among server farms. The paper makes major contributions by providing a comprehensive consolidation strategy using secure VM live migration. It discusses efficient ways of deploying virtual machines to servers and migrating virtual machines among server's clusters based on sever workload utilization using dynamic round-robin algorithm. The ultimate result of the method is the reduction of the number of physical machines used since the number of physical machines used greatly affects the overall power consumption. The paper further discusses known security threats to VM'S during live migration and presents mitigation strategies to the threats.

Key words. Data processing, rotation mode, cloud computing, virtual machine.

1. Introduction

Cloud computing can be defined as "a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned" and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers. Some of the examples for emerging Cloud computing infrastructures/platforms are Microsoft Azure, Amazon EC2, Google App Engine, and Aneka 121. From the energy efficiency perspective, a cloud computing data center Can be defined as a pool of computing and communication resources organized in the way to transform the received power into computing or data transfer work to satisfy user demands [1–2]. The operation of large geographically distributed data

[1]General Education Center, Xi'an Peihua University, Shaanxi, 710125, China

centers requires considerable amount of energy that accounts for a large slice of the total operational costs for cloud data centers. One implication of Cloud platforms is the ability to dynamically adapt (scale. up or scale-down) the amount of resources provisioned to an application in order to attend variations in demand that are either predictable, and occur due to access patterns observed during the day and during the night; or unexpected, and occurring due to a subtle increase in the popularity of the application service [3].

This capability of clouds is especially useful for elastic (automatically scaling of) applications, such as web hosting, content delivery, and social networks that are susceptible to such behavior. These applications often exhibit transient behavior (usage pattern) and have different Quality of Service (QoS) requirements depending on time criticality and users' interaction patters (online/offline). Virtualization in computing is the creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, a storage device or network resources [4]. Virtualization can be viewed as part of an overall trend enterprise IT that includes autonomic computing, a scenario in which the IT environment will be able to manage itself based on perceived activity and utility computing, in which computer processing power is seen as a utility that clients can pay for only as needed. The usual goal of virtualization is to centralize administrative tasks while improving scalability and overall hardware–resource utilization.

A virtual machine (VM) is a software implementation of a machine (i.e. a computer) that executes programs like a physical machine [5]. VM's are separated into two major categories, based on their use and degree of correspondence to any real machine. A system virtual machine provides a complete system platform which supports the execution of a complete operating system (OS). Technologies, such as Dynamic Voltage and Frequency Scaling (DVFS), and Dynamic Power Management (DPM) were extensively studied and widely deployed. Because the aforementioned techniques rely on power-down and power-off methodologies, the efficiency of these techniques is at best limited. In fact, an idle server may consume about 2/3 of the peak load [6–8]. However most of the methods presented have security vulnerabilities and low energy saving. The way to address high energy problem is power required to feed a completely utilized system. This workload consolidation onto a set of servers by minimizing the peak can be achieved through idle after consolidation. The main idea is to and powering off servers that become do not have reduce power wastage by idle servers that any workload. This paper presents an efficient method of server consolidation through virtual machine migration among server farms.

2. Overview

Current Cloud computing providers have several data centers at different geographical locations over the Internet in order to optimally serve costumer needs around the world. Figure 1 depicts such a Cloud computing architecture that consists of service consumers' (Software as a Service-SaaS providers') brokering and providers' coordinator services that support utility-driven internet working of clouds:

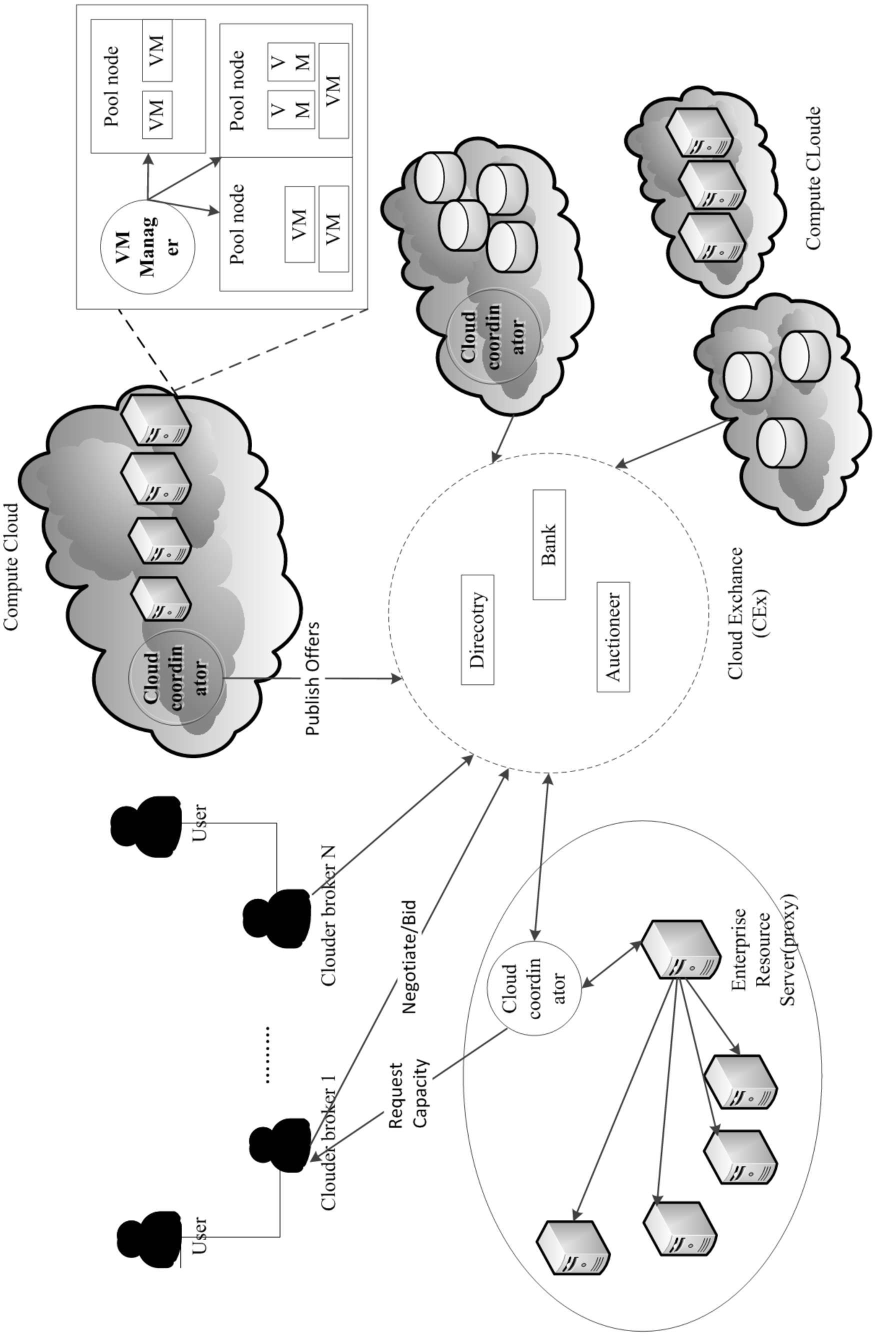

Fig. 1. Clouds and their federated network

application provisioning and workload migration. Federated inter-networking of administratively distributed clouds offers significant performance and financial benefits such as:

(i) Improving the ability of Software as a Service (SaaS) providers in meeting QoS levels for clients and offer improved service by optimizing the service placement and scale.

(ii) Enhancing the peak-load handling and dynamic system expansion capacity of every member cloud by allowing them to dynamically acquire additional resources from federation. This frees the Cloud providers from the need of setting up a new data center in every location.

(iii) Adapting to failures such as natural disasters and regular system maintenance is more graceful as providers can transparently migrate their services to others domains in the federation, thus avoiding Software License Agreement (SLA) violations and resulting penalties. Hence, federation of clouds not only ensures business continuity but also augments the reliability of the participating Cloud providers. One of the key components of the architecture presented in Fig. 1 is the Cloud Coordinator.

3. Layered model design and algorithm

Physical Cloud resources along with core middleware capabilities form the basis for delivering Infrastructure as a Service (IaaS) and Platform as a service (PaaS). The user-level middleware aims at providing SaaS capabilities. The top layer focuses on application services (SaaS) by making use of services provided by the lower layer services. PaaS/SaaS services are often developed and provided by 3rd party service providers, who are different from the IaaS providers.

3.1. Cloud applications

This layer includes applications that are directly available to define end-users as the active entity applications over the Internet. These applications may be that utilizes the SaaS supplied by the Cloud provider (SaaS providers) and accessed by end-users either via a subscription model or a pay-per-use basis. Alternatively, in this layer, users deploy their own applications. In the former case, there are applications such as Salesforce.com that supply business process models on clouds (namely, customer relationship management software) and social networks. In the latter, there are e-Science and e-Research applications, and Content-Delivery Networks. User-Level Middleware: This layer includes the software frameworks such as Web 2.0 Interfaces (Ajax, IBM Workplace) that help developers in creating rich, cost-effective user-interfaces for browser-based applications. The layer also provides those programming environments and composition tools that ease the creation, deployment, and execution of applications in clouds. Finally, in this layer several frameworks that support multi-layer applications development, such as spring and Hibernate, can be deployed to support applications running in the upper level.

3.2. Core middleware

This layer implements the platform level services that provide runtime environment for hosting and managing User-Level application services. Core services at this layer include Dynamic SLA Management, Accounting, Billing, Execution monitoring and management, and Pricing. The well-known examples of services operating at this layer are Amazon EC2, Google App Engine, and Aneka. The functionalities exposed by this layer are accessed by both SaaS (the services represented at the top-most layer in Figure 1) and IaaS (services shown at the bottom-most layer in Figure 1) services. Critical functionalities that need to be realized at this layer include messaging, service discovery, and load balancing. These functionalities are usually implemented by Cloud providers and offered to application developers at an additional premium. For instance, Amazon offers a load balancer and a monitoring service (Cloud watch) for the Amazon EC2 developers consumers. Similarly, developers building applications on Microsoft Azure clouds can use the .NET Service Bus for implementing message passing mechanism. System Level: The computing power in Cloud environments is supplied by a collection of data centers that are typically installed with hundreds to thousands of hosts. At the System Level layer, there exist massive physical resources (storage servers and application servers) that power the data centers. These servers are transparently managed by the higher-level virtualization services and toolkits that allow sharing of their capacity among virtual instances of servers. These VMs are isolated from each other, thereby making fault tolerant behavior and isolated security context possible.

The estimated finish time of a task managed by a VM is given as

$$eft(p) = est(p) + \frac{rlis}{capacity}\,, \tag{1}$$

where $est(p)$ is the estimated start time of task p and $rlis$ is the total number of instructions that the cloud network will need to execute. Thus the total capacity of a VM has multiple p tasks given as

$$capacity = \frac{\sum_{i=1}^{np} cap(i)}{np}\,, \tag{2}$$

where i is the processing strength of an individual element. Therefore, the total capacity of a VM in the cloud computing network is

$$capacity = \frac{\sum_{i=1}^{nap} cap(i)}{\max(\sum_{j=1}^{network} cores(j), np)}\,, \tag{3}$$

where $cap(i)$ is the processing strength of an individual element.

4. Experimental result and discussion

This paper considered two sets of experiments to evaluate the proposed approach. The first set of experimental tests was done to test the approach in a two-tier (2T), three-tier (3T), and three-tier high-speed (3Ths) architectures. The second experiments that were done validate the effectiveness of energy-conscious VM provisioning technique proposed. Proposed algorithms were evaluated through simulations using the C1oudSim toolkit which offers the following novel features

(i) Support for modeling and simulation of large scale Cloud computing environments, including data centers, on a single physical computing node.

(ii) A self-contained platform for modeling Clouds, service brokers, provisioning, and allocations policies.

(iii) Support for simulation of network connections among the simulated system elements.

(iv) Facility for simulation of federated Cloud environment that inter-networks resources from both private and public domains, a feature critical for research studies related to Cloud-Bursts and automatic application scaling. Some of the unique, compelling features that make C1oudSim our chosen simulation framework are:

(i) Availability of a virtualization engine that aids in creation and management of multiple, independent, and co-hosted virtualized services on a data center node.

(ii) Flexibility to switch between space-shared and time-shared allocation of processing cores to virtualized services. In contrast with other architectures 2T data center does not include aggregation switches. The core switches are connected to the access network directly using 1 GE links (referred as C2-C3) and interconnected between them using i0 GE links (referred as C1-C2). The 3Ths architecture mainly improves the 3T architecture with providing more bandwidth in the core and aggregation parts of the network. The bandwidth of the C1-CZ and C2-C3 links in 3Ths architecture is ten times of that in 3T and corresponds to 100 GE and LOGE respectively. The availability of 100 GE links allows keeping the number of core switches as well as the number of paths in the ECMP routing limited to 2 serving the same amount switches in the access. The propagation delay of all the links is set to 10 ns. The workload generation events and the size of the workloads are exponentially distributed. The average size of the workload and its computing requirement depends on the type of task. For CIW workloads the relation between computing and data transfer parts is chosen to be 1/10, meaning that with a maximum load of the data center, its severs will be occupied for 100 % while the communication network will be loaded for 10 % of its maximum capacity. For DIW workloads the relation is reverse. Under the maximum load the communication network is loaded for 100 % while computing servers far only 10 %. Balanced workloads load computing servers and data center network proportionally. The workloads arriving at the data center are scheduled for execution using secure energy-aware CIoudSim scheduler. This scheduler tends to group the workloads on a minimum possible amount of computing servers. In order to account for DIW workloads the scheduler continuously tracks buffer occupancy of network switches on the path. In case of a congestion the scheduler avoids using congested routes even if they lead to the servers able to satisfy computational requirement of the workloads. The servers left idle are put into sleep mode (DNS scheme) while on the under-loaded servers the supply voltage is reduced (DVFS scheme). The time required to change the power state in either mode is set to 100 ms.

Figure 2 presents a workload distribution among severs. The whole load of the data center (around 30 % of its total capacity) is mapped onto approximately one third of the servers maintaining load at a peak rate (left part of the chart). This way, the remaining two thirds of the servers can be shut down using DNS technique. A tiny portion of the approximately 50 out of 1536 servers which load represents a falling slope of the chart are under-utilized on average, and DVFS technique can be applied on them. Figure 3 presents data center energy consumption comparison for different types of user workloads: balanced. Balanced workloads consume the most as the consumptions of both servers and switches become proportional to the offered load of the system. CIWs stress the computing servers and leave data center network almost unutilized. On the contrary, execution of DIWs creates a heavy traffic load at the switches and links leaving the servers mostly idle. The process of scheduling for DIWs requires performing load balancing for redistributing the traffic from congested links. As a result, these workloads cannot be fully grouped at the minimum amount of the servers due to the limitations of the data center topology. This way, in real data centers with the mixed nature of workloads the scheduler may attempt a grouped allocation of CIWs and DIWs as optimal allocation policy.

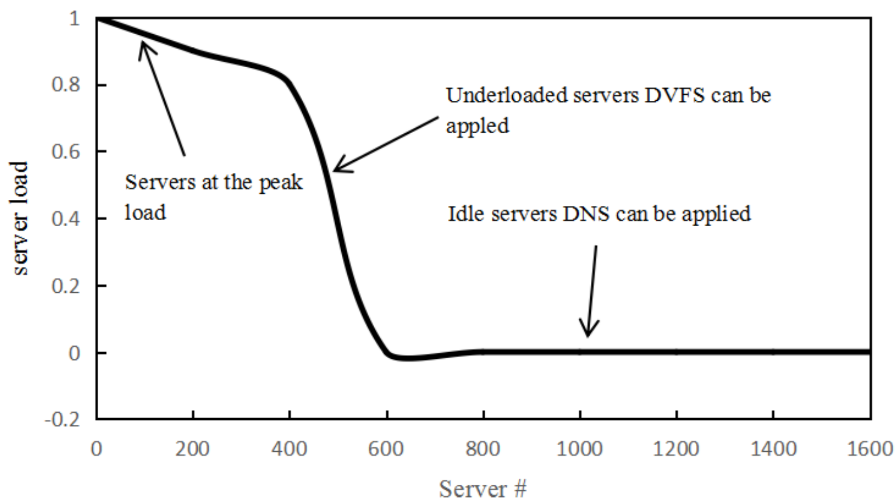

Fig. 2. Server workload distribution with a CloudSim scheduler

This component is instantiated by each cloud in the system whose responsibility is to undertake the following important activities:

(i) Exporting Cloud services, both infrastructure and platform-level, to the federation

(ii) Keeping track of load on the Cloud resources (VMs, computing services) and

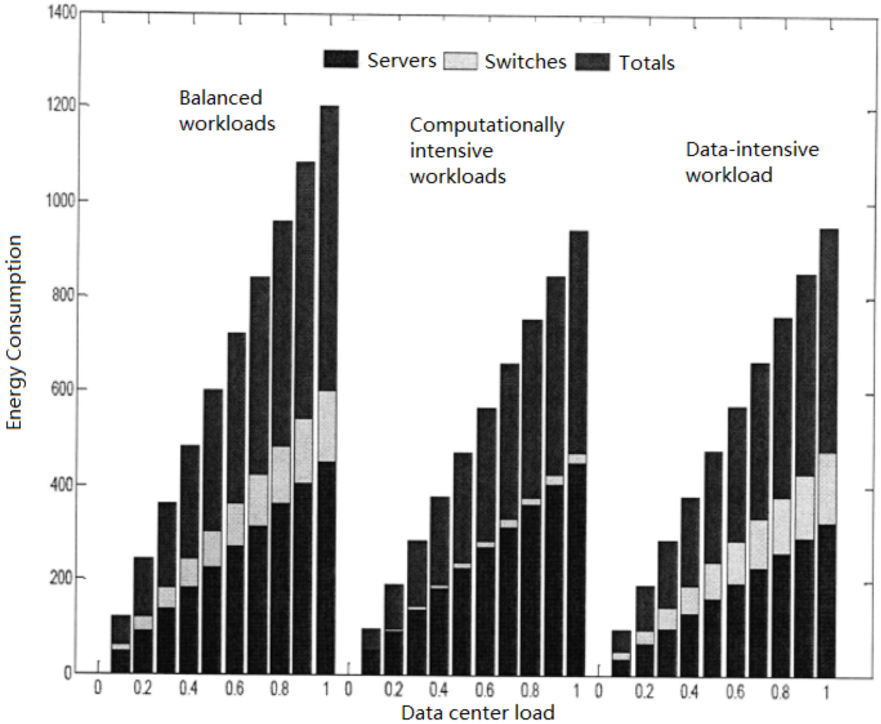

Fig. 3. Data center energy consumption for different types of workloads

undertaking negotiation with other Cloud providers in the federation for handling the sudden peak in resource demand at local cloud.

(iii) Monitoring the application execution over its lifecycle and overseeing that agreed SLAs are delivered. The Cloud brokers acting on behalf of SaaS providers identify suitable Cloud service providers through the Cloud Exchange. Further, Cloud brokers can also negotiate with respective Cloud Coordinators for allocation of resources that meets the QoS needs have hosted or to be hosted SaaS applications. The Cloud Exchange (CEx) acts as a market maker by bringing together Cloud service (IaaS) and SaaS providers. It aggregates the infrastructure demands from the Cloud brokers and evaluates them against the available supply currently published by the Cloud Coordinators. The applications that may benefit from the aforementioned federated Cloud computing infrastructure include social networks such as Facebook and MySpace, and Content Delivery Networks (CDNs). Social networking sites serve dynamic contents to millions of users, whose access and interaction patterns are difficult to predict. In general, social networking web-sites are built using mufti-tiered web applications such as Web Sphere and persistency layers like the MySQL relational database. Usually, each component will run in a different virtual machine, which can be hosted in data centers owned by different

Cloud computing providers. Additionally, each plug-in developer has the freedom to choose which Cloud computing provider offers the services that are more suitable to run his/hers plug-in. As a consequence, a typical social networking web application is formed by hundreds of different services, which may be hosted by dozens of Cloud-oriented data centers around the world. Whenever there is a variation in temporal and spatial locality of workload (usage pattern), each application component must dynamically scale to offer good quality of experience to users [9–10]. Figure 4 shows the data center energy consumption comparison and Fig. 5 shows the total energy consumption by the system in the proposed model.

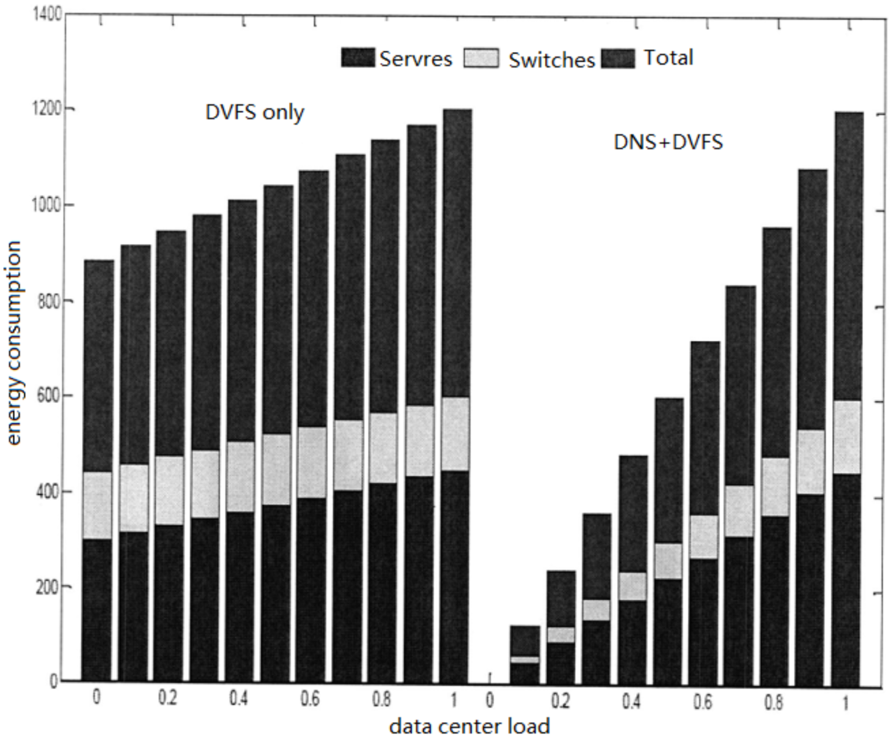

Fig. 4. Data center energy consumption comparison

Domain experts and scientists can also take advantage of such mechanisms by using the cloud to leverage resources for their high-throughput e-Science applications, such as Monte-Carlo simulation and Medical Image Registration. In this scenario, the clouds can be augmented to the existing cluster and grid based resource pool to meet research deadlines and milestones. Cloud computing servers applications as localized attacker or malware to use the same operating systems, enterprise and web virtual machines and physical servers. The ability for a remotely exploit vulnerabilities in these systems and threat to virtualized cloud computing environments. Figure 6 shows the Number of VM migrations and Fig. 7 shows the number of SLA

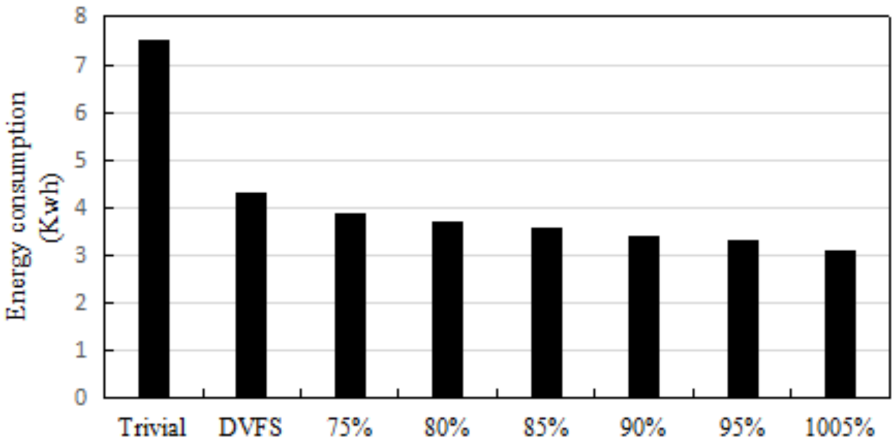

Fig. 5. Total energy consumption by the system in the proposed model

violations. In virtual machines increases the attack surface and risk of VM-to-VM compromise. Furthermore, several security vulnerabilities exist in migration, especially live migration of such systems that may degrade the protection strength or even break the protection. In VM live migration, the challenges lie in the following three aspects:

(i) Preserving the privacy and integrity of protected contents.

(ii) Packing the maintenance metadata in the Virtual Machine Monitor (VMM), conflict, and re-establishing the protection base on the solving the namespace target platform.

(iii) Eliminating the security vulnerabilities imposed by live migration. Time-of-Check to time-of-use (TOCTTOU) attack and replay attack have identified as possible attacks that a VM faces during live migration. This provides mitigation strategies against these vulnerabilities that a VM is exposed Live Migration.

5. Conclusion

This paper has done extensive investigation on various power-aware VM provisioning schemes like DVFS and DNS. The paper does two sets of experiments to test the presented approach in various architectures like two-tier (2T), three-tier (3T), and three-tier high-speed (3Ths) architectures. Previously, other researcher had performed simulation on two tier architecture only. They overlooked the fact that data centers behave differently when implemented using 2T, 3t and 3Ths.For this purpose, the paper is able to present an optimal power aware provisioning scheme based on comparative evaluation the existing schemes on various architectures. The effectiveness of these contributions has been appraised through a comprehensive simulation-driven analysis of the proposed approach based on realistic and well-known data center conditions in order to capture the transient behaviors that prevail in existing

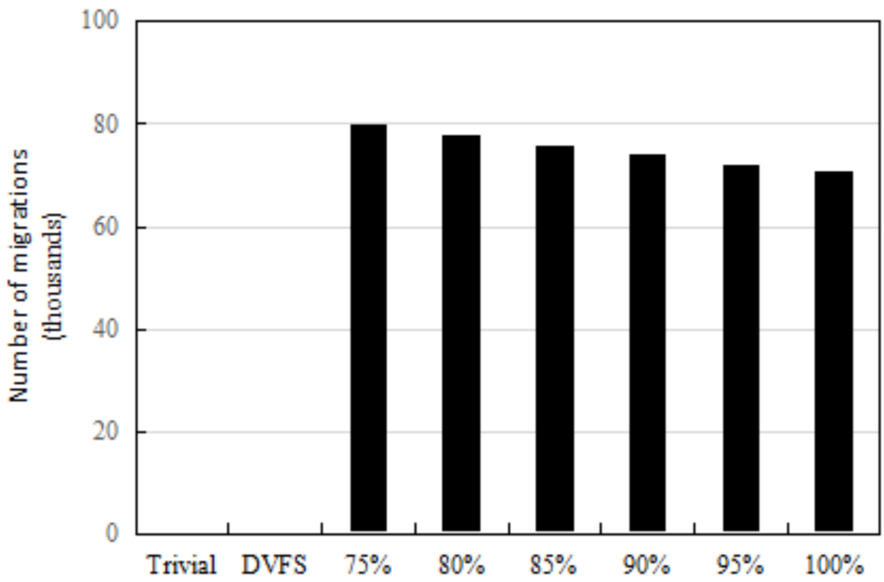

Fig. 6. Number of VM migrations

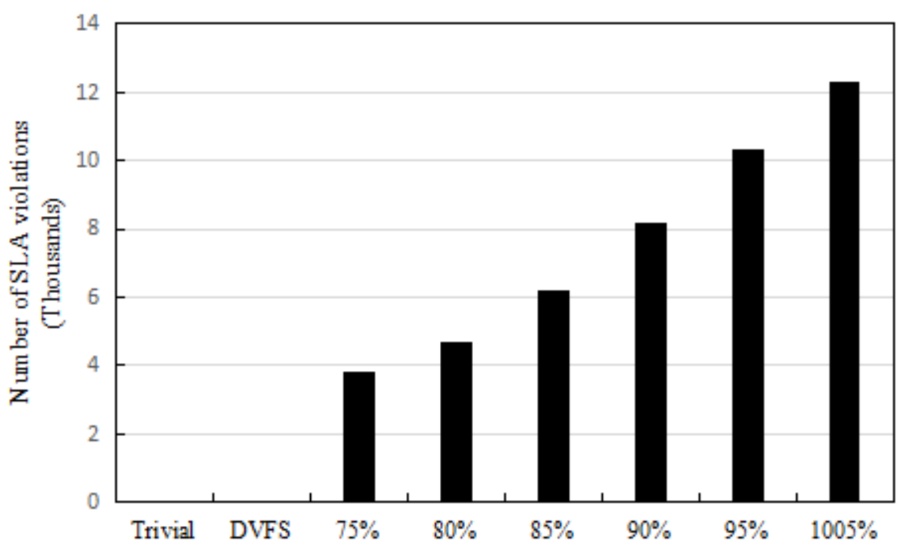

Fig. 7. Number of SLA violations

data center environments.

References

[1] M. Jalali, A.,Bouzer, B. Arasteh, M. Moloudi: *The effect of cloud computing technology in personalization and education improvements and its challenges.* Procedia - Social and Behavioral Sciences *83* (2013), 655–658.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield: *Virtual machine monitors: xen and the art of virtualization.* ACM Symposium on Operating System Principles (SOSP'03), 19–22 Oct. 2003, Bolton, Landing, New York, USA, *36*, No. 8, 164–177.

[3] T. Garfinkel, B. Pfaff, J. Chow, M. Rosenblum, D. Boneh: *Terra: a virtual machine-based platform for trusted computing.* ACM Symposium on Operating System Principles SOSP'03), 19–22 Oct. 2003, Bolton, Landing, New York, USA, *37*, No. 5, 193–206.

[4] A.Kivity, Y. Kamay, Y. Laor, U. Lublin, A. Liguori: *Kvm: the linux virtual machine monitor.* Proceedings of the Linux Symposium, 27–30 June 2007, Ottawa, Ontario, Canada, *1*, 225–230.

[5] T. Garfinkel, M. Rosenblum: *A virtual machine introspection based architecture for intrusion detection.* Internet Society's 2003 Symposium on Network and Distributed System Security (NDSS'03), 6–7 Feb. 2003, San Diego, California, USA, 191 to 206.

[6] P. Levis, D. Culler: *Matd: A tiny virtual machine for sensor networks.* ACM SIGOPS Operating Systems Review *36* (2002) No. 5, 85–95.

[7] A. Geist, A. Beguelin, J. Dongarra, J. Jiang, W. Manchek, V. Sunderam, M. Novotny, S. McKay: *PVM-parallel virtual machine: A users' guide and tutorial for networked parallel computing.* Computers in Physics *9* (1995), No. 6, 607–607.

[8] R. P. Goldberg: *Survey of virtual machine research.* J Computer *7* (1974), No. 9, 34–45.

[9] G. W. Dunlap, S. T. King, S. Cinar, M. A.Basrai, P. M. Chen: *Revirt: enabling intrusion analysis through virtual-machine logging and replay.* ACM SIGOPS Operating Systems Review *36* (2002), No. SI, 211–224.

[10] T. Lindholm, F. Yellin: *The java TM virtual machine specification.* Institute of Information & Computing Sciences *15*, (1997), No. 6, 27–59.

Received November 16, 2016