Goal-driven Optimization Strategy for Energy and Performance-Aware Data Centers for Cloud-Based Wind Farm CMS

Frank Elijorde[1], Sungho Kim[2], and Jaewan Lee[3]
[1]Institute of Information and Communication Technology, West Visayas State University
Iloilo City, Philippines
[2]Department of Control and Robotics Engineering, Kunsan National University
[3]Department of Information and Communication Engineering, Kunsan National University
Gunsan, South Korea
[e-mail: frank, shkim, jwlee @kunsan.ac.kr]
*Corresponding author: Jaewan Lee

Received October 20, 2015; revised January 12, 2016; accepted February 3, 2016; published March 31, 2016

Abstract

A cloud computing system can be characterized by the provision of resources in the form of services to third parties on a leased, usage-based basis, as well as the private infrastructures maintained and utilized by individual organizations. To attain the desired reliability and energy efficiency in a cloud data center, trade-offs need to be carried out between system performance and power consumption. Resolving these conflicting goals is often the major challenge encountered in the design of optimization strategies for cloud data centers. The work presented in this paper is directed towards the development of an Energy-efficient and Performance-aware Cloud System equipped with strategies for dynamic switching of optimization approach. Moreover, a platform is also provided for the deployment of a Wind Farm CMS (Condition Monitoring System) which allows ubiquitous access. Due to the geographically-dispersed nature of wind farms, the CMS can take advantage of the cloud's highly scalable architecture in order to keep a reliable and efficient operation capable of handling multiple simultaneous users and huge amount of monitoring data. Using the proposed cloud architecture, a Wind Farm CMS is deployed in a virtual platform to monitor and evaluate the aging conditions of the turbine's major components in concurrent, yet isolated working environments.

Keywords: Cloud Data Centers, Cloud Computing, Green Computing, Wind Turbines, Condition Monitoring System

A preliminary version of this paper was presented at APIC-IST 2015 and was selected as an outstanding paper. This work (Grants No.C0268141) was supported by Business for Academic-industrial Cooperative establishments funded Korea Small and Medium Business Administration in 2015.

http://dx.doi.org/10.3837/tiis.2016.03.024 **ISSN : 1976-7277**

1. Introduction

Cloud computing has received a growing amount of attention in recent years due to its promising service delivery model that requires a limited amount of resources on the customer's side. Its business model is based on supplying infrastructure consisting of large pools of high-performance computing resources and high-capacity storage devices that are shared among the services offered by the provider [1]. 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. However, upholding the SLA to guarantee the QoS is another crucial, yet conflicting interest for Cloud Providers. To address this issue, we present a strategy which enables the data center to alter its behavior in dealing with the conditions pertaining to performance and energy efficiency. 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.

Another field of interest explored in this paper is in the area of wind energy. Among wind farms, Condition Monitoring Systems (CMS) aim to provide operators with information regarding the health of their turbines, which in turn can help them improve operational efficiency by allowing more informed decisions regarding maintenance. Generally, the basic objectives of the maintenance activity are to deploy the minimum resources required to ensure that components perform their intended functions properly, to ensure system reliability and to recover from breakdowns [2]. However, SCADA data could quickly accumulate and create large and unmanageable volumes that can hinder attempts to deduce the health of a turbine’s components. As such, it would prove beneficial from the perspective of wind farm operators if the data could be analyzed and interpreted automatically to support the operators in identifying defects.

Traditionally, wind farm operators have their own dedicated infrastructure with their own servers, software applications and development platforms for their monitoring and control systems. Eventually, the measures involved are associated with purchasing hardware and software, in which update and maintenance could be costly and time intensive. As shown in **Fig. 1**, as a wind farm expands, a traditional on-site infrastructure could be more costly due to the large number of computer hardware that needs to be maintained and managed. Considering the tendency towards the use of numerous wind turbines and the fact that wind farms are geographically dispersed and often located in remote areas, cost considerations make it necessary to reevaluate the traditional monitoring setup. Putting it in the context of condition monitoring systems, with Cloud Computing wind farm operators can have convenient, on-demand access to a shared pool of configurable computing resources (networks, servers, storage, applications etc.) that can be quickly provisioned and released with minimal management effort or service provider interaction. Specifically, cloud computing services can address needs for large scale real-time computing, communication, transfer and storage of data generated by geographically dispersed SCADA systems.

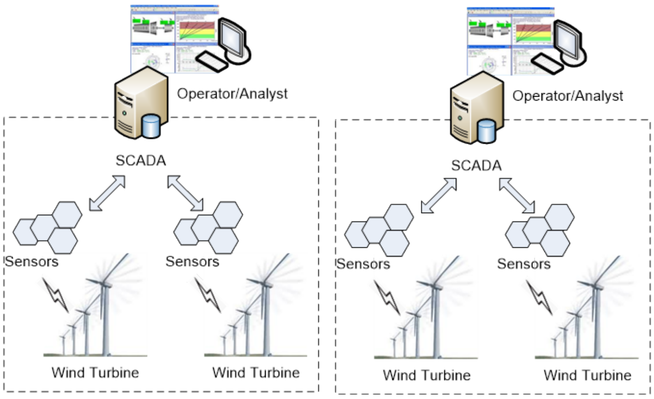


Fig. 1. Architecture of a typical on-site wind power control and monitoring system.

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. In [3], 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. A similar approach for SLA enforcement is presented in [4] 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 [5], an SLA Manager is presented alongside with proposed techniques for VM selection and allocation during live migration of VMs. Using the proposed 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.

2.2 Performance and Energy-efficiency Tradeoff in Cloud Data Centers

In data centers, huge energy consumption is caused not only by the physical servers and other hardware, but also by the required power supply and cooling system. As such, acquiring the most efficient components and coming up with good architectural designs and configurations regarding energy demand, availability, and performance are deemed crucial [6]. In [7], they presented a cloud infrastructure that combines on-demand allocation of resources with opportunistic provisioning of cycles from idle cloud nodes which aims to improve the utilization of Infrastructure Clouds. The VM placement algorithms of IaaS providers need to know the current and future energy efficiency at different levels. The work in [8] provides a mathematical formulation for the said concern, as well as the design of a CPU utilization estimator used to calculate the needed forecasts. It is found that proper adjustment of the configuration parameters leads to considerably improved estimator precision. In [9] 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. They found in [10] that although the deployment of energy-efficient hardware is a crucial step, getting rid of underutilized servers is a far more effective approach. Thus, the scalability of monitoring and managing a cloud data center is improved by taking advantage of workload data which is descriptive of a VM's behavior.

2.3 Condition Monitoring Systems

To date, vibration analysis remains the most popular condition monitoring technology employed in WT especially for rotating equipment [11]. It is well-suited for monitoring the gearbox, bearings, and other selected WT elements. The measurement and interpretation of acoustic emission parameters for fault detection in ball bearings has been demonstrated at different speed ranges in [12]. From a case study of a WT gearbox in [13], vibration may possibly not be evident while faults are developing, but analysis of the oil can provide early warnings. For lifetime forecasting and protection against high stress levels especially in the blades, stress measurement is another viable option. In [14], an assessment of strain gauge signal interpretation from strain gauge sensors installed on the blade has been performed in order to adjust calibration practices and sensor selection. Lastly, thermography is often used for monitoring electronic and electric components and identifying failure. The technique is only applied off-line, and often involves visual interpretation of hot spots that arise due to bad contact or a system fault. The work in [15] used infrared cameras to visualize variations in blade surface temperature and indicate cracks as well as places threatened by damage.

3. Data Center for Cloud-Based Wind Farm CMS

3.1 Cloud System Architecture

As shown in **Fig. 2**, the proposed cloud system is presented as a three-layer structure composed of the Service Provisioning, Resource Management, and Virtual Machine layers. Each layer includes various components which contribute to the functionalities of the system.

The core of the provisioning mechanism resides within the Resource Management Layer. Specifically, the Load Predictor and VM manager are responsible for the load prediction and efficient management of the resources provided to the users. At the bottom, the Virtual Machine (VM) Layer represents an abstraction of the functionalities inherent to each virtualized platform. Each VM instance is configured and provided with the necessary components according to the specifications of the user. As decided by the VM Manager, VM's can be reconfigured, redeployed, and replicated according to the demands. At the physical level, each server has its own local resource manager which keeps track of its resource consumption making sure that it meets the expected performance thereby keeping the quality of service.

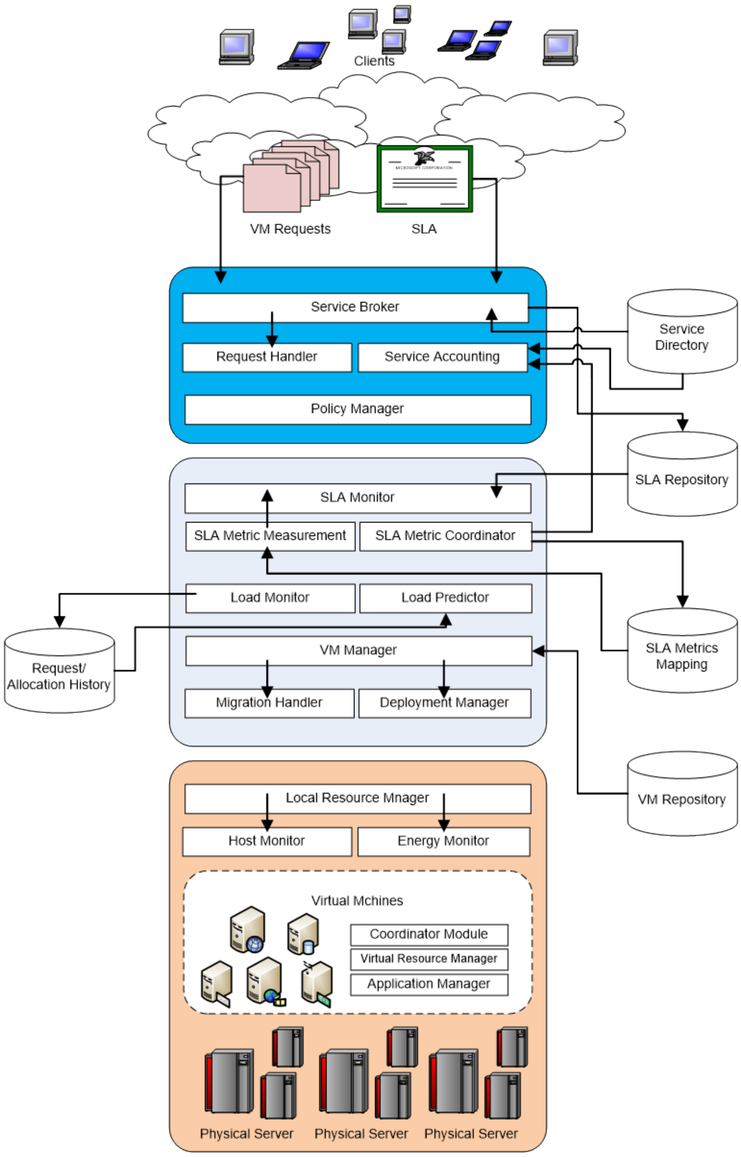


Fig. 2. The architecture of the proposed cloud system.

As shown in **Fig. 3** The proposed strategy consists of various components that perform equally important tasks towards the 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 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*.

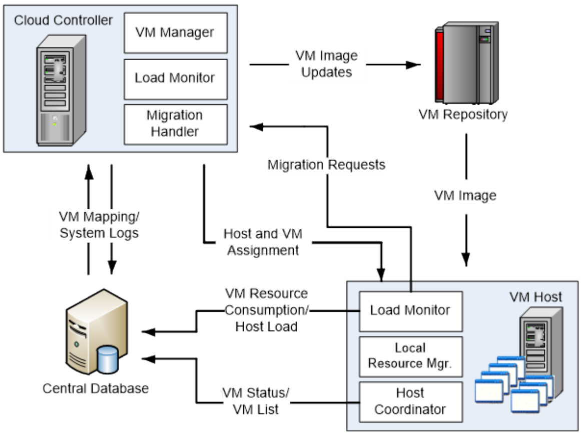


Fig. 3. The architecture of the cloud's optimization strategy.

3.2 Datacenter Monitoring Strategy

In a datacenter, the VMs experience highly dynamic workloads as reflected by the CPU usage which varies over time. Based on the findings of Intel Labs [16], 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 [17, 18] show that 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.

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 $Util_{dc} = CPU_u / CPU_{dc}$.

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. 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.

```
Performance Monitoring Strategy
For each DC interval {
$CPU_u$ ← 0 //initialize dc utilization
  For each Host h {
    $CPU_u$ =+ $CPU_h$ //update cpu utilization
    $CPU_{dc}$ =+ $maxCPU_h$ //update dc capacity
  }
  $Util_{dc}$ ← $CPU_u/CPU_{dc}$
         //get dc utilization percentage
If (strategy = Power) {
  If ($Util_{dc}$ = max) {
   If GetSLAV >= threshold
     SwitchStrategy(SLA)
      // switch strategy to sla-aware
  }
}
If (strategy = SLA) {
  If ($Util_{dc}$ = min) {
   if $Pwr_{eff}$ <= threshold
     SwitchStrategy(Power)
      // switch strategy to power-aware
  }
}
}
```

Algorithm 1. The monitoring strategy.

3.3 Cloud-based CMS Architecture

As for the practical application of the proposed cloud system's service provisioning, we aim to implement a fault detection approach [19] which combines intelligent methods and data mining to accurately reflect the deteriorating condition of a wind turbine and to indicate the components that need attention. Using SCADA data, we will extract operational status patterns and develop a rule repository for monitoring wind turbine systems. This will enable wind farm operators to detect the deteriorating condition of a wind turbine as well as to explicitly identify faulty components. The architecture of the Cloud-based Wind Farm Condition Monitoring System is shown in **Fig. 4**.

Taking advantage of the Cloud computing paradigm will dramatically improve remote monitoring by providing highly-scalable services and platforms for carrying out classification, diagnosis, and prediction operations. Because cloud-based services can reside on many platforms, cloud-based CMSs can be accessed by multiple users working on different types of operating systems and devices. Wind farm operators can use the cloud model to store and process large quantities of data collected from sensors deployed all over wind turbines. Cloud platforms could be a large contributor in the design and development of software that would enable more effective use of SCADA and remote monitoring systems. As indicated by their architecture, cloud data centers can accommodate large-scale data interactions that take place on several wind farms and are better structured than centralized systems to process the huge, persistent flows of data. Instead of having in-house computing resources for each wind farm owner, at the very least, a single cloud service provider can deliver services via its PaaS and SaaS delivery models by consolidating its virtualized resources. By using virtualization, transparency and uniformity of systems can be easily imposed across several wind farms. As an added advantage, such feature would allow collaboration among experts inside and even outside the field of wind energy.

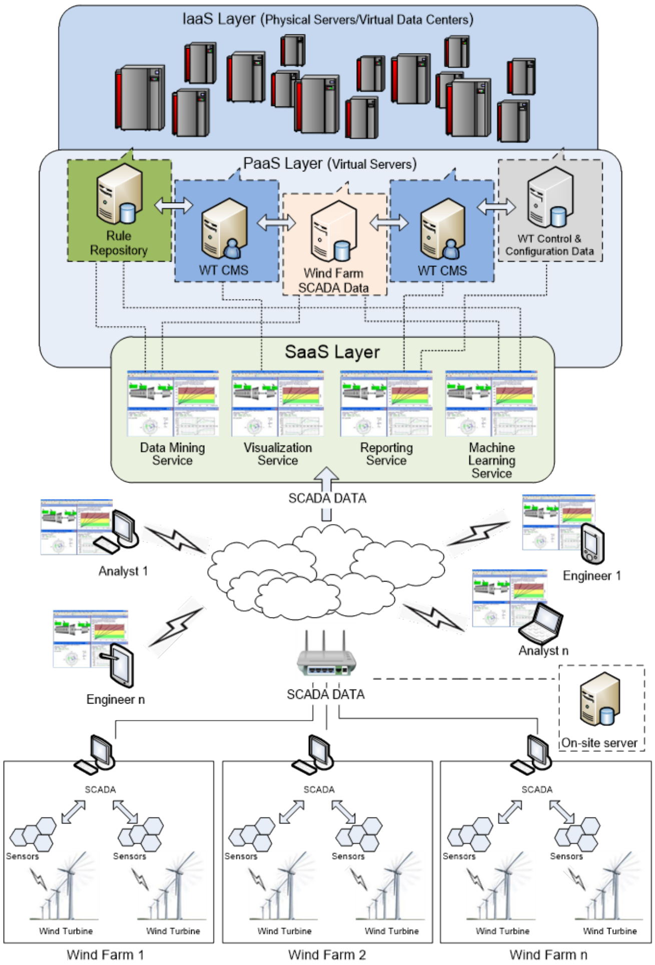


Fig. 4. Architecture of the Cloud-based Wind Farm CMS.

3.4 WT Fault Detection Approach

The fault detection system presented in this work is applicable to wind turbines equipped with SCADA system. As shown in **Fig. 5**, the fault detection scheme is composed of various procedures which include the acquisition and pre-processing of SCADA data, clustering and classification, itemset generation, frequent pattern mining, rule generation, and fault detection [20-21].

During the operation of a wind turbine, a normal behavior can be characterized by its power curve. An apparent advantage of using normal behavior models to monitor wind turbine signals is that no prior knowledge about the signal behavior is necessary; thus, normal behavior SCADA data were used in the initial stage of clustering and classification. It is crucial that we first identify outliers and eliminate them from the dataset so as to assure its integrity. Since data mining algorithms construct models using large datasets, it requires data preprocessing. Thus, a significant portion of the analysis time may be spent on data sampling, parameter selection, and other data analysis tasks.

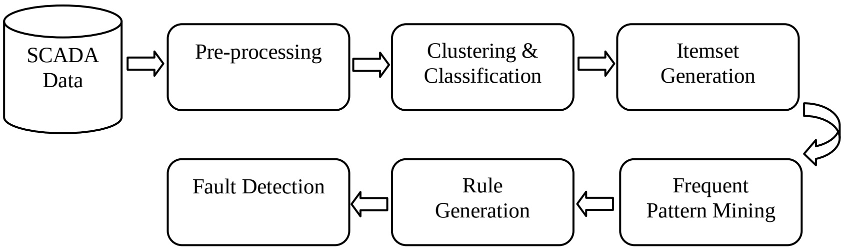


Fig. 5. The Wind Turbine CMS design.

4. Implementation and Evaluation Results

4.1 Implementing the Cloud Infrastructure

In order to evaluate the proposed approach in a realistic setting, we implemented a prototype in a testbed environment. The system setup is composed of 3 VM Hosts, 1 VM Server, and 1 Cloud Controller which has 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

The VMs were provided with bridged virtual NIC, 256 MB RAM, and 1 CPU core. A local network is also set up for the test environment. The purpose of setting the virtual network adapters as bridged is to enable them to perform networking with other physical machines aside from their host. The cloud system is implemented using the C# programming language while MySQL Server is used for its database. For the VM hypervisor, Oracle VirtualBox is utilized. 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. As shown in **Fig. 6**, each VM Host is assigned with 9 virtual machines running Windows7 installed in their respective virtual hard drive.

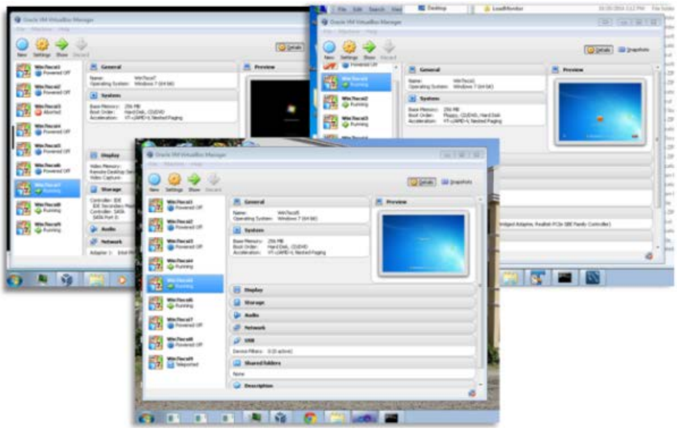
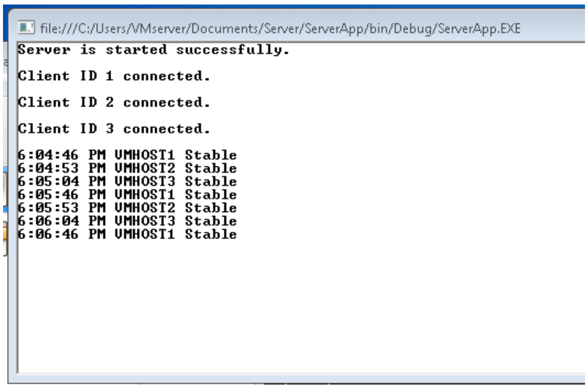


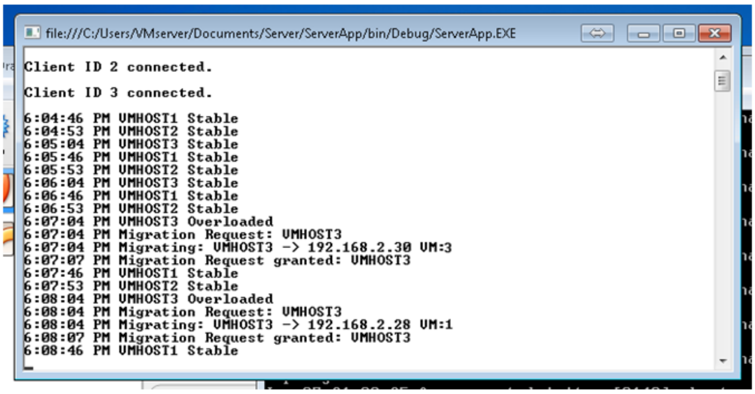
Fig. 6. Virtual Macines deployed to Servers.

In **Fig. 7**, the *VM Manager* is started and waits for the *VM Servers* to establish a connection. As soon as a connection is established, the *Load Monitor* of the host machines will start sending notification messages informing the *VM Manager* of their status. In **Fig. 8**, it is shown how the *VM Manager* handles a migration request in case of host overloading.


file:///C:/Users/VMserver/Documents/Server/ServerApp/bin/Debug/ServerApp.EXE

```
Server is started successfully.

Client ID 1 connected.

Client ID 2 connected.

Client ID 3 connected.

6:04:46 PM VMHOST1 Stable
6:04:53 PM VMHOST2 Stable
6:05:04 PM VMHOST3 Stable
6:05:46 PM VMHOST1 Stable
6:05:53 PM VMHOST2 Stable
6:06:04 PM VMHOST3 Stable
6:06:46 PM VMHOST1 Stable
```

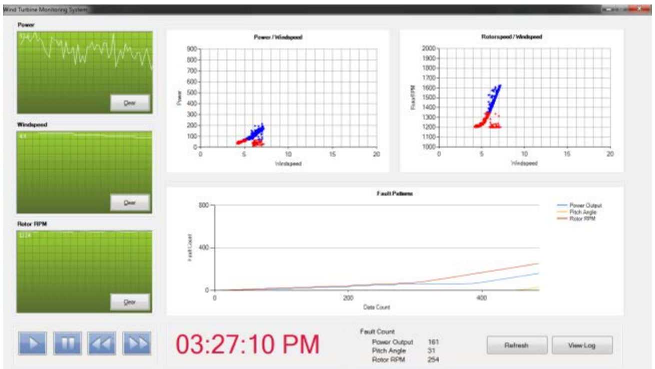
Fig. 7. VM Manager receiving host logs.


file:///C:/Users/VMserver/Documents/Server/ServerApp/bin/Debug/ServerApp.EXE

```
Client ID 2 connected.

Client ID 3 connected.

6:04:46 PM VMHOST1 Stable
6:04:53 PM VMHOST2 Stable
6:05:04 PM VMHOST3 Stable
6:05:46 PM VMHOST1 Stable
6:05:53 PM VMHOST2 Stable
6:06:04 PM VMHOST3 Stable
6:06:46 PM VMHOST1 Stable
6:06:53 PM VMHOST2 Stable
6:07:04 PM VMHOST3 Overloaded
6:07:04 PM Migration Request: VMHOST3
6:07:04 PM Migrating: VMHOST3 -> 192.168.2.30 VM:3
6:07:07 PM Migration Request granted: VMHOST3
6:07:46 PM VMHOST1 Stable
6:07:53 PM VMHOST2 Stable
6:08:04 PM VMHOST3 Overloaded
6:08:04 PM Migration Request: VMHOST3
6:08:04 PM Migrating: VMHOST3 -> 192.168.2.28 VM:1
6:08:07 PM Migration Request granted: VMHOST3
6:08:46 PM VMHOST1 Stable
```

Fig. 8. VM Manager handling a migration request.

After the cloud system implementation, the software to be deployed in its SaaS layer is developed. In Fig. 9, a test run of the WF CMS is shown. As soon as SCADA data is fed into the CMS, it is plotted on the graph to immediately provide a visualization of the wind turbine's condition. On the left side, the probe shows the fluctuation of parameter values for the different components. At the top, the main charts show the actual visualization of the Power Output and Rotor RPM relative to the Windspeed. The color assigned to each plotted point corresponds to the cluster to which they belong. In the figure, a red cluster is composed of values that indicate a fault. At the bottom, the graph shows the increasing number of component faults detected by the system.



Fault Count
Power Output 161
Pitch Angle 31
Rotor RPM 254

Fig. 9. Test run of the WF CMS.

The WF CMS is then deployed on the proposed cloud infrastructure. After the entire system is completely set up, the VM Manager and the VM Servers were started, and the virtual machines hosted by the servers were simultaneously activated. Multiple instances of the WF CMS application were also run and accessed from different devices as shown below:

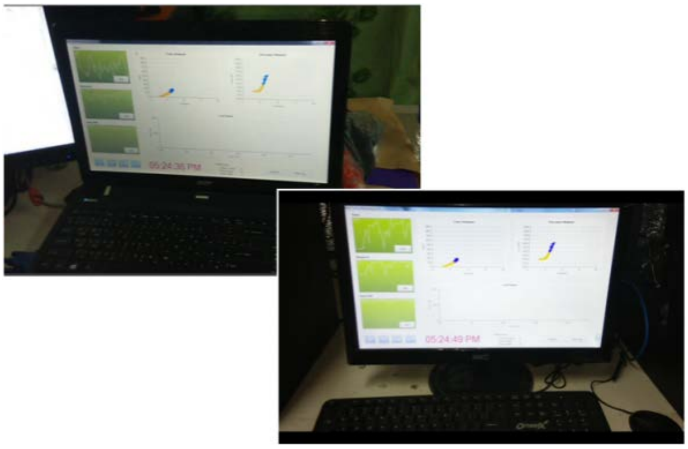


Fig. 10. Instances of the WF CMS accessed remotely from a laptop and a desktop PC.

Fig. 11. Instances of the WF CMS accessed remotely from an iOS and Android device.

4.2 Evaluation Results

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; each of them is respectively designed to handle performance-awareness and energy efficiency. *MinCPU* works by migrating a 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 CPU consumption, performance degradation of the cloud system is minimized while at the same time service disruption on the part of the client is barely noticeable. As for *MinAgg*, the strategy intends to minimize the overhead by selecting the VM with the smallest average aggregate of the compute resources utilized for a certain monitoring period. This strategy makes 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.

From an initial experiment, we would like to emphasize that the said techniques, although aimed to enforce high QoS and low power consumption respectively, 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. 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.

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

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 our *Switch* strategy. As for the average host power, our proposed approach is also able to consume the lowest average power per host. 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.

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 granted 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.

	MinCPU	**MinAgg**	**Switch**
Migrations Granted	215	272	322
Migrations Not Granted	46	64	68
Migration Success %	97.86	97.65	97.89
SLA Violation %	2.605	3.015	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 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. Our proposed dynamic switching strategy was able to outperform those which only focus on a single metric pertaining to Performance and Power. As for the application of the proposed cloud system, another objective of this effort is to develop a Wind Farm CMS that is virtual, yet self-contained. Automation of CM and diagnostic systems will become important as Wind Farm operators acquire a larger number of turbines and manual inspection of data becomes impractical. Taking advantage of the Cloud computing model significantly improves remote monitoring by providing highly-scalable services and platforms that are accessible in a ubiquitous manner.

References

[1] J. Baliga, R.W.A. Ayre, K. Hinton, R.S. Tucker, "Green cloud computing: balancing energy in processing, storage and transport," in *Proc. of the IEEE 2011*, Vol 99, No. 1, pp.149–67, 2011. Article (CrossRef Link)

[2] J. Knezevic, "Reliability, maintainability and supportability engineering: a probabilistic approach," McGraw Hill, 1993.

[3] 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)

[4] M. Macias and J. Guitart, "Client Classification Policies for SLA Enforcement in Shared Cloud Data-centers," in *Proc. of The 12th IEEE/ACM Int. Symp. on Cluster, Cloud and Grid Computing*, 2012. Article (CrossRef Link)

[5] 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.

[6] G. Schomaker, S. Janacek, and D. Schlitt, "The energy demand of data centers," *ICT Innovations for Sustainability*, Springer, pp. 113–124, 2015. Article (CrossRef Link)

[7] P. Marshall, K. Keahey, and T. Freeman, "Improving utilization of infrastructure clouds," in *Proc. of The 11th IEEE/ACM Int. Symp. on Cluster,* Cloud and Grid Computing, 2011. Article (CrossRef Link)

[8] J. Subirats and J. Guitart, "Assessing and forecasting energy efficiency on Cloud computing platforms," *Future Generation Computer Systems*, 2015. Article (CrossRef Link)

[9] 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.

[10] F. Elijorde and J.W. Lee, "Attaining Reliability and Energy Efficiency in Cloud Data Centers Through Workload Profiling and SLA-Aware VM Assignment," *International Journal of Advances in Soft Computing & Its Applications, Vol.* 7, No.1, pp. 41-58, 2015.

[11] Z. Hameed, Y.S. Hong, Y.M. Choa, S.H. Ahn, and C.K. Song, "Condition monitoring and fault detection of wind turbines and related algorithms: a review," *Renewable and Sustainable Energy Reviews*, pp. 1-39, 2009. Article (CrossRef Link)

[12] N. Tandon, B.C. Nakra, "Defect detection in rolling element bearings by acoustic emission method," *Journal of Acoustic Emission*, Vol 9, No. 1, pp. 25-28, 1990.

[13] S. Leske and D. Kitaljevich, "Managing gearbox failure," *Dewek. Dewi Magazine*, No. 29, 2006.

[14] E. Morfiadakis, K. Papadopoulos, and T.P. Philippidis, "Assessment of the strain gauge technique for measurement of wind turbine blade loads," *Wind Energy*, Vol. 3 No. 1, pp. 35-65, 2000. Article (CrossRef Link)

[15] M.A. Rumsey and W. Musial, "Application of infrared thermography nondestructive testing during wind turbine blade Tests," *Journal of Solar Energy Engineering*, 2001. Article (CrossRef Link)

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

[17] 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)

[18] 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)

[19] F. Elijorde, S.H. Kim, and J.W. Lee, "A Wind Turbine Fault Detection Approach Based on Cluster Analysis and Frequent Pattern Mining," *KSII Transactions on Internet and Information Systems*, Vol. 8, No. 2, pp. 664-677, 2014. Article (CrossRef Link)

[20] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc of the 5th Berkeley symposium on mathematical statistics and probability*, 1967.

[21] J. Han , J. Pei , and Y. Yin, "Mining frequent patterns without candidate generation," in *Proc of ACM SIGMOD international conference on management of data*, 2000. Article (CrossRef Link)

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 (now Iloilo Science and Technology University), 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.

Sung-Ho Kim received his B.S., M.S., and Ph.D. degrees in Electrical Engineering from Korea University in 1984, 1986, and 1991, respectively. Currently, he is a professor at the Department of Control & Robot Engineering in Kunsan National University, Gunsan City, South Korea. His research interests include wind turbine system, fault detection and diagnosis and process control system.

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.