Integrated Data Management System for Data Center

Debasis Dash[1], Juthika Dash[2], Younghee Park[3]*, Jerry Gao[4]
Computer Engineering Department
San Jose State University
California, USA
{debasis.dash[1], juthika.dash[2], younghee.park[3], jerry.gao[4]}@sjsu.edu

Abstract— **Cloud computing has become the most popular computing architecture in the world. The multiplicity of available platforms for end users makes it very inconvenient for service providers to manage, calling for a unified approach for resource management and security in virtualized data centers in order to achieve seamless cloud management. In this paper, we present data center management as a service (DMaaS) as such a unified approach, using multi-tenant software. Virtualized data centers pool cloud resources used to meet enterprise needs, mostly related to storage and processing. The proposed work provides a service model for the cloud infrastructure and supports multiple tenants to connect their data center management in a shared and secure manner. Our service manages disparate virtual environments using a single interface.**

Keywords- Datacenter, OpenStack, VMWare hypervisor, server virtualization, cloud, SaaS, analytics, recommendation, access control list

I. INTRODUCTION

Major challenges arise in providing security and resource management in virtualized data centers, which pool computing and networking resources. Some papers have explained issues and benefits involving the use of virtualization in data security services in cloud computing [6] [7]. Some of the security issues faced while managing virtual infrastructure include maintenance, configuration, and updates for software, and generally maintaining consistency and integrity [6]. High performance and failure recovery in data centers are essential since so many users depend on the data center computing power. The complicated underlying services and systems in data centers create challenges for heterogeneous data center infrastructure. It is hard to manage system resources and monitor cloud computing in such diverse data centers.

The challenges of monitoring data center performance [11], which is itself a synergy of server performance, application performance, and security threats, multiply when disparate virtualization platforms and monitoring tools are used. In addition, an on-demand, mandatory security monitoring system in the virtualized datacenter environment is required, one that won't depend on pre-installed guest components. The ultimate aim is to utilize, deploy, work, and monitor the security in a large-scale virtualized datacenter cloud environment, such as the IaaS cloud [1]. The need for virtualization administration and automation has been emphasized [3], requiring a unified approach to data center management and security monitoring. Our proposal offers a unified solution to manage, integrate, and monitor a heterogeneous datacenter.

In this paper, we propose a data management system called DMaaS (Data Management as a service) in order to integrate different data center environments into a single platform for easy configuration and efficient management. The proposed system is composed of different service layers and critical subsystems to provide a single platform to manage different data centers depending on user requirements. The proposed solution provides an abstraction interface to manage data centers with disparate resources in a virtualized infrastructure. Users can use a single platform to monitor and manage their data centers with a high degree of usability. The proposed system offers resource recovery and management, user pattern recognition, and recommendation systems to provide best-effort services for end users. Through monitoring system resources and user activities, our system provides a wide range of metrics for VM's in the data center. Our experimental results demonstrate that our system can neutralize overhead in a single data center, and efficiently assign system resources to different data centers by using various profiling techniques in terms of systems and users.

Section II discusses our motivation for engaging in this work. Section III presents the proposed system with a detailed description of system architecture. Section IV shows experimental results with our implementation testbed. Section V discusses directions for future work. Section VI explains related work. Lastly, our conclusions are in Section VII.

II. MOTIVATION

Virtualization has led to a dramatic rise in the developmen of data center technology, especially in cloud platforms. The modern cloud environment offers different hardware configurations designed for classes of applications or user requirements. A typical cloud environment supports multiple generations of hardware simultaneously. Heterogeneous advanced technologies make it difficult to manage various data center services and to efficiently utilize data center resources. Therefore, data center management software must be independent of any platform. An integrated service management system to unify heterogeneous data centers is needed to minimize operation costs and provide high usability.

Figures 1 and 2 show two cases in common use to illustrate the motivation behind our research. Figure 1 shows two companies, A and B, which have their own virtualized environments, each in their own data center. If company A integrates company B's data center, with its different

*This author is a corresponding author.

DOI reference number: 10.18293/SEKE2016-128

infrastructure, company A must first update its existing data center software to create a unified platform that can seamlessly manage both data centers without changing the underlying infrastructure. For such an integrated single data center working with disparate virtual infrastructures, it is critical to have a common monitoring and managing platform.

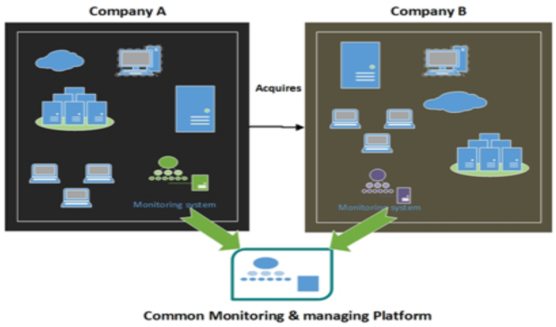

Figure 1. Different virtual environments for two companies

Figure 2 illustrates another case. A company has multiple data centers in different geographical locations with different virtual environments. Its multiple data centers have different capabilities and different sets of management environments. Managing all the data centers in a unified manner is challenging, but necessary, in order to provide efficient resource management and to avoid security problems. Therefore, this paper proposes a unified management system to manage and control multiple data centers from a single piece of software and a single location.

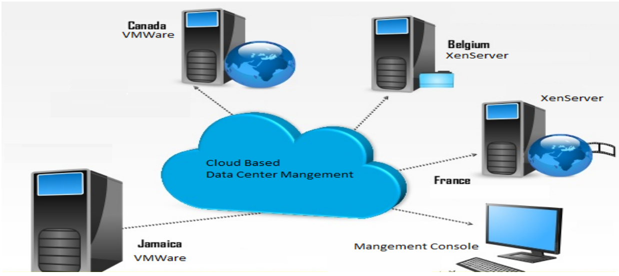

Figure 2. A company has multiple datacenters in different geographical locations with different IT environments.

III. SYSTEM ARCHITECTURE

A. Overview

We propose datacenter management as a service (DMaaS) for streamlined IT operations at heterogeneous data centers. The proposed solution is able to process a large amount of data in a virtual system that monitors resource usage and threats in the data center. The solution provides a unified platform implemented in Multi-tenant Service Oriented Architecture, which means that a group of users share common access to every tenant including every tenant's data, configuration, user management, and individual tenant functionalities.

Multi-Tenant service oriented architecture provides a cost effective highly scalable software model. This architecture reduces the user's initial capital expenditure (capex) by sharing the cost of infrastructure and maintenance. Service oriented architecture (SOA) helps the model to scale easily as each functional module is loosely coupled with the others. The distributed nature of SOA provides high availability to the system. The proposed solution utilizes this architecture to achieve a distributed, scalable, cost-effective software model.

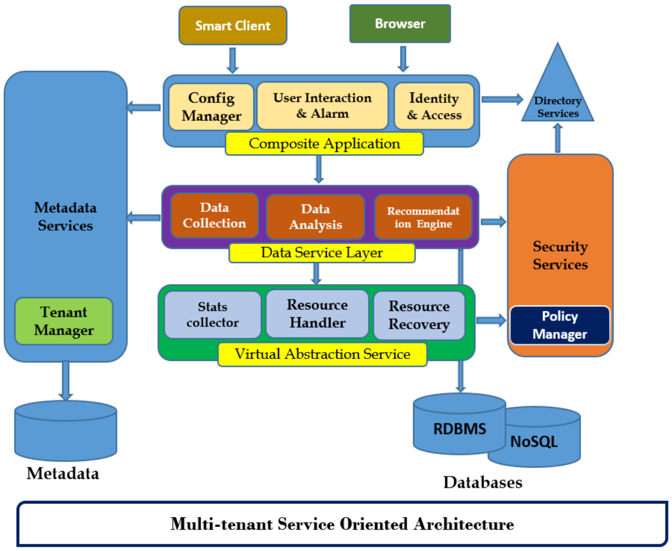

Figure 3. System Architecture for DMaaS (Datacenter Management as a Service)

Figure 3 presents the system architecture we propose to achieve our goal, our Datacenter Management as a Service (DMaaS). The proposed system consists of different service layers: metadata, analytics, directory, and security services, with frontend user interfaces including a smart client and a browser. To support these complex services, there are three main system components: a presentation system, a business process service system, and a data analysis system. The proposed solution comes under the SaaS (Software as a Service) model. We have designed our model to utilize this service-oriented architecture in a multi-tenant environment. The following sections describe each individual component in detail.

B. Various Service Layers

This proposed system architecture has various service layers: metadata, analytics, directory, and security services, in order to integrate heterogeneous data centers into a single unified platform to support streamlined data center management.

The ***Metadata Services*** layer is provided by the tenant manager, a major element to ensure sanity among users from different enterprises. This provides a primary means of customizing and configuring a new unified cloud environment for each user while collecting users' requirements from different infrastructures. Based on user input, our system provides a service to monitor parameters installed by end users. It supplies a user-friendly control environment enabling users to configure the service and to monitor workflow and business rules.

The **Analytics Services** layer performs data cleansing and analysis based on Spark, an open source cluster computing framework. It interacts with the business process service system and data storage to keep track of system resource usage with various large data from systems and networks. Through data analysis, the proposed system can provide a single unified data center for each user by using the system and network resources efficiently through the results obtained from data analysis.

The ***Directory Services*** layer provides access control functionality to the service based on role-based access control. In other words, system requirements are affected by different job privileges for each user. Each user has a profile based on roles and responsibilities for configuration management and based on web service recommendations. This provides user authentication and authorization based on different user profiles.

The ***Security Services*** layer monitor users' activities for system and network resources. Through user profiles and usage patterns, it provides a security monitor to check any malicious abnormal behavior in the allocated data center. It monitors system activities through memory usage, CPU usage, file access patterns and access permission, and network usage with protocols and port numbers. Through this security services layer, our system can block malicious users and replace compromised configurations into a clean state environment by configuring a new data center on the fly. It can be achieved since all the data for each user is saved in data storage.

The front-end user interfaces (i.e. smart client and browser) utilize these different service layers to visualize resource usage, anomalies, and statistics provided from the data center. The smart client is a mobile application and the browser is a web client to connect into the data center management service in a secured manner.

C. *Three Main Systems*

To support these integrated services, there are three main system components: a presentation system, a business process service system, and a data analysis system. First, the presentation system provides access to service configuration, alarms, and recommendations using modern web services. It is integrated to provide data visualization in a most effective manner. Second, the business process service system provides the functionality to connect to the datacenter to collect system logs, network log, and interaction with hypervisors for collecting usage metrics. The recommendation engine is a part of this system and interacts with the analytics service layer to query required data and provide recommendations using these data. Lastly, the data analysis system analyzes the system log, resource usage patterns and user activities on Spark to provide better recommendations to end users while maintaining efficient data management.

These three subsystems have important components to achieve DMasS: resource discovery, resource management, role-based access control, usage pattern recognition, and a recommendation system as described below.

Resource discovery supports scale-out of the data center. Application modules work as a controller with a global view of resource availability and usage. When a user requests computing resources, if the local datacenter doesn't have sufficient resources then it can acquire resources from a different datacenter under the same enterprise. It will use overlay technology to stitch both virtual systems. Hence, users can leverage the availability of remote resources.

Resource management plays an important part by providing a hyper-call abstraction to the application service. Virtual Abstraction Interface (VAI) implements hypervisor-specific implementation for translating an application request to the appropriate hyper call. Major functionality involves resource creation, maintenance, and extraction of resource usage patterns. VAI sends all the requested data using a northbound interface towards the application module.

Role-based access control provides the access control list for role-based usage. The system implements various user levels and assigns access control based on roles. This provides an additional level of security to the hosted service. There are three user roles: high-class, medium-class and low-class. Users who compute intensive scientific flow analysis and computing programming belong to the high class that requires high memory and CPU resources. However, users having common jobs working on documents, checking emails, and surfing can be classified into the low class. These users usually need smaller resources compared to high-class users. Others are classified into medium-class users. This class system is based on user profiles from past or current activities through user pattern recognition.

User Pattern Recognition is useful for efficient usage of resources available in the resource pool. It goes hand in hand with the recommendation system of DMaaS described in the next section. It can detect abnormal users if any resources are being under or over-used by a particular user. This information is given to the recommendation system, which can provide recommendations for better resource usage for increased performance standards. This module is useful in tracking user activities, and analyzing resource usage. The information is also useful in training the recommendation system to provide accurate and useful recommendations.

The ***Recommendation System*** for DMaaS has been designed to improve the user experience. Two main features of the recommendation system are providing recommendations for better resource usage and providing recommendations for a new user to ease the creation of Virtual Machines (VM) depending on his/her preferences (processing, memory, network bandwidth need). It generates recommendations for a user depending on the user's usage of VM depending on user profiles. For example, some users utilize VM for surfing the web in the low class, or other users need to have VM for intensive computing in the high class. These extreme cases must each be handled differently by our proposed system for efficient resource management and better service in the data center. To achieve our goal for best-effort services, we record user usage in the data center as a training model. Based on training data, the proposed system provides a recommendation to match user preferences with current system status. When a user input is given, the proposed system makes a recommendation based on the highest similarity between the current input and the profiled usage. The similarity is computed with *Jaccard Index*. We extract a set of features, such as memory usage, CPU usage, duration and frequency to use the data center, job objectives depending purpose to use the data center and geographical locations. To compute the values of Jaccard index, we assign a numerical value from one to ten depending on scale for each feature.

A cold start problem would occur if there were absolutely no users registered for the DMaaS system and hence the recommendation system cannot generate any recommendations. This potential problem will be addressed by asking users generic questions pertaining to their preferences of activities performed on the computer, such as how frequently they carry out computer intensive work. Asking these questions will help in developing their user profiles, which will be used in the future for training the model and generating recommendations.

Finally, our proposed solution provides many visual charts for easy understanding of resource usage patterns. The web service includes various graphical interfaces to create and manage virtual machines, data centers, adding new resources, logging, utilization, and recommendations.

IV. Implementation and Experiments

A. *Implementation*

The test setup is comprised of two systems with a quad-core processor with 16GB of RAM. One of the systems is installed with VMWare ESXi and other system is installed with OpenStack software. These two systems are used to simulate individual data centers. Two cloud instances are required to run web service and data analysis for the recommendation system. Proof-of-Concept is implemented in the three following major components: Web Service, Virtual Abstraction Interface (VAI) and Recommendation Engine.

Web service includes an access control list for role-based access to the web service, using LDAP. It receives user requests, processes them, and creates southbound calls. It also visualizes resource utilization, recommendations, and logging. It is implemented on Ruby in Rails, LDAP, high charts and cloud database as a part of the web service development. VAI decouples the hypervisor-specific interface from the application layer. It consumes java-based VMWare. It interacts with API and OpenStack. Message brokers like RabbitMQ are used for inter-process communication. Lastly, we have implemented our recommendation engine using java language. This module consumes data from VAI to train the recommendation engine.

B. *Experiments*

The system will be tested for accuracy, security, efficiency and performance. The accuracy of the system mainly depends on the recommendations provided, the resource allocation and keeping track of over and underuse of resources. The security feature of the system defines controlled access to the system, i.e. only authorized and registered users can access the system. To implement security features, DMaaS has incorporated the use of Access Control Lists. The system will be evaluated on efficiency and performance. Efficiency can be determined by how quickly the system can adapt to changes and how quickly the system responds to user requests. Handling of failures will also determine the efficiency of DMaaS. Performance can be determined by how efficient the resources are from the resource pool used.

A few experiments have been conducted to evaluate these systems. Currently, each sub-module has been evaluated separately. One such experiment was conducted to check the accuracy and performance of the Recommendation System sub-module. The experiment was to verify whether the system was getting trained appropriately. A few users were added to the system with their preferences about VM usage. Each user request was considered as a training rule and each such training rule was given a score. Whenever users selected the exact same rule and the exact same VM type, the score was increased. A threshold was maintained. When the score of a rule reached the threshold, it was considered that the particular VM type associated with the rule was popular and hence could be recommended to other users with similar preferences. The experiment was successful for a small number of users. More user data will help in increasing the accuracy of this sub-module.

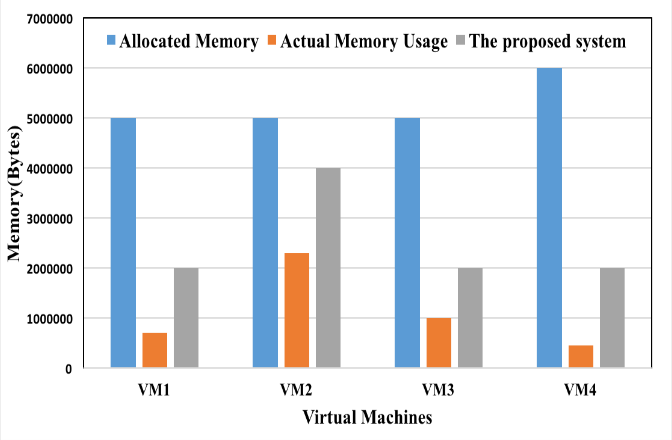

Figure 4 - Memory utilization vs memory allocation

Figure 4 shows the memory utilization for individual virtual machines versus allocated memory inside the data center. As we can see that some virtual machines were underutilized while a few were closer to the maximum allocation. In other words, the allocated memory prior to using our system showed less than a 30% utilization, compared to the allocated memory. However, the proposed system optimizes the resource allocation in terms of memory as shown in Figure 4. The proposed system uses this information to recommend better resource distribution for current or future users. We can increase more than 70% utilization while reducing the memory size depending on actual memory usage patterns for end users.

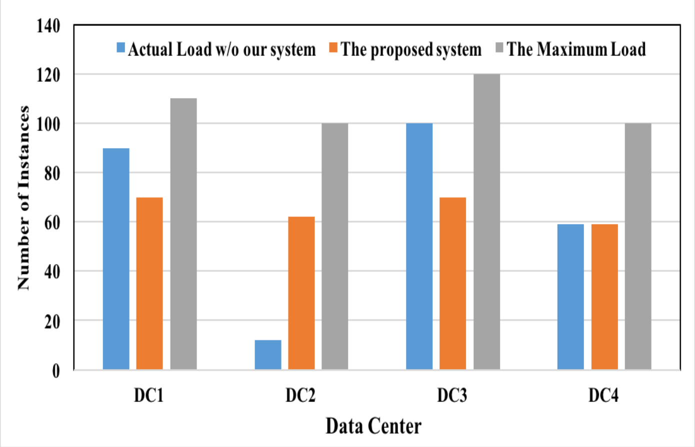

Figure 5 - Datacenter utilization (based on number of VMs)

Figure 5 shows data-center level workload distribution. The chart consists of four data centers identified as DC1 to DC4. The blue bar indicates the current actual load at each data center. The orange bar represents the workload recommended for our proposed system. The gray bar indicates the maximum workload for user requests that each data center can accomodate. The proposed system can recommend current or future users to move some of their workload from the DC1 and DC3 data centers to the DC2 data center to balance the workload across all data centers. Real-time monitoring results can be immediately applied to the current system in the data center for balanced workload distribution.

The proposed system monitors system resource utilization in terms of CPU, memory, disk, IO and network bandwidth utilization. This information enables the proposed system to understand average usage for different users and different virtual machines. Figure 6 depicts memory usage among various virtual machines for a given data center. The mentioned parameter data types are collected from all the virtual machines running in the individual data center and displayed after user login to the system. In addition, it also gave the resource utilization for the various systems.

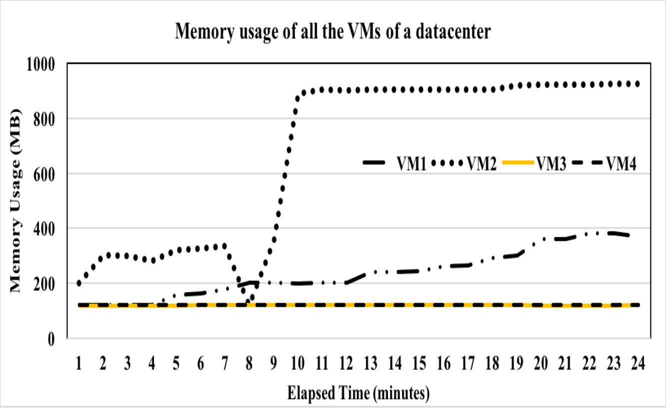

Figure 6. Data center memory usage

V. Discussion and Future Work

DMaaS (Data Management as a Service) is a cloud-based service that can be deployed in the public cloud, as illustrated in Figure 7. This figure demonstrates a typical deployment scenario. Two companies, Company A and Company B, utilize our DMaaS platform. The companies are able to create and monitor VMs and get recommendations for them across different hypervisors like VMWare, OpenStack, and Xenserver [12] using our proposed solution. The data center administrators of both these companies find it easy to maintain security, monitor activity, and managing workload using recommendations through DMaaS. Major components required for the deployment are a cloud computing service, load balancer, database, and firewall.

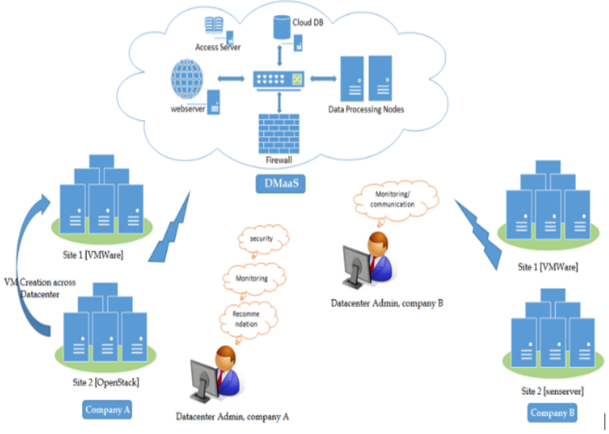

Figure 7. An illustration of DMaaS Deployment

The Cloud computing service uses multiple cloud instances. This includes both small and large cloud instances. All the micro instances will be used for web service hosting, the virtualization layer, and security analysis. The proposed system in this paper utilizes a large instance for the data analysis and recommendation system. The proposed system also requires a load balancer for balanced workload distribution. The load balancer can configure new instances if the traffic rate increases. It provides easy scaling with respect to traffic and resource load.

A firewall can be used along with our security features to add security towards the enterprise data pipeline,. This will provide additional security for accessing data service from the enterprise pipeline towards DMaaS service.

In future work, we will work on dynamic adjustment features by using virtualization techniques, such as resizing memory and migrating data, to shed further light on our analysis results and to recommend a better environment for an efficient and streamlined data center management system. In addition, the proposed system will be upgraded through fine-grained recommendation systems by using machine-learning techniques and resource monitoring systems by using logging technologies and computer forensics. Through multiple use case study, we will thoroughly evaluate our proposed system based on benchmarking tools.

VI. Related Works

A pre-existing monitoring and reporting tool for VMWare VSphere [9] and Microsoft Hyper-V was instrumental in providing a detailed case study of monitoring of virtual machines and environments, notification and alert mechanisms for failure and backup, and troubleshooting capabilities to ensure that the system remains working and is up at all times. This study helped to describe the challenges that can be faced by a system or system administrator while monitoring a VM or data center. The tool not only carries out monitoring tasks but also provides alerts when any performance issues are encountered. This is an effective tool for monitoring as well as for failure recovery and system backup [11].

Monitoring of critical performance metrics and providing VM statistics to the user is an essential performance monitoring tool called "VM Manager Plus", which can increase the efficiency of the system by providing backup and alert notifications and help in easy recovery after failure. CPU, memory, and disk utilization, network usage, memory swap, datastore read/write latency, and a memory ready are some of the various metrics monitored by the tool [12]. If the monitored results show any discrepancy between the required and current values of the metrics, then remedial suggestions can be made and recovery of VMs can be carried out.

The Trusted Virtual Private Datacenter model was proposed for securing and managing cloud resources and services [4]. The objective was to propose a security mechanism considering the trusted relationship between the client and the IaaS provider. The model uses this relationship so as to allow both parties to set security controls to protect data and infrastructure within the cloud and virtualized data center [4].

The VES model takes into consideration the security aspects of an entire environment and suggests measures for each dimension [5]. The entire environment including the virtual machines and the virtual data centers should be secure enough to store and process data [5].

VII. Conclusion

This paper proposes a data management system called DMaaS to control and manage heterogeneous cloud environments. It consolidates different data centers into one single service platform to efficiently utilize system resources in the data centers. The system consists of different service layers and the main subsystems include a presentation system, business process service system, and data analysis system. Our proposed system consists of core components of resource discovery and management, role-based access control, user pattern recognition, and a recommendation system. The proposed system enables us to utilize existing data centers efficiently and effectively without extra management tools.

The proposed system is a promising service software, solving many issues related to data center consolidation. The multi-tenant nature of this software will reduce per user cost since the fixed cost for setting up the service is shared across multiple users. Both small and large businesses can reduce their operational costs and focus on the main goal of their business rather than spending time and human resources to manage their complicated data centers.

VIII. References

[1] Yu-Sung Wu, Pei-Keng Sun, Chun-Chi Huang, Sung-Jer Lu, Syu-Fang Lai and Yi-Yung Chen, "EagleEye: Towards Mandatory Security Monitoring in Virtualized Datacenter Environment," 43rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), June 2013.

[2] Mohamed Esam Elsaid and Christoph Meinel, "Live Migration Impact on Virtual Datacenter Performance," IEEE International Conference on Future Internet of Things and Cloud, Barcelona, 2014.

[3] N.Staalinprasannah and S.Suriya II, "Implementation of XenServer to ensuring business continuity through the power of virtualization for cloud computing," IEEE International Conference on Computing, Communications and Networking Technologies (ICCCNT), Tiruchengode, 2013.

[4] Xin Wan, ZhiTing Xiao and Yi Ren, "Trusted Virtual Private Datacenter: A Model Toward Secure IaaS Cloud," Fourth International Conference on Multimedia Information Networking and Security, Nanjing, 2012.

[5] Annette Tolnai and Sebastiaan von Solms, "A Virtualized Environment Security (VES) Model for a Secure Virtualized Environment," International Conference on Internet Technology and Secured Transactions (ICITST), , London, 2010.

[6] Fu Wen and Li xiang, "The Study on Data Security in Cloud Computing based on Virtualization", International Symposium on IT in Medicine and Education (ITME), Cuangzhou, 2011.

[7] Andre van Cleeff, Wolter Pieters, and Roel Wieringa, "Security Implications of Virtualization: A Literature Study," IEEE International Conference on Computational Science and Engineering, 2009.

[8] Risto Vaarandi, Mauno Pihelgas, "Using Security Logs for Collecting and Reporting Technical Security Metrics," IEEE Military Communications Conference, 2014.

[9] vSphere from VM Ware.
http://www.vmware.com/products/vsphere

[10] XenServer from Citrix http://xenserver.org/open-source-virtualization-download.html

[11] Veeam ONE Availability Suite.
http://www.veeam.com/virtualization-management-one solution.html

[12] VM Manager Plus.
https://www.manageengine.com/virtualization-management/

[13] Netflix System Architectures for Personalization and Recommendation.
http://techblog.netflix.com/2013_03_01_archive.html

[14] Ziyu Wang, Jiahai Yang, and Fuliang Li. "An on-line anomaly detection method based on a new stationary metric - entropy-ratio," IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Sept 2014.