Benefits and Challenges of Managing Heterogeneous Data Centers

Jinho Hwang
The George Washington University
jinho10@gwu.edu

Sai Zeng and Frederick y Wu
IBM T. J. Watson Research Center
{saizeng, fywu}@us.ibm.com

Timothy Wood
The George Washington University
timwood@gwu.edu

***Abstract*—Today's businesses have a wealth of options available to them for running their applications. They can use proprietary infrastructures, private data centers, co-location facilities, or public clouds. Applications can be deployed on native hardware or virtualization can be used to improve consolidation and manageability. Public cloud platforms are growing in popularity, but businesses must decide both what types of services to use within a cloud and which cloud platform to use in the first place. While making use of these diverse resources poses management challenges, it also opens new opportunities for optimizing performance and ensuring reliability by carefully matching applications to the resources that will best support them. In this paper we present the benefits and drawbacks of data center diversity, and discuss our experience enhancing IBM's Tivoli Endpoint Manager system to better support mixed proprietary infrastructure, public, and private cloud environments.**

***Keywords*—Cloud Computing, Heterogeneous Infrastructure, Resource Management**

I. Introduction

Historically, enterprises have owned proprietary infrastructures to provide storage and computing nodes to their applications. In the last decade, this picture has changed significantly, and many enterprises now rely on a heterogeneous infrastructure by mixing their own proprietary infrastructures and resources provided by a service operator such as a public or a private cloud computing platform. Figure 1 shows heterogeneous infrastructure composed from different cloud resources, and private data centers. Public cloud infrastructures offer on-demand self-service, broad network access, resource pooling, and rapid elasticity [1]. Despite these benefits, a public cloud alone often does not exactly match the needs of many businesses, resulting in the development of a hybrid cloud model that combines private and public resources. This approach gives companies greater flexibility, but it leaves each business responsible for integrating these resources.

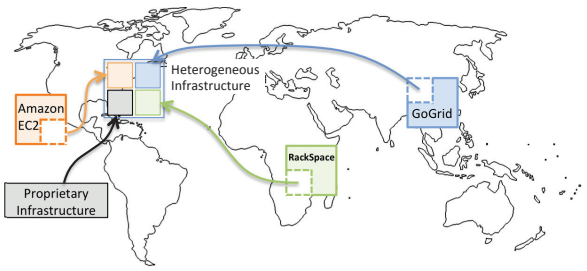


Fig. 1. Heterogeneous Infrastructure

A hybrid cloud combining public and private resources gives enterprises more options, but it is only the first step for a company that seeks to exploit the benefits of many different types of server and storage resources. Different clouds offer varying performance guarantees, reliability systems, virtualization platforms, and pricing models—a savvy consumer should be able to mix and match resources from their proprietary infrastructure, private and public data centers in order to obtain the resources that best match their applications' needs.

Unfortunately, connecting different types of resources poses many challenges related to security [2], deployment [3], and resource management [4]. Even with only private data centers completely under the enterprise's control, managing resources of diverse types running on different virtualization platforms is a major challenge. Both infrastructure management products [5–7] and research systems [8–10] attempt to mitigate some of these issues by providing a cohesive way to monitor and manage applications deployed across mixed public and private environments. However, these systems must continue to evolve as the types of resources enterprises use to run their applications change.

The absence of standards for management and configuration of heterogeneous infrastructures deprives companies, who can and want to utilize heterogeneous infrastructures in the business model, of benefiting from infrastructure heterogeneity. A non-standardized, but growing way to support multiple clouds is with a unified HTTP proxy [8, 11]. This creates a single API interface for multiple clouds, consolidating the management interface used by administrators. However, this causes problems: 1) the management interface has tight dependency with each cloud hosting company, so small changes to these interfaces can break the unified HTTP proxy; 2) each hosting company has different management characteristics in that the management interface can not be unified with the same units; 3) the approach usually follows a "lowest common denominator" style, preventing the use of advanced, but unique features provided by different services; 4) user account/maintenance are not easily managed.

Despite the challenges inherent in connecting different types of resources, we believe that the growing diversity of resources available can be a boon to enterprises that are able to effectively make use of them. A single cloud such as Amazon EC2 offers many different server instance types and multiple locations across the world, but if an enterprise considers the resources offered by *multiple cloud providers*, the diversity is even greater considering the pricing model, resource management, hypervisor type benefits, geographical

978-3-901882-50-0

advantages, recovery, and security.

In this paper we discuss the important management challenges around heterogeneous infrastructures. We also describe our experience enhancing IBM's Tivoli Endpoint Manager (TEM) system with a Heterogeneous Infrastructure Manager (HIM), to move closer to this vision. HIM's implementation must not be intrusive because TEM already has hundreds of thousands of machines registered in real businesses. In particular, we focus on the need for TEM, without any architecture modification, to understand the server type (virtualized/non-virtualized), the kind of hypervisors being used, and the infrastructure topology being used in a heterogeneous data center environment.

II. Why is Heterogeneous Cloud Infrastructure Promising?

HIM should enable end-to-end heterogeneous management capable of managing both public/private cloud infrastructures and proprietary infrastrucures. This feature opens up a new possibility for companies that do not own infrastructures or own a small infrastructure, but want to own or use a scalable cloud-based business to easily expand their business. This type of heterogenous resource mix can provide benefits from the perspective of pricing model, resource management model, geographical locations, recovery, and security. We will describe why HIM is promising in terms of these perspectives.

A. Best Pricing

Cloud hosting companies have different pricing models depending on company policies. One cloud hosting company may provide a better price for CPU usage, but it charges more on network I/O and memory, whereas the other cloud hosting company may provide better price on memory, but not on CPU usage. If an enterprise uses different infrastructures with different pricing policies, it can manage the resources depending on the price to achieve the best performance/price ratio.

Many websites (FindTheBest [12], Servdex [13], Cloudsurfing [14]) provide a service to compare the cloud infrastructure prices from different cloud hosting companies. Table I shows the base plans of several cloud infrastructure hosting companies. Each company has a base plan for the minimum price with the minimum computing unit. Uniquely, Amazon EC2 has a free micro instance that customers can use for a year without charge.

TABLE I. Infrastructure as a Service Pricing Comparison [12]

Items	Amazon EC2	GoGrid	RackSpace
Inbound Bandwidth	0¢ per GB	0¢ per GB	8¢ per GB
Outbound Bandwidth	12¢ per GB	29¢ per GB	22¢ per GB
Base Plan Cost	8¢ per hour	8¢ per hour	1.5¢ per hour
Base Plan	1.7GB RAM, 160GB local storage, 1 EC2 Comput Unit	0.5GB RAM, 10GB local storage	256 MB RAM, 10GB local storage, 10 Mbps Network Throughput
Additional IP Cost	$0	$0	$2 per month

As shown in Table I, each cloud infrastructure hosting company has its own strength on the resource price. Amazon provides a better price on in and out network bandwidth, memory, and local storage, whereas RackSpace provides a cheaper base plan with less capabilities. Also, Amazon and GoGrid do not charge for the additional IP. These different levels of granularity in price can satisfy different users' demands better.

B. Resource Management

In a HIM scenario, various options of resource management from different infrastructure companies can provide flexibility to the administrator. The goal of cloud resource management is to pay less and provide better user experience. In HIM, different infrastructures may have different performance under the same hardware setting. Also, this becomes clear when multiple tenants reside in the same physical machine, and some virtual machines (VMs) use the same resources simultaneously.

In order to see the performances of different cloud hosting companies, base plans of Amazon, GoGrid, and RackSpace, as shown in Table I, are tested with the freebench software [15]. As shown in Table II, freebench is an open-source multi-platform benchmarking suite providing different kinds of benchmarks such as game, auto encoding, compression, decompression, scientific, photo procesing, and encryption.

It happens that Amazon, GoGrid, and RackSpace all use the Xen hyperivosr (version unknown) as a cloud platform, so we do not have hypervisor bias on performance measurements. Linux kernel 2.6.32 (Ubuntu 10.04) is used as the guest operating system in each benchmark. Figure 2 illustrates the benchmark results, and the y-axis shows the absolute scores for each benchmark. The higher score, the better performance. For gnugo, ogg, comp, and decomp, RackSpace outperforms other companies' base models, despite having a lower hourly cost. Openssl has almost the same performance over all base models. The figure shows that if we can utilize different infrastructure properties, it may give better options to economically manage resources depending on the advantages of each infrastructure and the nature of the applications being run.

The key management challenges to do so are in determining which applications best fit into each cloud, as well as being able to dynamically relocate applications depending on their behavior and cloud price models (e.g., price fluctuations in Amazon's Spot Market). This will require advances in application profiling and resource management techniques such as live WAN migration.

TABLE II. Freebench Benchmarks

Items	Category	Notes
gnugo	Game	GNU Go is a free program that plays the game of Go
ogg	Audio Encoding	Encoding a song, Tibetan Chant
comp	Compression	Compress a file in bzip2
decomp	Decompression	Decompress a file in bzip2
scimark2	Scientific	Java benchmark for scientific and numerical computing
dcraw	Photo Rendering	Photo rendering tool
openssl	Encryption	OpenSSL AES encryption

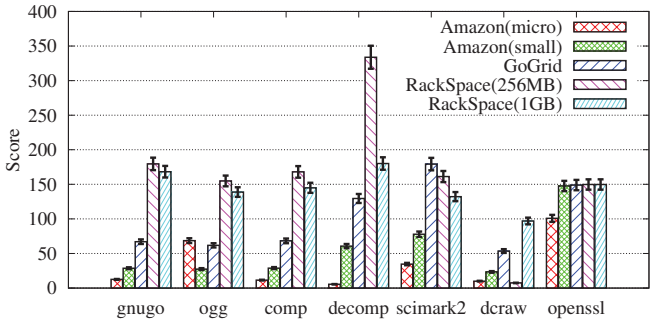


Fig. 2. Freebenchmark with Base Plans: Amazon offers a free tier (micro) with 613MB RAM, so both micro and base plan (small) are shown. The price of RackSpace is based on the memory size, so two options (256MB = 1.5¢ per hour and 1GB = 6¢ per hour) were used for RackSpace.

C. Hypervisor Benefits

Many hypervisors stem from different architecture considerations, and platforms. Among popular hypervisors, Xen and KVM are based on the linux kernel as their hosting operating system, whereas Hyper-V uses Windows 2008 Server, and VMware has developed its own operating system initially started from the linux kernel. The primary goal of these hypervisors is to support VMs without any performance impact due to the management operations. However, they may have different performance characteristics.

The Linux compile benchmark is one of the most common benchmarks that uses multiple resources including CPU, memory, and disk. Figure 3 shows the absolute times to compile an identical linux kernel under each hypervisor. Xen has the worst performance, and it takes about 40% longer than the time vSphere takes. However, even if the compile performance shows that Xen has the worst performance among these hypervisors, it does not necessarily mean that Xen has the worst performance on everything, instead we can interpret that the performance is sensitively affected by any component. In this case, Xen may have bad performance on disk since the device driver resides in the management domain (dom0), resulting in slow disk access. Our preliminary results suggest that just like the performance variation found among clouds, different hypervisors may offer different levels of performance depending on the types of resources being used by different applications.

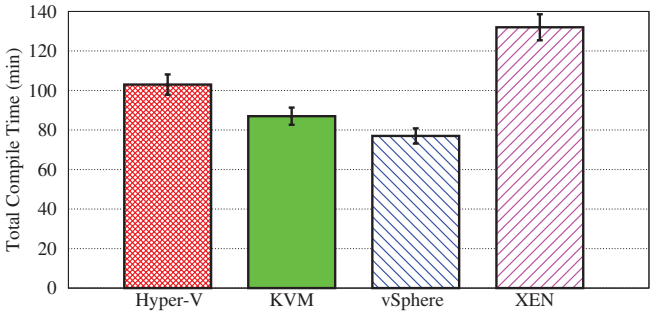


Fig. 3. Linux Compile Workload Benchmark

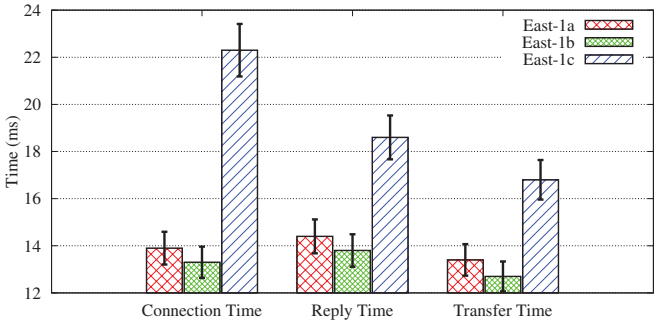


Fig. 4. Geographical HTTP Time Difference in Amazon EC2: Client is located in NY; Web servers are in East-1a, East-1b, and East-1c.

D. Geographical Advantage

Data centers in a cloud hosting company are located in geographically different locations. A data center location provides a network latency benefit if the data center is close to where a user resides. Further, making use of geographically distributed data centers can improve reliability by reducing the likelihood of correlated failures. However, even within a single cloud's data centers there may be performance differences based on load or infrastructure configuration.

To demonstrate this, we focus on Amazon EC2's East Coast region and consider its three different sub regions: east-1a, east-1b, east-1c. Figure 4 illustrates connection time, response time, and html file transfer time when instances host the same web service. The slowest data center, east-1c, has the worst communication time for all three measurements, with an overhead of nearly 50%. Clearly, if there is this much variation within a single cloud region, there will be even greater performance disparities between different cloud platforms with more diverse data center locations. As a result, we believe future managing systems must minimize client perceived delay while meeting cost and reliability goals.

E. Security

Security in the cloud is the most critical feature to many companies whether or not they have sensitive information. It has been the biggest barrier to those companies that consider to move IT infrastructure to public or private cloud [16].

If a trusted storage is a part of the cloud, it may be easy to manage the data security by restricting the boundary of data transfer. Since HIM provides the management system for any infrastructure, the proprietary infrastructure, instead of abandoning it, can be used for the sensitive data storage to be protected at all times.

User account management across clouds is tricky if a unifired HTTP proxy model is used because the management system must get permission from each cloud hosting infrastructure. However, for HIM, the user account management is easier than the unified HTTP proxy model because we do not need to get a permission from the cloud hosting companies, instead HIM can control the user account management without obtaining the permission from the infrastructure company by providing its own management system.

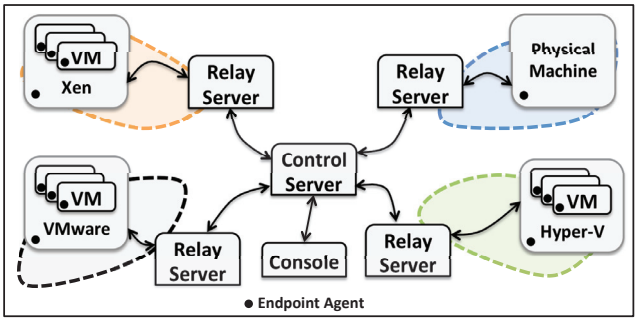


Fig. 5. HIM Architecture: Each endpoint machine has an endpoint agent that communicates with a relay server. A central server is connected to many relay servers; The relay server can scale; User/Admin console can connect to server to manage the infrastructure.

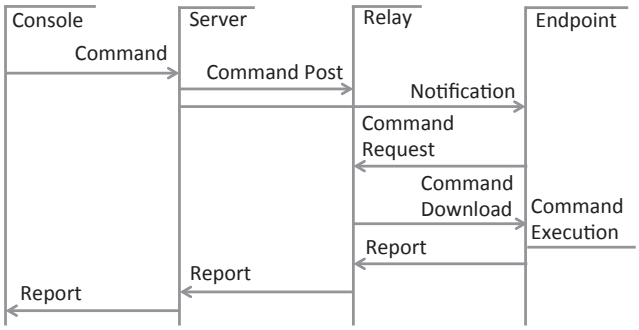


Fig. 6. HIM Operation Flow

III. HIM Operations

HIM follows a server-client architecture as shown in Figure 5. As the number of endpoints easily become thousands, the system is distributed by putting a relay server holding the control of a certain number of endpoints. Each endpoint only communicates with its parent relay server. That is, the communication between a server and relay servers is asynchronous in order to provide a seamless service reponsiveness.

Since HIM uses the native script approach including BASH script in Linux architecture and VBScript in Windows architecture, it can support both agent and agentless models. When HIM uses an agent model, every endpoint should have an agent running as a root (administrator) right, in every targeted entity, and being ready to receive commands from the server. The agent should be kept very small in order not to affect the performance of other running applications. As an example of an efficient practice, the average CPU usage of the agent is 2% over the course of executing the commands. This agent-approach provides a better control of each entity regardless of whether it is virtualized or not, and removes the dependency between HIM and each cloud hosting company's management system.

On the other hand, agentless approach requires a network connection for instant access to the machine when an administrator or an user wants to send commands to endpoints. Due to frequent connection establishment and disconnection, the endpoint performance degrades more than when using agent approach.

Figure 6 illustrates the work flow of HIM. A console of an administrator or an user connects to the HIM server to access management information, and to command endpoints. A server does not directly communicate with endpoints, instead posts a command in a relay server, and notifies endpoints with a simple ping. When a notification ping arrives in an endpoint, the endpoint talks to the relay server asking whether it has a new command to execute. If yes, the endpoint retrieves the command, executes, and reports it to the relay server. The relay server delivers the report to the server, and to the console.

IV. Topology Discovery

In the perspective of heterogeneous infrastructure environments, the topology discovery finding whether an endpoint is virtualized, whether an endpoint resides in proprietary infrastructure, public, or private clouds, which hypervisor an endpoint is running on, and the relationship of endpoints is an important element because it helps system management, especially administrative decision, resource management, patch management, and license management. For example, in virtualization environments, it is a common mistake to turn off the hosting computer, resulting that VMs running on it are turned off as well. For public clouds, we do not have a full control over hypervisor level, so endpoints of public clouds are managed in horizontal hierarchy without the information of endpoint relationship.

We use the network-based approach to discover the whole structure of the infrastructures. Each physical machine has a client agent running inside and reports the status of the machine periodically. The reason we choose the network-based approach is because the virtualization can hide everything about the physical machine including the hardware features. However, the network properties should be shared with VM in order to provide a proper connection management. Within the same gateway, all the MAC addresses are unique. So, the VMs and the hosting machines are categorized under the same gateway.

When a VM starts, a hypervisor in the hosting machine assigns a MAC address to the VM to initiate the network interface. So, each hosting machine (hypervisor) can provide MAC address information to HIM server. Also, each VM can send its MAC address to HIM server. From the server side, it is easy to compare these information to figure out how the topology looks like.

In HIM architecture, the server sends a script to run in the client, and the client leaves information in a file. Then, the server retrieves the information whenever it needs to use it. In general, the shell scripts in Linux and in Windows are different. Bash in Linux and VBScript in windows are used. Each hypervisor has a different way of obtaining the MAC information of VMs within the hosting machine. We target the four most popular hypervisors in the market, but it is easy to further extend to other hypervisors.

Xen: Xen has a built-in command, *xm list -l*, to list all the VMs including MAC addresses of them. However, the information also contains a lot of other information as well. We use the following pipelined shell command to retrieve only MAC address of the VMs.

```
xm list -l | egrep -o "([0-9a-fA-F]{2}:){5}[0-9a-fA-F]{2}"
```

KVM: Since KVM does not have a built-in command to show all the VMs, virsh that uses libvirt is used in our system. However, it is really tricky to list all the MAC addresses in the system. Two steps are needed to see the list as follows:

```
list=$(virsh list | awk '{gsub(/ +/," ")}1' | awk '{print $2}')
for elem in $list
do
   virsh dumpxml $elem | egrep -o "([0-9a-fA-F]{2}:){5}[0-
9a-fA-F]{2}"
done
```

VMware: VMware uses its own operating system in the hypervisor, called vmkernel. VMware at large has two different branches, ESX and ESXi, depending on the kernel. ESX vmkernel is similar to linux, which allows a third party agent to run insde of it, whereas ESXi vmkernel has only kernel core and kernel modules, which means a third party agent can not run on it. Instead, ESXi kernel offers another way of accessing the hypervisor information, that is, via open APIs. A third party can call open APIs or send a Perl script to run in the ESXi vmkernel to get information.

For ESX 3.5 and 4.1, we use a command "vmware-cmd -l" to extract a configuration file list of VMs, and then parse each configuration file to obtain a MAC address by using the same egrep regular expression as Xen and KVM. ESXi 5 has an agentless approach for VMware system management and remote command-line interface such as vSphere vCLI and PowerCLI, providing commands and scripting capabilities in a more controlled manner. vSphere vCLI is a combination of traditional VMware management tools such as esxcli and vmware-cmd. So, once a remote connection is established between ESXi 5 and a management console, the same method can be used as previous versions [17].

Hyper-V: Hyper-V is the only hypervisor running on the Windows system. Bash-script-based approaches do not work here, instead VBScript can be used to extract information from the Hyper-V hypervisor. In order to access the system or hypervisor information, we have to use windows management instrumentation (WMI) with an administrator access right.

For accessing the network information of Windows system, we need to get a WMI object handle with "winmgmts:\localhost\root\CIMV2". Then, we can query with "SELECT * FROM Win32_NetworkAdapterConfiguration" to get gateway address, IP address, subnetmask, and MAC address: DefaultIPGateway, IPAddress, IPSubnet, and MACAddress fields, respectively.

For accessing the hypervisor namespace, the WMI object handle with "winmgmts:\localhost\root\virtualization" should be obtained first. With the handle, we should query with "SELECT * FROM Msvm_SyntheticEthernetPort", or "SELECT * FROM Msvm_EmulatedEthernetPort". The query results include PermanentAddress field, which includes MAC address information of VMs running on the machine.

V. Implementation

A prototype system of HIM is developed on top of the IBM Tivoli Endpoint Manager (TEM) framework. Four hypervisors, Xen, KVM, VMware, and Hyper-V, are deployed in IBM blade servers with VMs as a proprietary infrastructure, there are many non-virtualized (physical) servers that have endpoint agents running as a proprietary infrastructure, and also VMs in Amazon, RackSpace, and GoGrid are also registered to the HIM system as public infrastructures.

Since HIM uses an agent-based system, every endpoint in question should run an agent, and be registered to HIM system. As shown in Figure 5, an agent communicates with a relay server in order to retrieve data (or commands) from the server or send reports to the server. The communication between endpoints and a relay server is asynchronous so that HIM can support highly distributed architecture for many endpoints.

Figure 7(a) shows an administrative dashboard display. On the left pane, it categorizes the sites depending on who takes charge of certain machines. Sending commands and patches can also be managed from the left pane control. On the right pane, it shows statistical data about the managed heterogeneous infrastructures, and detail information about the machine including status, and monitoring results. Figure 7(b) shows an user dashborad display that shows list of machines registered, and detail information and performance monitoring of one machine clicked. The topology can show the different categories such as machine type, hypervisor type, operating system type, and network classification.

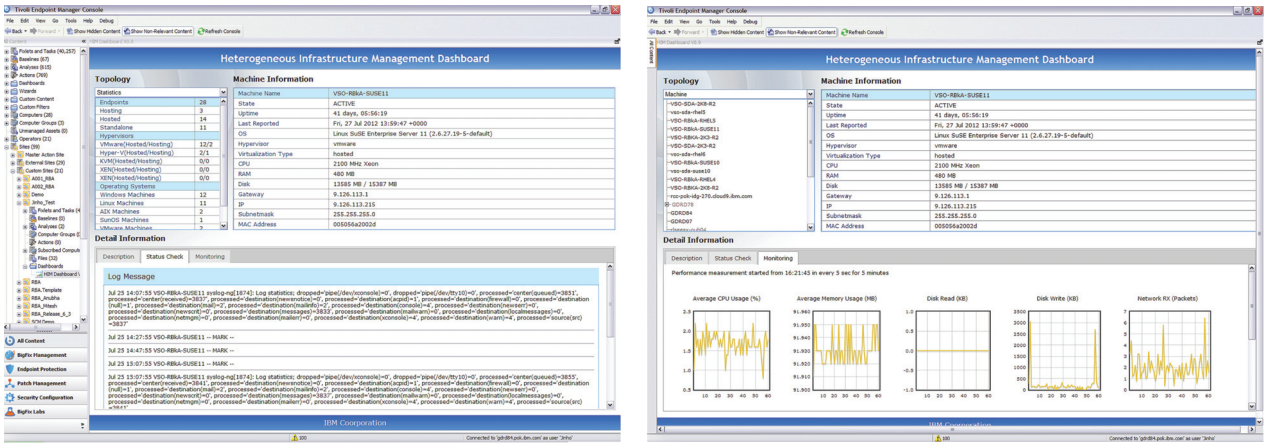
VI. Related Work

In large, there are two approaches to manage the heterogeneous cloud infrastructure: employing a new management layer and integrating existing management layers. The former requires a new management connection between the management of different hypervisor layers and applications, whereas the latter can utilize and combine the existing interfaces (or API) and is easier to implement. However, both have pros and cons. The former takes a long time to implement a new layer and to accept a new hypervisor, but can fully control the management layer, while the latter can deploy the management system in a short time, but it is highly dependent on the existing interfaces.

As the way of employing a new management layer, open source cloud management systems such as OpenStack [7], Eucalyptus [6], Nimbus [18], and OpenNebula [19] have broadened their popularity in the management of linux-based hypervisors. Among them, only OpenStack supports Hyper-V hypervisor while the others can not support Windows-based hypervisor. However, all of them only target the virtualized machines, so the proprietary infrastructure without virtualization can not be a part of the heterogeneous infrastructure. This does not meet our goal.

Also, simple network management protocol (SNMP) based monitoring of heterogeneous virtual infrastructure [9] deploys a new management layer, but it does not scale well because the SNMP protocol has to extend standard or proprietary protocols whenever there are needs for other functionalities.

As the way of integrating existing management interfaces, an Unified HTTP proxy [8, 11] aims to provide a consolidated



(a) Administrator Screenshot

(b) User Screenshot

Fig. 7. HIM Prototype Screenshots

web interface integrating web interfaces of public cloud infrastructures. As we pointed out in Section I, this approach has many problems to be applied in the hetegoreneous infrastructures.

VII. Conclusion

The IaaS model used by many cloud platforms is attractive, however, different clouds offer different features and costs, causing many enterprises to use a hybrid model. This results in diverse infrastructures with varying platforms, hardware, hypervisors, locations, pricing models, and features. While this heterogeneity increases management complexity, it also may provide benefits to savvy customers able to effectively exploit this diversity. In this paper, we have demonstrated the performance, cost, and feature variations found in different clouds and hypervisors. We describe our work building a Heterogenous Infrastructure Management system into the IBM TEM system framework. This expands TEM's control plane to understand what virtualization platforms and host topologies are being used within a set of heterogenous resources. We believe enhancing management systems with this kind of information is the first step towards building automated systems capable of more efficiently managing resources by exploiting cloud and resource diversity.

References

[1] Peter Mell and Timothy Grance, "The nist definition of cloud computing," *NIST*, 2011.

[2] Timothy Wood, Alexandre Gerber, K. K. Ramakrishnan, Prashant Shenoy, and Jacobus Van der Merwe, "The Case for Enterprise-Ready Virtual Private Clouds," in *Proceedings of the Conference on Hot Topics in Cloud Computing (HotCloud)*, June 2009.

[3] Mohammad Hajjat, Xin Sun, Yu-Wei Eric Sung, David Maltz, Sanjay Rao, Kunwadee Sripanidkulchai, and Mohit Tawarmalani, "Cloudward bound: planning for beneficial migration of enterprise applications to the cloud," in *Proceedings of the ACM SIGCOMM 2010 conference*, New York, NY, USA, 2010, SIGCOMM '10, pp. 243–254, ACM.

[4] Tian Guo, Upendra Sharma, Timothy Wood, Sambit Sahu, and Prashant Shenoy, "Seagull: Intelligent Cloud Bursting for Enterprise Applications," in *Proceedings of the Usenix Annual Technical Conference (short paper)*, June 2012.

[5] IBM, "http://www-01.ibm.com/software/tivoli/solutions," 2012.

[6] Eucalyptus, "http://www.eucalyptus.com," 2008.

[7] OpenStack, "http://www.openstack.org," 2010.

[8] Tiancheng Liu, Yasuharu Katsuno, Kewei Sun, Ying Li, Takayuki Kushida, Ying Chen, and Mayumi Itakura, "Multi cloud management for unified cloud services across cloud sites," *CCIS*, 2011.

[9] Ya-Shiang Peng and Yen-Cheng Chen, "Snmp-based monitoring of heterogeneous virtual infrastructure in clouds," *APNOMS*, 2011.

[10] Shixing Yan, Bu Sung Lee, Guopeng Zhao, Ding Ma, and Peer Mohamed, "Infrastructure management of hybrid cloud for enterprise users," *Systems and Virtualization Management (SVM), 2011 5th International DMTF Academic Alliance Workshop on*, 2011.

[11] Shi Xing Yan, Bu Sung Lee, Guopeng Zhao, Chunqing Chen, Ding Ma, and Peer Mohamed, "Monsoon: Policy-based hybrid cloud management for enterprises," *HP Technical Report*, 2012.

[12] FindTheBest, "http://www.findthebest.com," 2012.

[13] Servdex, "http://www.servdex.com," 2011.

[14] CloudSurfing, "http://www.cloudsurfing.com," 2012.

[15] Freebench, "http://code.google.com/p/freebench," 2008.

[16] Thomas Ristenpart, Eran Tromer, Hovav Shacham, and Stefan Savage, "Hey, you, get off of my cloud: Exploring informatioin leakage in third-party compute clouds," *CCS*, 2009.

[17] VMware, "Vmware esxi 5.0 operations guide," *VMware Technical White Paper*, 2011.

[18] Nimbus, "http://www.openstack.org," 2004.

[19] OpenNebula, "http://opennebula.org," 2002.