CloudSwitch: A State-aware Monitoring Strategy Towards Energy-efficient and Performance-aware Cloud Data Centers

Frank Elijorde[1] and Jaewan Lee[2]

[1]*Institute of Information and Communication Technology, West Visayas State University*
Iloilo City, Philippines

[2]*Department of Information and Communication Engineering, Kunsan National University*
Gunsan, South Korea
[e-mail: frank, jwlee @kunsan.ac.kr]
*Corresponding author: Jaewan Lee

Received December 26, 2014; revised September 6, 2015; accepted October 7, 2015; published December 31, 2015

Abstract

The reduction of power consumption in large-scale datacenters is highly-dependent on the use of virtualization to consolidate multiple workloads. However, these consolidation strategies must also take into account additional important parameters such as performance, reliability, and profitability. Resolving these conflicting goals is often the major challenge encountered in the design of optimization strategies for cloud data centers. In this paper, we put forward a data center monitoring strategy which dynamically alters its approach depending on the cloud system's current state. Results show that our proposed scheme outperformed strategies which only focus on a single metric such as SLA-Awareness and Energy Efficiency.

Keywords: Cloud Computing, Cloud Data Centers, VM Consolidation, VM Migration, Resource Provisioning, Green Computing

Following are results of study on the "Leaders in Industry-University Cooperation" Project, supported by the Ministry of Education.

http://dx.doi.org/10.3837/tiis.2015.12.002 **ISSN : 1976-7277**

1. Introduction

Evidently, businesses and organizations that invest on IT infrastructures have acknowledged the benefits of adopting a virtual infrastructure. Virtualization not only improves hardware efficiency via server consolidation, it also dramatically reduces the time required to bring a new server online; this ability to bring machines online quickly provides companies with improved business agility. Though they seem enthusiastic about the new computing paradigm, many of them haven't realized the associated challenges that need to be addressed. As with most improvements new issues may arise, and if not addressed, can reduce a virtual infrastructure's overall efficiency over time. For example, virtual machine sprawl is one such concern facing many companies using desktop or server virtualization [1]. Simply put, VMs that are created without regard for resources typically result in over-provisioning, or consuming resources well after they are no longer required. This scenario results to wastage which can easily go undetected until bottlenecks affect performance [2]. A round the world, a large number of data centers is having huge energy consumption which has an evident and serious impact on the environment. According to a study, each data center in the world consumes as much energy as 250,000 households on average [3]. In another report, it is found that the overall estimated energy bill for data centers in 2010 is $11.5 billion while energy costs in a typical data center double every five years [4]. Currently, even the rapidly-growing number of richer applications in cloud-assisted mobile ad-hoc networks will make the energy consumption problem worse [5]. Thus, lowering the energy consumption of data centers is a major issue that comes along with the rapid growth of computing applications and data.

Efficient resource provisioning is a key challenge in fulfilling the SLA (Service Level Agreement) to improve user satisfaction and to justify the investment in cloud-based deployments. For this reason, many researchers have proposed various approaches [6] to minimize the energy consumption of data centers while ensuring the desired QoS (Quality of Service). However, upholding the SLA to guarantee the QoS is another crucial, yet conflicting interest for Cloud Providers. Although over-provisioning of compute resources would guarantee good performance, this is at the expense of underutilized hardware and increased power consumption. The reduction of energy consumption in large-scale datacenters is being accomplished through an extensive use of virtualization to consolidate multiple workloads and reduce overall datacenter power consumption. Nevertheless, these consolidation strategies must also take into account additional parameters of utmost importance for datacenters such as performance, reliability, and profitability.

In this paper, we present a strategy which enables the data center to alter its behavior in dealing with the conditions pertaining to performance and energy efficiency. The issue that we intend to address is that although modern virtualization technologies can ensure isolated execution environments between VMs sharing the same physical servers, the aggressive behavior in which consolidation of compute resource and variability of the workload are carried out, would cause some VMs not getting the required amount of resource when requested. If not handled properly, this leads to SLA Violations characterized by degraded system performance as reflected by decreased throughput, increased latency, and in the worst case, failure. In this junction, we argue that it is not enough for cloud providers to just focus on the power-performance tradeoff of their cloud systems; instead, our idea is to reactively impose the most appropriate approach at a given situation. For instance, if the cloud system is stable in terms of performance where SLA violations are less likely to occur, we switch the monitoring technique to one which favors energy efficiency. In cases where cloud the system

is in an energy efficient state, then it is free to carry out a monitoring approach to improve its performance. Results show that dynamic swapping of monitoring strategies can further improve the performance-to-power ratio of a data center, setting a good balance between performance and energy efficiency.

2. Related Work

2.1 Cloud monitoring and SLA Management

Continuously monitoring the Cloud and managing the upkeep of SLA in terms of its QoS is the primary means for controlling and managing the entire infrastructure; also, it serves for providing indicators of platform and application performance. Generally, its main goal is to retrieve information that reflects the physical infrastructure of the whole Cloud platform. Due to the layered nature of hardware/software communications in a virtualized cloud data center, this procedure is extremely important to the Cloud provider and is typically hidden from the Cloud clients. Cloud monitoring and SLA management is an active field in which many commercial systems have been developed and numerous researches have been conducted. The previously-proposed approaches have the same goal which is to come up with a better way of monitoring the Cloud system in order for the Cloud providers to maintain the good performance and availability of the services they offer.

In [7], they propose the automation of SLA establishment based on a classification of cloud resources in different categories with different costs, such as on-demand instances, reserved instances and spot instances in Amazon EC2 cloud. However, this approach does not provide guarantees in terms of performance, dependability, and some other factors. A similar approach for SLA enforcement is presented in [8] which is based on classes of clients with different priorities. In the said strategy, a relative best-effort behavior is provided for clients with different priorities, but no strict performance and dependability SLOs are guaranteed. In a recent work [9], an SLA Manager is presented alongside with proposed techniques for VM selection and allocation during live migration of VMs. Using the proposed autonomous SLA violation filtering framework, they simulated a combination of IaaS and PaaS in a multi-domain setting and evaluated the performance of the aforementioned VM placement strategies. Using a SLA pricing & penalty model, they were able to manage trade-offs between the operational objectives of service providers and the customers' expected QoS requirements. An SLA-aware-Service (SLAaS) cloud model is presented in [10] as a way to integrate QoS SLA into the cloud. The Cloud SLA specific language is proposed to describe SLAs associated with cloud services in a convenient way in which a control-theoretic approach is followed to provide performance, dependability and cost guarantees for online services. The work in [11] analyzed the factors that affect live virtual machine migration time in shared clouds. As migration is essential to satisfy both service level agreements between the cloud user and the cloud provider, its strengths and weaknesses is crucial. The paper emphasized virtual machine size, network speed and dirty rate of the application which play important roles in optimizing the performance of live VM migration.

2.2 VM Consolidation in Cloud Data Centers

Even in the state-of-the-art data centers, the huge number of physical servers remains as one of the main contributors to high power consumption and carbon footprint of the data center which relates to high operation costs [12]. There are many studies on energy efficient datacenters.

The authors in [13] studied a number of schemes that transition the CPU into various low-power and sleep states to reduce the CPU idle power. In their work, they propose a dynamic idle interval prediction scheme that can estimate future CPU idle interval lengths and choose the most cost-effective sleep state to minimize power consumption at runtime. The authors in [14] aims to improve the QoS of Cloud data centers in terms of optimizing the response time in a parallel environment through a fuzzy controlled load balancer (FCLB). The proposed FCLB uses different parameters like number of processors to be used and processor speed. The work in [15] presents a power-aware application placement controller in virtualized heterogeneous systems. The placement of VMs has been optimized to minimize power consumption and migration cost at each time frame. In [16] they propose a resource provisioning approach based on dynamic thresholds to detect the workload level of the host machines. The VM selection policy uses utilization data to choose a VM for migration, while the VM allocation policy designates VMs to a host based on its service reputation. To improve the reduction of energy consumption, number of VM migrations, and number of SLA violations, the work in [17] investigated the effectiveness of VM and host resource utilization predictions in the VM consolidation task formulated as a bin-packing problem which considers both the current and future utilization of resources. In [18], the number of VM migrations is minimized by using a polynomial time algorithm. Real data center workloads were used in the experiment to validate the strategy. To optimize resource usage and reduce energy consumption of IaaS Cloud, authors in [19] proposed a robust consolidation approach for continuous monitoring and consolidation of VMs using live migration and switching idle hosts to the sleep state. Additionally, they employed an adaptive historical window selection algorithm for reducing ineffective VM migration. The work in [20] proposed an approach to achieve efficient pro-active VM scheduling which uses a multi-capacity bin packing technique that efficiently places VMs onto physical servers. They use time-series analysis techniques to extract low frequency and high frequency information about future VM workloads and their correlations.

3. Dynamically-Switching Cloud Datacenter Optimization

3.1 Energy Consumption Model

To make the power consumption model used in this study as realistic as possible, we decided to use real-world data from the components' manufacturers. So as not to complicate the model, we only consider the major components that compose a significant portion of a server's power consumption: the processor, memory, hard disk drive, and the network interface card.

As demonstrated in a benchmark [21] that subjects servers to variations of load ranging from 0-100%, it is observed that the power consumption of a server's processor increases linearly with respect to its utilization which can be represented as:

$$P_{core} = P_{max} * \frac{U_{core}}{100} \quad (1)$$

where U_{Core} is the utilization, which represents the workload of a given core, while $P_{\max}$ refers to the power consumption of the processor at maximum load as provided by the manufacturer. Therefore, the power consumption of multi-core processors is:

$$P_{cpu} = P_{idle} + \sum_{i=1}^{n} P_{core}i \quad (2)$$

where P_{idle} is a manufacturer-specific constant that denotes the power of the processor in the idle and peak state [22]. For example, on an Intel Core i7-3770 quad-core processor, these have the values 75W and 128.3W respectively. From the same source, the performance per watt rating of 4.97 is also derived.

The system memory consumes power when it is idle as well as while performing either read or write operations. Given a number of n DDR3 memory modules, their idle power consumption is expressed as:

$$P_{RAM_{idle}} = \sum_{i=1}^{n} S_i * f_i * V_i^2 \quad (3)$$

where s_i, f_i and V_i denote respectively the RAM's size, frequency and the voltage of a certain DDR3 memory module i. Given a manufacturer-specific value [23] for $P_{RAMdynamic}$ to denote a RAM's typical power consumption, the total consumption of the system memory is:

$$P_{RAM} = P_{RAMidle} + \gamma * P_{RAMdynamic}. \quad (4)$$

where:

$$y = \begin{Bmatrix} 0, if\ cpu\ is\ idle \\ \frac{RAMused}{RAMtotal}, if cpu\ is\ active \end{Bmatrix} \quad (5)$$

The power consumption of a hard disk can be categorized as *startup*, *idle*, and *accessing* modes, where each mode has a respective power consumption. Further, the idle mode power consumption can be broken down into three states: *idle*, *standby* and *sleep*. It was observed in [24] that the hard drive's power consumption in startup and accessing modes is respectively in average 3.7 and 1.4 times more than that of the idle state power consumption. From this insight, the power consumption of the hard disk is given by:

$$P_{HDD} = x* 1.4 * P_{idle} + y * PHDD_{idle} + z * 3.7 * P_{idle} \quad (6)$$

such that $x, y, z \in [0, 1]$ denote respectively whether the hard disk is in accessing, idle and startup modes, whereas *Pidle* is the idle state power consumption provided by the manufacturer's data sheet [25].

A network interface will be either in idle mode or actively transmitting or receiving packets. Simply put, the total energy consumption of an interface will be given by:

$$P_{NIC} = P_{NICidle} * T_{idle} + P_{NICdynamic} * T_{dynamic} \quad (7)$$

where $P_{NICidle}$ is the power consumption of the idle interface, $P_{NICdynamic}$ is the power when active, T_{idle} is the total idle time and $T_{dynamic}$ denotes the total active time in an observation period T.

Combining the four components, the derived power consumption of the server is given by the following equation:

$$P_{Server} = P_{CPU} + P_{RAM} + P_{HDD} + P_{NIC} \quad (8)$$

Finally, the power consumption of a datacenter is derived as:

$$P_{DCenter} = \sum_{i=1}^{n} PServer_i \quad (9)$$

3.2 System Architecture

The proposed approach consists of various components that perform equally important tasks towards a common goal of optimizing datacenter utilization in terms of performance and power consumption. At the topmost level, the *Cloud Controller* serves to oversee the global view of the cloud system. It is the server responsible for coordinating the assignment, load distribution, migration, and mapping of the VMs running in the datacenter. Within the *Cloud Controller*, a number of sub-components can be found: a) the *VM Manager*, responsible for the consolidation of the required VM image from the *VM Repository*, b) the *Load Distribution Monitor* which is tasked in keeping track of the VM's respective resource consumption as well as keeping an updated mapping of the VMs currently hosted by the VM Hosts, and c) the *Migration Handler*, which is the entity responsible for executing the process of VM migration which utilizes the algorithms for Host and VM selection. Each host is assigned a *Local Resource Manager* which is responsible for the supervision of resources being made available to the VMs that are hosted. Furthermore, each host is also provided with its own *Load Monitor* to keep track of its resource consumption in order to detect and report potential occurrences of underloading and overloading to the *Cloud Controller*. The Host is also equipped with a *Host Coordinator* which is useful for keeping an updated list of the VMs hosted by the server as well as the resource consumption of each VM, which is also forwarded to the *Central Database*. The architecture of the proposed approach is show in **Fig. 1**.

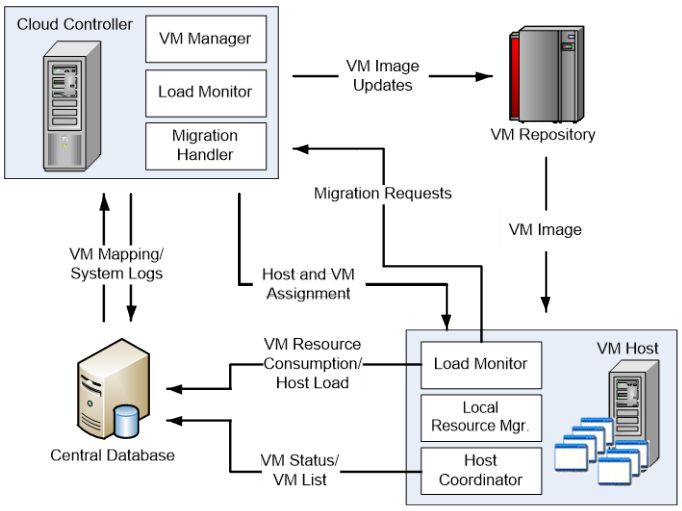

Fig. 1. The System Architecture.

As already understood, overloading and underloading are both undesirable scenarios in a datacenter for the reason that they would cause performance degradation and inefficient power usage. The process of resolving the occurrence of such events is shown in **Fig. 2**.

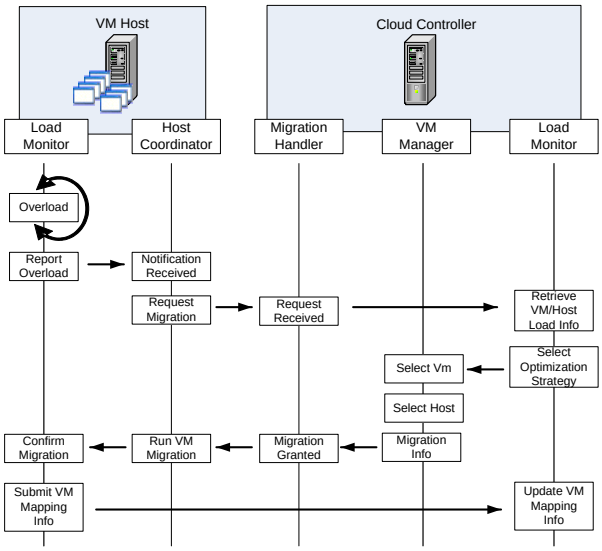

Fig. 2. Overload and Underload resolution

Whenever the *Load Monitor* detects an overloading it is reported to the *Host Coordinator* via a notification message, which in return will initiate a migration request upon receipt of the notification. The migration request will be received by the *Cloud Controller* via its *Migration Handler*, which will then inform the *Load Monitor* to retrieve the VM and Host info from the central database. After the required VM/Host information is retrieved, it will carry out an important procedure which is the selection of the optimization strategy to be utilized for determining the VM to be migrated and the host to which it will be passed. After the VM and host selection, the migration details will be passed by the *VM Manager* to the *Migration Handler*, which will grant the final authority for the migration process of the requesting host. Back to the host's side, the *Host Coordinator* will initiate the migration procedure. If successful, the migration will be confirmed by the host's *Load Monitor* and an updated VM mapping information will be sent back to its counterpart at the *Cloud Controller*. The same procedure applies for instances of underloading.

3.3 Datacenter Monitoring Strategy

In a datacenter, the VMs experience highly dynamic workloads as reflected by the CPU usage which varies over time. Normally, the power consumption is mostly derived from the actual usage of the CPU, memory, disk storage and network interfaces in the data center. Based on the findings of Intel Labs [26], a significant portion of a server's power consumption is attributed to the CPU, followed by the memory, and losses due to the power supply inefficiency. Recent studies [27, 28] show the CPU utilization has an impact on power consumption; that is, the impact is linear when dynamic voltage and frequency scaling is applied. Therefore, the resource capacities of the host and resource usage by VMs can be characterized by a single parameter, the CPU performance.

In this section, the general view of the proposed datacenter monitoring strategy is discussed. As mentioned, our goal is to come up with an approach which allows the cloud controller to switch its optimization strategy based on the current state of the datacenter. In **Algorithm 1,** it is shown that in each monitoring interval of the data center the current cpu utilization CPU_u is derived by adding up the cpu utilization of the active hosts. Moreover, the current capacity CPU_{dc} of the datacenter is calculated from the total cpu capacity of the active hosts. The actual utilization level $Util_{dc}$ of the data center for the given interval is then derived by $Util_{dc} = CPU_u / CPU_{dc}$.

```
Algorithm: Performance Monitoring Strategy
1.  For each DC interval {
2.  CPU_u ← 0 //initialize dc utilization
3.    For each Host h {
4.      CPU_u =+ CPU_h //update cpu utilization
5.      CPU_dc =+ maxCPU_h //update dc capacity
6.    }
7.    Util_dc ← CPU_u/CPU_dc
8.          //get dc utilization percentage
9.  If (strategy = Power) {
10.   If (Util_dc = max) {
11.    If GetSLAV >= threshold
12.      SwitchStrategy(SLA)
13.      // switch strategy to sla-aware
14.   }
15. }
16. If (strategy = SLA) {
17.   If (Util_dc = min) {
18.    if Pwr_eff <= threshold
19.      SwitchStrategy(Power)
20.      // switch strategy to power-aware
21.   }
22. }
23. }
```

Algorithm 1. The monitoring strategy.

Finally, the switching strategy is performed. If the current strategy is focused on power consumption, the algorithm will check if the data center utilization is at maximum level. If so, the overall SLA violation is compared to a threshold. Once the threshold is met or surpassed the monitoring strategy is switched to one that emphasizes performance. Conversely, if the current strategy is aimed at keeping the SLA low, the data center utilization is watched until it reaches the minimum level. In such case, the overall power efficiency is also compared to a given threshold. If the power efficiency drops below the threshold, the monitoring strategy is switched back to a Power-aware state. Using this strategy, it is assured that whenever the need arises, the cloud controller is able to enforce an optimization scheme which would benefit a specific goal whether it is towards performance or energy-efficiency.

3.4 Virtual Machine Selection Strategy

During the migration process, selecting the right VMs for migration is of utmost importance. It is understood that migrations should be carried out with caution due to the overhead involved when performing the procedure. Especially in the case of overloading, simply choosing the

most heavily utilized VM during the monitoring period could sometimes cause more trouble than benefits. Due to the amount of resources bound to it, the overhead involved during migration would cause performance degradation not only to the VM itself, but possibly to the whole cloud system as well. In this work, two VM selection algorithms are utilized, one directed towards power consumption and the other towards performance.

In **Algorithm 2**, the power-aware strategy called Minimum Host Aggregate is shown. Using the power consumption model presented in the previous section, the average aggregate of the compute resources utilized by the VMs for a certain monitoring period is determined. In each iteration, hosts requesting for migration had their VMs checked until a candidate for migration is selected and added to the final migration list. Since migrating a considerably loaded VM is sure to cause disruption, the strategy intends to minimize the overhead by selecting the VM with the smallest footprint. Aside from the migration overhead, this strategy serves another benefit by making sure that every server is utilized to its optimum level by leaving as little unallocated resource as possible after migrating a VM from a host, thereby maintaining a desirable performance-to-power ratio.

```
Algorithm: Minimum Host Aggregate
1.  Input: HMigList, //host migration list
2.  Output: VMList //VM migration list
3.  Sort(HMigList, utilization)
4.     //sort hosts, decreasing utilization
5.  For each host in HMigList {
6.     curAggregate ← Max
7.     For each vm in host{
8.        vmCPU ← vm.CPU
9.        vmRAM ← vm.RAM
10.       vmHDD ← vm.HDD
11.       vmNIC ← vm.NIC
12.       vmAgg ← vmCPU+vmRAM+vmHDD+vmNIC
13.       GetMeanAggregate(vmAgg)
14.       if vmAgg < curAgg {
15.          curAgg ← vmAgg
16.          minVM ← vm
17.       }
18.       VMList.Add(minVM)
19.    }
20. }
21. Return VMList
```

Algorithm 2. Power-aware VM selection strategy

A strategy intended for maintaining QoS is shown in **Algorithm 3**. The strategy, called Minimum Migration Time works by selecting the VM with the least average CPU utilization for the given period in order to make the overhead as little as possible. By choosing the VM with smallest resource consumption, mutual benefits can be achieved by the cloud provider and its client. With a tightly controlled VM migration environment, performance degradation of the cloud system is minimized while at the same time service disruption on the part of the client is barely noticeable.

```
Algorithm: Minimum Migration Time
1.  Input: HMigList, //host migration list
2.  Output: VMList //VM migration list
3.  Sort(HMigList, utilization)
4.  //sort hosts, decreasing utilization
5.  For each host in HMigList {
6.     curCPU ← Max
7.     For each vm in host{
8.        vmCPU ← GetMeanCPU(vm)
9.        if vmCPU >= curCPU {
10.          curCPU ← vmCPU
11.          maxVM ← vm
12.       }
13.       VMList.Add(maxVM)
14.    }
15. }
16. Return VMList
```

Algorithm 3. Performance-aware VM selection strategy.

3.5 SLA-Aware Host Selection Strategy

Another equally important consideration during the VM migration process is the host selection strategy. The strategy for choosing the hosts for the migrating VMs is concerned not only about finding hosts that can support them but also to maintain the desired system throughput by keeping the disruption as little as possible [29]. Selecting the server that will host the virtual machine to be launched or migrated is an equally important consideration. Thus, a VM assignment strategy should not only look for host servers that can support them but also those that can maintain a desirable SLA adherence. For this purpose, we utilize the host selection approach from our previous work [30].

```
Algorithm: SLA-Aware Host Selection
Input: VMRequest, ActiveHosts
Output: VMAssigment //VM Assigment to hosts
1.  Sort(ActiveHosts, utilization)
2.     //sort hosts, decreasing utilization
3.  Sort(VMRequest, volume)
4.     //sort VMs, decreasing volume
5.  For each vm in VMRequest  {
6.     BestSLAV ← Max
7.     AssignedHost ← null
8.     For each host in ActiveHosts {
9.        if host.canSupport(vm) {
10.          curSLAV ← GetSLAHistory(host)
11.          if curSLAV < BestSLAV {
12.             BestSLAV ← curSLAV
13.           VMAssigment.update(vm, host)
14.             Break //go to next vm
15.          }
16.       }
17.    }
18. }
19. Return VMAssigment
```

Algorithm 4. The VM assignment strategy.

The resource consumption of a VM is defined as the volume v of the resources actually consumed derived by summing up the fractions of resources (e.g. CPU, RAM, Network Bandwidth) actually consumed by the VMs, which is multiplied with corresponding weights. The weight values are assigned depending on the type of the VM machine to be provisioned. For example, a VM for serving compute intensive applications would give more weight to CPU while a transactional database server would require more weight for network bandwidth. From this, the volume set is defined as $\{v_1, v_2,\ldots, v_N\}$ composed of the VM's resource consumption accumulated on a given period. In Algorithm 4, high-volume VMs are assigned to hosts based on the level of SLA violations they have encountered. To realize the goal of minimizing the overhead resulting from migration, we utilize an approach which considers the VM's resource consumption behavior. Therefore, hosts which encountered lower SLA violations are favorably chosen to handle a VM with higher demands while light VMs are assigned to servers with fair SLA violation.

4. Implementation and Evaluation Results

4.1 Implementation

In order to evaluate the proposed approach in a realistic setting, we implemented a prototype in a test bed environment. The system setup is composed of 4 VM Hosts, 1 VM Server, and 1 Cloud Controller which have the following specifications:

Table 1. Hardware specifications.

CPU	Intel Core i7-3770 Quad-Core Processor 3.4 GHz
RAM	Kingston DDR3-1333 RAM VM Host/Cloud Controller – 4GB VM Server – 12 GB
HDD	Western Digital WD2500AAKX 250GB 7200RPM
NIC	100/1000 Mbps Ethernet Controller

Each VM Host is assigned with 8 virtual machines running Windows7 installed in their virtual hard drive of 20GB. Furthermore, the VMs were also provided a bridged virtual NIC, 256 MB RAM, and 1 CPU core. All the Server side and Client side cloud components were implemented using the C# programming language. For the central database, MySQL Server is used, while for the VM repository an ISCSI NAS was used. The testbed datacenter is put into test for 24 hours using a workload generator. In order to stress the cloud datacenter and to encourage aggressive VM consolidation, a workload ranging from 75-95% is introduced. During the system startup, each VM Host is assigned with a pre-defined set of virtual machines just enough to induce a busy but stable state. The VM Hosts of the cloud system were simultaneously started and as soon as the first batch of Host and VM workloads have been transmitted to the central database, the components of both the VM Host and Cloud Controller came into action. Prior to the actual performance evaluation, a couple of interesting behaviors were observed.

The first observation that we would like to point out is the time that each VM Selection approach encountered its first SLA Violation since system startup. The first to suffer SLA Violation is the Minimum Host Aggregate (MinAgg) which is recorded at 31 minutes, followed by Minimum Migration Time (MinCPU) at 1.5 hours, then by Switching Strategy (Switch) at 3.33 hours. The observations are presented in the following figures:

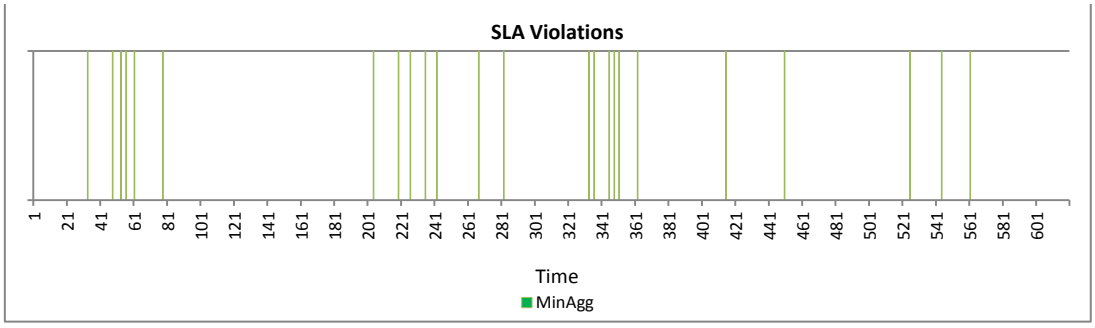

Fig. 3(a). SLA Violations of MinAgg approach.

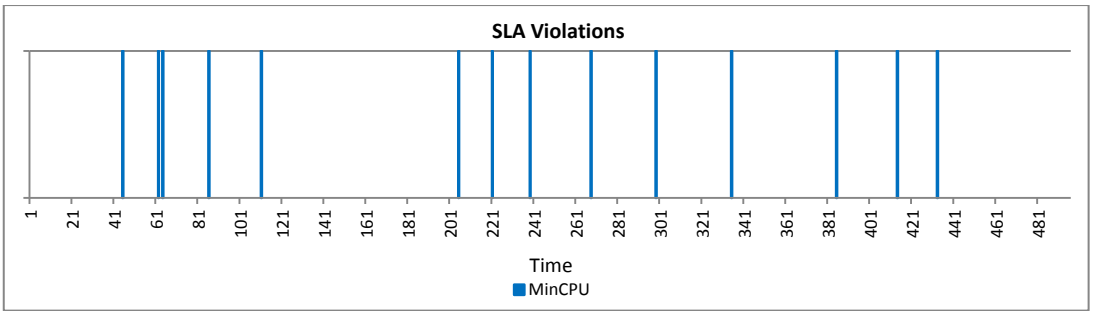

Fig. 3(b). SLA Violations of MinCPU approach.

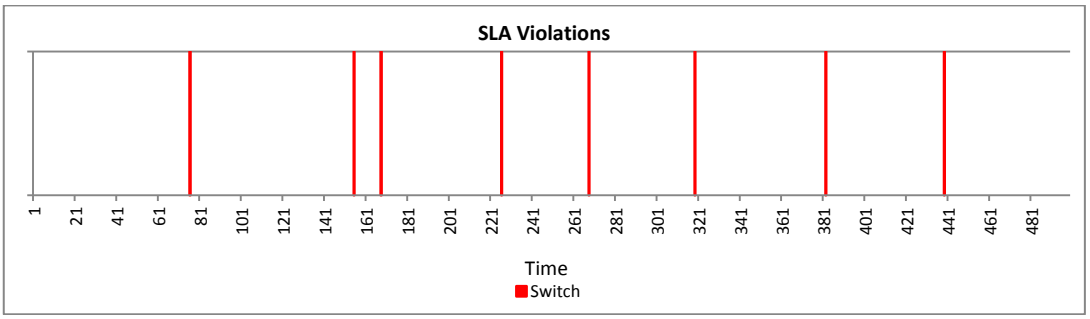

Fig. 3(c). SLA Violations of Switch approach.

Putting them altogether in **Fig. 4**, the SLAV rate of each VM selection strategy can be easily compared with each other. As the figure points out, the MinAgg approach has the most frequent occurrences of SLA violation followed by MinCPU, while that of the Switch is the lowest. It is also worth pointing out that the frequency of SLAV for Switch has better spacing compared to other approaches, which means that SLA violations rarely occur. This is due to its dynamic switching of monitoring strategies which results to a better host utilization. Such characteristic would greatly benefit the cloud system by minimizing service disruption caused by migrating VMs.

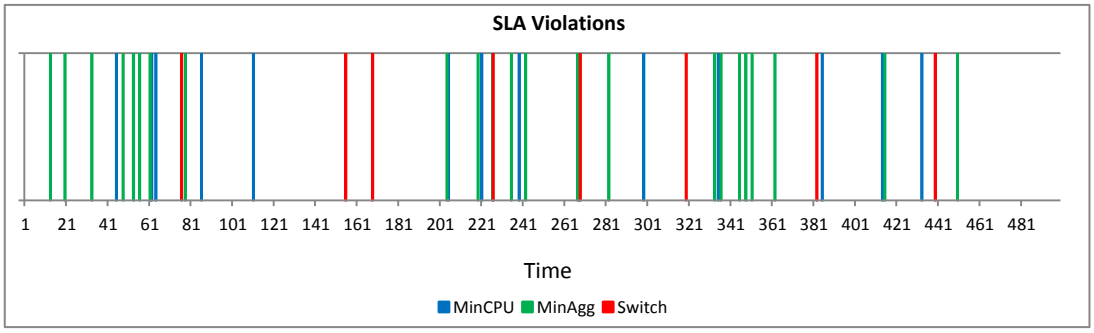

Fig. 4. SLA violation rates compared.

4.2 Evaluation Results

In this section, we will compare the performance of the aforementioned techniques and the evaluation results of the study will be discussed. Before we proceed, recall that the main goal in designing optimization strategies for cloud data centers is to attain a good balance between performance and power consumption. Standing on that argument, we put forward two approaches (MinCPU, MinAgg) each respectively designed to handle performance-awareness and energy efficiency. We would like to emphasize that the said techniques, although attempted to enforce high QoS and low power consumption, were not able to keep a good balance between the two metrics. That is, the attainment of one goal would mean a trade-off to the other.

In **Fig. 5,** the respective power consumptions of the two strategies are shown. It can be seen that the MinCPU has the higher power consumption as characterized by its graph. On numerous instances, it reached a peak consumption of more than 300W. As with MinAgg, its power consumption is only within levels below those of the MinCPU strategy.

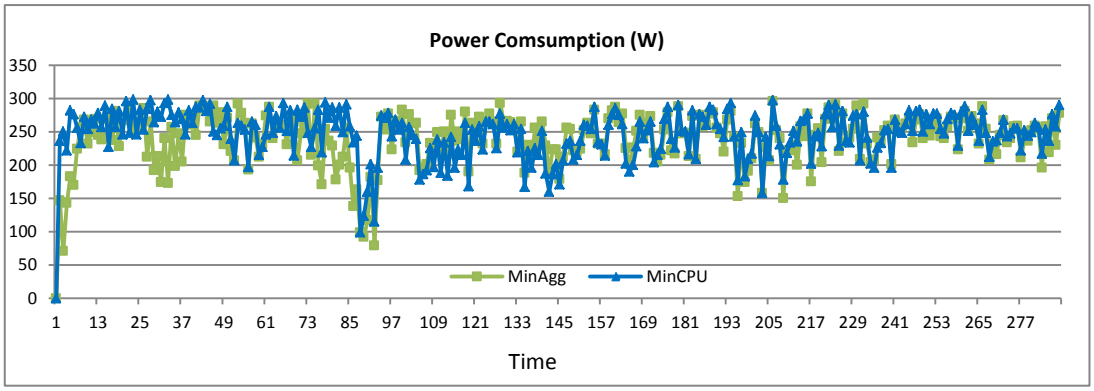

Fig. 5. Power consumption comparison.

In **Fig. 6**, the CPU utilization of the two strategies is presented. By looking at the figure, it can be readily noted that the CPU utilization for MinAgg is higher compared to MinCPU. This is due to the policy imposed by MinAgg which favors higher utilization level among hosts, thereby resulting to fewer but considerably loaded active servers as opposed to MinCPU.

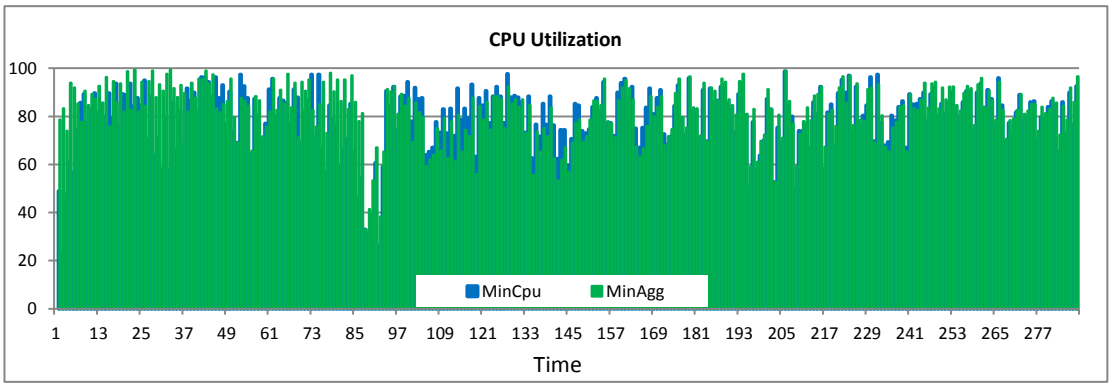

Fig. 6. CPU utilization comparison.

The findings that we have can now be used as an opportunity to derive the strengths of the two strategies and combine them to come up with a better approach of monitoring and optimizing the cloud data center. This is exactly the motivation for the Dynamic Switching Strategy presented in this paper.

In **Table 2**, the respective average cpu, average host power, and total power consumption of the three approaches are presented. With regards to the average cpu utilization, the lowest value is that of the Switch strategy. As for the average host power, our proposed approach is also able to consume the lowest average power per host. Quite expectedly, it also ended up with the lowest total power consumption for the entire operating period of the data center. All these were due to the ability of our monitoring scheme to enforce an appropriate action for a given scenario. This allows the system to prioritize performance if the current power consumption of the datacenter is still found to be efficient. Otherwise, the reduction of power consumption is given more consideration as long as the occurrence of SLA violations is still tolerable. Looking at these results it can be surmised that the outcomes are indeed affirmative of the initial findings regarding Power consumption and CPU utilization trade-off between the MinCPU and MinAgg strategy.

Table 2. Summary of CPU and Power Consumption comparison.

	AveCPU	AveHostPwr	TotalPwr
MinCPU	79.70	87.62	55341.90
MinAgg	82.21	89.87	53756.70
Switch	78.69	84.81	53383.70

Finally in **Table 3**, we show the number of migrations granted, migrations not granted, migration success rate, and SLA violation rate achieved by the respective strategies. Looking at the number of successful migrations, Switch was able to achieve the highest; as for the number of migrations not granted, MinCPU has the lowest. With regards to the migration success rate, Switch was able to complete the most number of migrations. Lastly, the SLA violation rates of the three approaches are shown. The results exhibited by the three approaches are obviously consistent with their respective migration success rates.

Table 3. Summary of Migration and SLAV comparison.

	Migrations Granted	Migrations Not Granted	Migration Success %	SLA Violation %
MinCPU	215	46	97.86	2.605
MinAgg	272	64	97.65	3.015
Switch	322	68	97.89	1.025

5. Conclusion

In a cloud data center, performance and power consumption are two opposing ends. Guaranteeing good performance is achievable by leveraging the amount of available hardware although at the expense of increased power consumption. In this paper, we presented a strategy which enables the data center to switch its monitoring strategy depending on its performance and energy efficiency. Initially, two different approaches were introduced and evaluated, and each of them was found to perform better towards a single goal. We took notice of the strengths of both approaches and combined them to come up with a better monitoring technique for cloud data centers. Results show that our dynamic switching strategy was able to outperform those which only focus on a single metric pertaining to Performance and Power. Our work was able to outperform its counterparts in terms of CPU utilization, power consumption, migration success rate, and SLA violation rate. These results conform to our idea that swapping monitoring strategies according to a desired state can further improve the performance-to-power ratio of a data center.

References

[1] VMWare, “Controlling Virtual Machine Sprawl.” Article (CrossRef Link)
[2] Dell, “Controlling Virtual Machine Sprawl: How to better utilize your virtual infrastructure.” Article (CrossRef Link)
[3] J. Kaplan, W. Forrest, N. Kindler, “Revolutionizing Data Center Energy Efficiency,” McKinsey, 2009.
[4] Supermicro, “Supermicro Titanium Power Supply Solution White Paper.” Article (CrossRef Link)
[5] N. D. Han, Y. Chung, M. Jo, "Green data centers for cloud-assisted mobile ad hoc networks in 5G," *Network, IEEE* , vol.29, no.2, pp.70-76, 2015. Article (CrossRef Link)
[6] R. N. Calheiros, R. Ranjan and R. Buyya, “Virtual Machine Provisioning Based on Analytical Performance and QoS in Cloud Computing Environments,” in *Proc. of International Conference in Parallel Processing*, February, 2011. Article (CrossRef Link)
[7] M. B. Chhetri, Q. B. Vo, and R. Kowalczyk, “Policy-Based Automation of SLA Establishment for Cloud Computing Services,” in *Proc. of The 12th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing, 2012.* Article (CrossRef Link)
[8] M. Macias and J. Guitart, “Client Classification Policies for SLA Enforcement in Shared Cloud Datacenters”, in *Proc. of The 12th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, 2012. Article (CrossRef Link)
[9] K. Lu, R. Yahyapour, P. Wieder, C. Kotsokalis, E. Yaqub, and A. I. Jehangiri, “Qos-Based Resource Allocation Framework for Multidomain Sla Management in Clouds,” *International Journal of Cloud Computing*, vol. 1, no. 1, 2013.
[10] D. Serrano, S. Bouchenak, Y. Kouki, T. Ledoux, J. Lejeune, J. Sopena, L. Arantes, P. Sens, "Towards QoS-Oriented SLA Guarantees for Online Cloud Services," in *Proc. ofCluster, Cloud*

and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, pp.50-57, 2013. Article (CrossRef Link)

[11] N. Kumar, S. Saxena, "Migration Performance of Cloud Applications- A Quantitative Analysis," *International Conference on Advanced Computing Technologies and Applications (ICACTA)*, vol. 45, pp 823–831, 2015. Article (CrossRef Link)

[12] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," *Univ. of Melbourne, Technical Report, CLOUDS-TR-2010-3*, 2010.

[13] D. Lide, Z. Dongyuan, H. Justin, "*Optimizing Cloud Data Center Energy Efficiency via Dynamic Prediction of CPU Idle Intervals*," in *Proc. of Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pp.985-988, 2015. Article (CrossRef Link)

[14] S.R. Jena, B. Dewan, "Improving quality of service constraints of Cloud data centers," in *Proc. of Computing for Sustainable Global Development (INDIACom), 2015 2nd International Conference on*, pp.153-158, 2015.

[15] A. Verma, P. Ahuja, and A. Neogi, "pMapper: power and migration cost aware application placement in virtualized systems," in *Proc. of the 9th ACM/IFIP/USENIX International Conference on Middleware*, pp 243–264, 2008. Article (CrossRef Link)

[16] F. Elijorde and J. Lee, "Performance Aware and Energy Oriented Resource Provisioning in Cloud Systems Based on Dynamic Thresholds and Host Reputation," *Journal of Korean Society for Internet Information*, vol. 14, no. 5, pp. 39-48, 2013. Article (CrossRef Link)

[17] F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, H. Tenhunen, "Utilization Prediction Aware VM Consolidation Approach for Green Cloud Computing," in *Proc. of Cloud Computing (CLOUD), 2015 IEEE 8th International Conference -on* , pp.381-388, 2015. Article (CrossRef Link)

[18] K. Ye , Z. Wu, C. Wang, B. B. Zhou, W. Si, X. Jiang, A.Y. Zomaya, "Profiling-Based Workload Consolidation and Migration in Virtualized Data Centers," *Parallel and Distributed Systems, IEEE Transactions on*, pp.878-890, 2015. Article (CrossRef Link)

[19] I. Takouna, E. Alzaghoul, C. Meinel, "*Robust Virtual Machine Consolidation for Efficient Energy and Performance in Virtualized Data Centers*," in *Proc. of Internet of Things (iThings), 2014 IEEE International Conference on, and Green Computing and Communications (GreenCom)*, pp.470-477, 2014. Article (CrossRef Link)

[20] H. Lin, Q. Xin, Y. Shuo, S. Midkiff, "Workload-Driven VM Consolidation in Cloud Data Centers," in *Proc. of Parallel and Distributed Processing Symposium (IPDPS), 2015 IEEE International Conference on*, pp.207-216, 2015. Article (CrossRef Link)

[21] Standard Performance Evaluation Corporation, "SPEC 2008 Benchmark Results." Article (CrossRef Link)
CPU Boss, "Intel Core i7 3770K vs AMD FX 8350."

[22] Tom's Hardware, "How Much Power Does Low-Voltage DDR3 Memory Really Save?" Article (CrossRef Link)

[23] EU Energy Star, "Technical Library – Hard Disk Drive." Article (CrossRef Link)

[24] Western Digital, "WD Caviar® Blue™ Spec Sheet." Article (CrossRef Link)

[25] L. Minas and B. Ellison, Energy Efficiency for Information Technology, "How to Reduce Power Consumption in Servers and Data Centers," *Intel Press*, 2009.

[26] A. Beloglazov and R. Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers," *Concurrency and Computation: Practice and Experience*, vol.24, pp.1397-1420, 2012. Article (CrossRef Link)

[27] D. Kusic, JO. Kephart, JE. Hanson, N. Kandasamy, G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," in *Proc. of the International Conference on Autonomic Computing*, pp.3-12, 2008. Article (CrossRef Link)
[28] F. Elijorde and J. Lee, "Workload Re-classification and Reputation-based Host Selection Towards Reliable and Energy Efficient Cloud Data Centers," in *Proc. of The 9th Asia Pacific International Conference on Information Science and Technology*, 2014
[29] F. Elijorde and J. Lee, "Attaining Reliability and Energy Efficiency in Cloud Data Centers Through Workload Profiling and SLA-Aware VM Assignment," *International Journal of Soft Computing and Its Applications*, Vol. 7, No. 1, 2015.

Frank Elijorde received his B.S. degree in Information Technology and M.S. degree in Computer Science from Western Visayas College of Science and Technology, Philippines, in 2003 and 2007 respectively. He received his Ph.D. in Information and Telecommunications Engineering from Kunsan National University, Korea, in 2015. Currently, he is a Professor at the Institute of ICT in West Visayas State University, Iloilo City, Philippines. His research interests include distributed systems, cloud systems, data mining, ubiquitous computing, and context-aware systems.

Jaewan Lee received his B.S., M.S., and Ph.D. degrees in Computer Engineering from Chung-Ang University in 1984, 1987, and 1992, respectively. Currently, he is a Professor at the Department of Information and Communication Engineering in Kunsan National University, Gunsan City, South Korea. His research interests include distributed systems, database systems, data mining and cloud systems.