CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3090

SPECIAL ISSUE PAPER

On construction of heuristic QoS bandwidth management in clouds

Chao-Tung Yang[1], Jung-Chun Liu[1], Rajiv Ranjan[2,*,†], Wen-Chung Shih[3] and Chih-Hao Lin[1]

[1]*Department of Computer Science, Tunghai University, Taichung 40704, Taiwan*
[2]*Information Engineering Laboratory, CSIRO ICT Centre, Canberra Australia*
[3]*Department of Applied Informatics and Multimedia, Asia University, Taichung 41354, Taiwan*

ABSTRACT

In recent years, cloud computing has become popular and its applications widespread. Thus, there exists a common concern, that is, how to arrange and monitor various resources in the cloud computing environment. In the literature, Ganglia and Network Weather Service (NWS) were used to monitor and gather node status and network-related data, respectively. With supports of Ganglia and NWS, one can effectively administer available resources in the cloud computing environment. In order to achieve high performance of cloud computing, comprehensive monitoring and efficient management are critical. Ganglia is often used to gather status data of resources, such as live states of hosts, CPU or memory utilizations, and surely Ganglia is also capable of monitoring network-related information; however, instead of Ganglia, we used NWS services to gather network-related information such as end-to-end transmission control protocol/Internet protocol performance data. Compared with Ganglia, NWS services offer more selections and flexibility for measurement schemes. Besides, NWS services could be deployed with nonintruding manner that makes it easier and faster in deploying services to cloud nodes. The network-related information is acquired immediately after deployment. Although NWS services also provide measurements for CPU and memory utilizations, but less functionality is provided by them than Ganglia in these aspects. Therefore, we combine advantageous features of Ganglia and NWS to achieve the aims of effective monitoring and management of available resources in the cloud environment. Nevertheless, Ganglia and NWS services may not provide sufficient data in realistic situations due to diversified needs of users, especially application developers. For instance, users are not able to directly access utilizations or allocations of resources in the cloud environment via interfaces or channels of Ganglia or NWS. In addition, NWS services based on a domain-based network information model could greatly decrease overheads caused by unnecessary measurements. Hence, we propose a heuristic QoS measurement approach based on the domain-based information model. This measurement approach is capable of providing essential information to satisfy user requirements, and thus let users manage and monitor various resources in the cloud environment in a more efficient way. © 2013 Wiley Periodicals, Inc.

Received 9 June 2013; Accepted 10 June 2013

KEY WORDS: cloud computing; Ganglia; Network Weather Service; heuristic QoS measurement; domain-based network information model

1. INTRODUCTION

The cloud computing technology [1–4] is to a greater extent widely adopted by organizations to attain high performance computing and sharing of heterogeneous resources. Because all computing nodes in cloud environments are connected by means of networks, all tasks executed in cloud environments will be influenced by complex network status caused by complicated and various communications among

*Correspondence to: Rajiv Ranjan, Information Engineering Laboratory, CSIRO ICT Centre, Canberra, Australia.
†E-mail: rajiv.ranjan@csiro.au

computing resources. While designing algorithms for particular functions or assigning tasks in cloud environments, we need to assess the network performance by related information and adjust algorithms or parameters to achieve optimal performance in real-time runs. Ideally, the cloud environment is equipped with some mechanisms to automatically retrieve status of the network and evaluate its performance. As a result, applications or web service agents could provide services with higher performance based on dynamic tuning of parameters and optimization algorithms.

While cloud computing has gradually become widespread, a common issue occurs, that is, how to manage and gather information of numerous resources in the cloud computing environment. Usually, Ganglia and Network Weather Service (NWS) are used to gather states of machines and network-related information, respectively. Due to varied user requirements, data acquired by these services are not sufficient for some realistic situations. In our previous works, we designed mechanisms to retrieve network-related information in the real time, and even provided advanced customization features for special purposes. With the customized shell scripts written by us to deploy NWS services, we could easily implement NWS services to each node and gather network-related information in a prefixed period. In addition, we could obtain extra statistics for job-scheduling and other purposes in our computing environment.

In our previous work, we found that the service provided by NWS is influenced by network environment changes; thus, NWS services need to be frequently redeployed manually. The term 'manual' is equal to 'inefficiency' in management of networks. A typical example is shown in Figure 1, in which a NWS clique is registered for cloud nodes A1, A2, A3, and A4. A1 is the header and is in charge of relevant network measurements among these nodes. If A1 has hardware failure or forced to reboot after software updating operations, the NWS clique will be terminated and network administrators are forced to restart the clique manually again. Besides, by default, we will not be notified if any node fails. Thus, we plan to include a network management system (NMS) using the Simple Network Management Protocol technique to work with NWS services to resolve this issue in the future.

To achieve dynamic detection and recovery of NWS services, we need to design some methods to effectively administrate cloud resources. Therefore, we propose a heuristic Quality of Service (QoS) measurement based on the domain-based information model to provide essential information to satisfy requirements of users. Besides, we hope that, via the proposed system, users could manage and monitor numerous resources in the cloud environment in more efficient ways. As for application developers, we expect to provide a generic interface to assess network performance before accessing cloud resources.

We have simplified deployment of NWS services into a few simple steps, which makes it fast and easy to deploy NWS services in cloud machines for periodically retrieving network information without intruding existing systems. Furthermore, by using relational database management system [9], we are able to store historical network data and use them to do statistics in advance for QoS assessment. Statistics is helpful in many aspects, such as job dispatching or replicas selection. In this

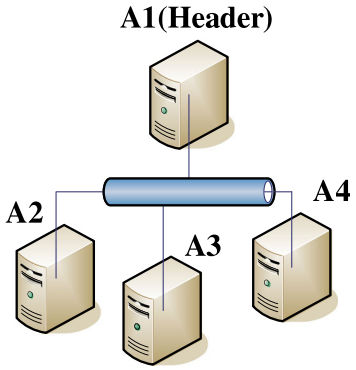

Figure 1. A typical Network Weather Service clique deployment of cloud nodes.

DOI: 10.1002/cpe

paper, the proposed QoS measurement is helpful for reducing complexity of network status prediction; besides, a generic interface is also provided for users to evaluate network performance before accessing cloud resources.

The rest of this paper is organized as follows: Section 2 gives a background review of machine and network information providers, QoS, and related works. Section 3 describes the proposed heuristic method, domain-based network information model, and design of QoS measurement for network bandwidth. Section 4 presents the experimental results and comparison of performance of different strategies. Finally, Section 5 summarizes contributions of the paper and discusses the future work.

2. REVIEW OF BACKGROUND AND RELATED WORKS

2.1. Cloud computing

Cloud computing [27–30, 42, 43] is a new type of IT service based on the Internet, and it can satisfy needs of users by sharing computing resources such as software, hardware, and information via networks. Cloud computing starts another colossal revolution in the industry after the mainframe and client-server model in the 1980s. To enjoy cloud services, users do not need to understand the details of its infrastructure, require no expertise, and have no direct control of the cloud. Cloud computing is further combined with dynamic scalable functions and virtualization resources and can be viewed at the following levels of services:

Infrastructure as a service [31–33, 44] provides the computing capacity for processing, storage, networking, and other basic computing resources, in which the user is able to implement and execute software, including operating systems and applications. The underlying cloud infrastructure is beyond the control of the user; however, the user has control over operating systems, storage, applications, and limited control of some networking components such as host firewalls.

Software as a service [34, 35] allows the user to access and use applications on a cloud infrastructure from various clients via a thin interface such as a browser. The user does not need to manage the underlying cloud infrastructure including networks, servers, operating systems, storage, and applications, except some limited user-specific application configuration settings provided by the service vender to let the clients to configure software as a service applications according to their needs

Platform as a service [35–37] provides a running software stack, usually with proprietary programming interfaces or application programming interfaces that allow the user to develop and run applications. The consumer-created or acquired applications deployed in the cloud infrastructure are created using programming languages and tools supported by the vender. Platform as a service may require the user to develop applications using specific application programming interfaces from service venders. The user does not handle the underlying cloud infrastructure but has control over the installed applications and application host's environment configurations.

The cloud can be classified as public, private, or hybrid types, too. Anyone who can access the Internet can use the public cloud service; whereas, only valid employees of enterprises, the government, or schools can use their own private cloud services. The hybrid cloud, such as Amazon S3 and EC2, contains both the private and public cloud, which, first built as a private cloud, provides services to the public after everything on it is stable.

2.2. Machine information provider

Ganglia [8], an open source project grown out of the University of California, Berkeley's Millennium initiative, is a scalable distributed system for monitoring status of nodes (processor collections) in wide-area systems. It uses a hierarchical, tree-like communication structure among its components in order to accommodate data from any large collection of multiple clusters. The data gathered by the Ganglia monitor comprises hardware and system data, such as CPU loads, processor types, memory and disk usages, operating system information, and other static/dynamic scheduler-specific details. It also provides a web interface for users to oversee machines. As shown in Figures 2 and 3, the Ganglia web portal is used to oversee resources of clusters or the cloud in the laboratory.

DOI: 10.1002/cpe

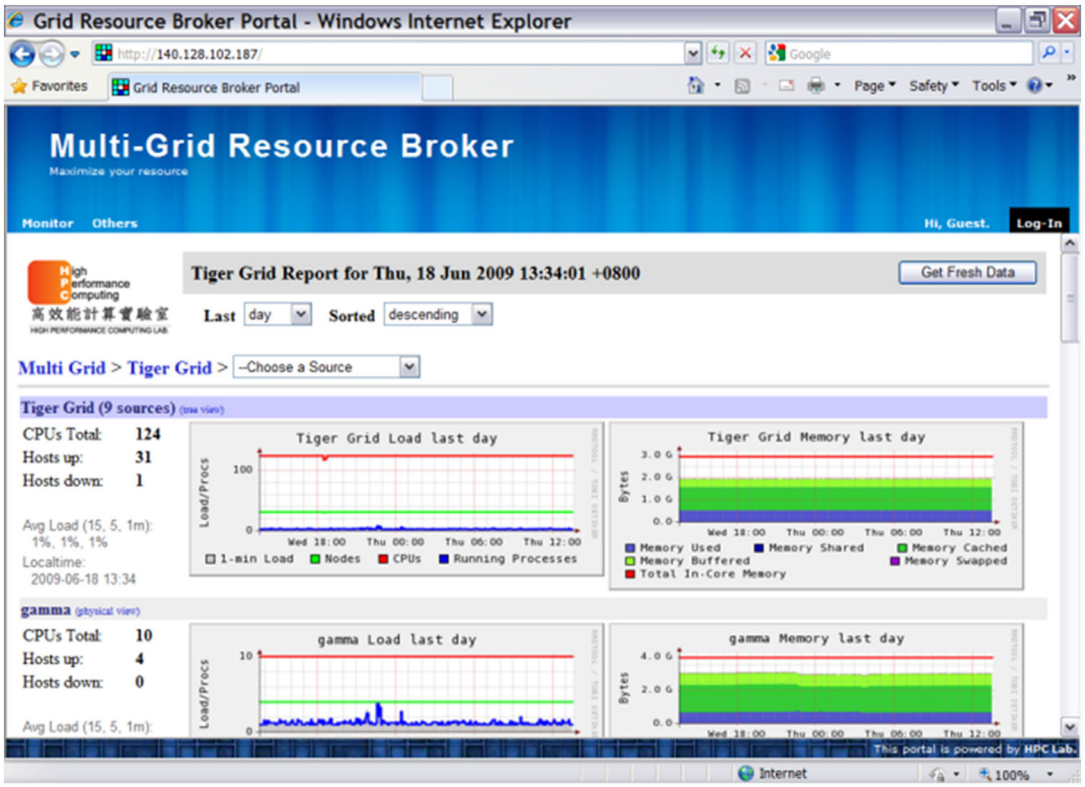

Figure 2. Resource Broker using Ganglia web portal.

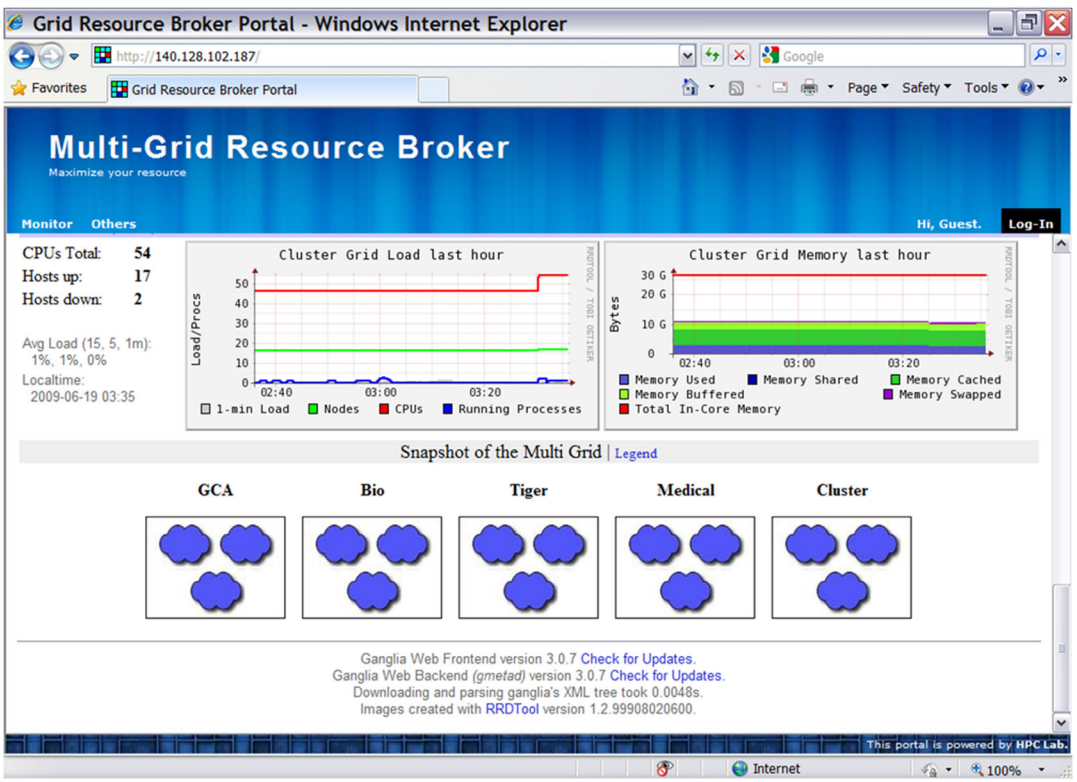

Figure 3. The single Ganglia web portal.

DOI: 10.1002/cpe

2.3. *Network information provider*

Network Weather Service [13] [17] is a distributed system that detects network status by periodically monitoring and dynamically forecasting over a given time range. The service operates on a distributed set of performance sensors (network monitors, CPU monitors, etc.), from which it gathers system status data and applies them into numerical models to forecast system status. The NWS system includes sensors to measure end-to-end transmission control protocol/Internet protocol (TCP/IP) performance (bandwidth and latency), available CPU percentage, and available non-paged memory. The sensor interface, however, allows new internal sensors to be configured into the system. We mainly used NWS for end-to-end TCP/IP measurements.

Rich Wolski *et al.* [17] mentioned that NWS is designed to maximize four possibly conflicting functional characteristics. Despite the highly dynamic execution environment and evolving software infrastructure provided by shared meta-computing systems, it must meet four main goals: predictive accuracy, nonintrusiveness, execution longevity, and ubiquity.

For automatic NWS deployment, we developed a number of shell scripts, which form the basis for management of NWS services. In addition, we combined NWS services with the Ganglia web portal to gather data and display statistics of network status, as shown in Figures 4 and 5.

2.4. *Quality of service*

QoS [25] [26] is the ability to provide various priority to different applications, users, and data flows, or to make sure of a certain level of performance to a data flow. It is widely used in the field of computer networking. We used QoS as a quality measurement in the cloud environment.

QoS sometimes refers to the guaranteed level of QoS. High level of QoS is an expectable crucial factor of highly reliable and high performance cloud environments. Characteristics, such as availability, accessibility, or maintainability will affect user experiences of services. In order to meet user requirements in various situations with sufficient QoS, network performance needs to be

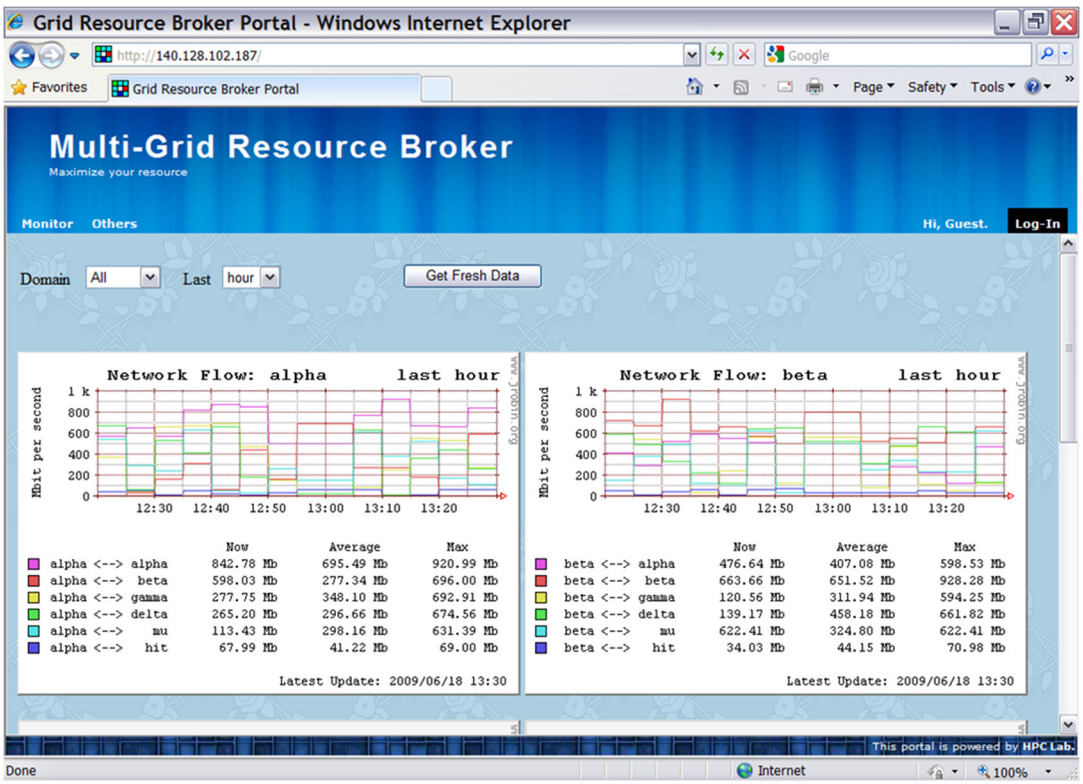

Figure 4. Network Weather Service combined with the Ganglia web portal.

DOI: 10.1002/cpe

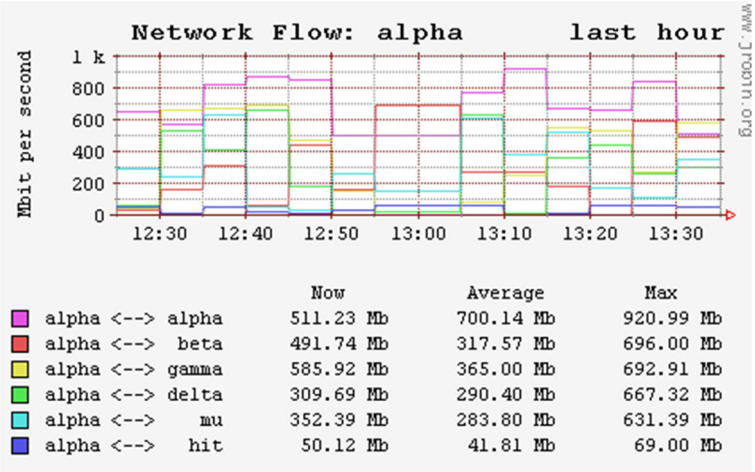

Figure 5. Network statistics by Network Weather Service measurements and displaying in web portal.

evaluated in advance. Some researchers have proposed network performance evaluation models [26], [15] to help network administrators to effectively examine network performance and accordingly adjust network resources. In this paper, we proposed a heuristic QoS measurement method that can be used to assess network performance in advance.

2.5. *Network management system*

In our previous work, we built a web portal composed of Ganglia and NWS services to monitor cloud environments. However, we found that it is inefficient to manage resources in an inactive manner. Thus, we decided to take advantage of NMS to manage and monitor resources in cloud environments actively. The primary communication mechanism among NMS, network devices, and cloud nodes is based on Simple Network Management Protocol.

We adopted the network management tool, NINO [14] as our experimental NMS, as shown in Figure 6. We believe that we could merge NMS like NINO, with our previous design to actively manage resources in cloud environments. Meanwhile, we are working on integration of Ganglia, NWS, and NINO. To sum up, we used NMS like NINO to monitor NWS services and operating status of cloud nodes. If any failure occurs, we are expected to be notified by NMS like NINO. And then, we could take appropriate actions to recover services semi-automatically or full-automatically. In addition, several service functions written with PERL [12] are also under development.

2.6. *Relation to our previous work*

In the paper [38], our focus is on how to manage and monitor numerous resources of grid computing environments [5, 6, 10]. Mostly, we use Ganglia and NWS to monitor machines' status and network-related information, respectively. But, information provided by Ganglia and NWS is not sufficient in some scenarios owing to varied user requirements. Therefore, we propose a heuristic QoS measurement constructed with domain-based information model that provides more effective information to meet user requirements. Furthermore, we expect that users could manage and monitor numerous resources of grid environments more effectively and efficiently. The paper [40] is shorter version of the [38] and it is only eight pages in length. Most the same paragraphs are related to background review parts. There are no detail experimental environment and results in both papers.

The paper [39] focuses on Comprehensive monitoring and effective management as criterions for archiving higher performance in grid computing environments. Unfortunately, owing to diverse user requirements, information provided by Ganglia and NWS services is not sufficient in real cases, especially for application developers. In addition, NWS services that deployed on the basis of 'domain-based network information model' could greatly reduce overheads caused by unnecessary measurements. On the other hand, in this paper, we propose a heuristic QoS measurement

DOI: 10.1002/cpe

Figure 6. A snapshot of NINO, which is able to scan device types of fabric layer, such as hosts, switches, and routers.

architecture, which constructed with a domain-based model, to provide effective information to meet user requirements, especially for application developers. This paper states more descriptions on this 'domain-based network information model.'

The paper [41] is totally different from this SI paper 'On Construction of Heuristic QoS Bandwidth Management in Clouds'. The focus of [41] on cloud services has been regarded as the significant trend of technical industries and applications after Web Services. The framework of cloud services contains the infrastructure, OS, virtual machines, platform, cloud web application services, and cloud devices. Due to the improvement of global network bandwidth over the past few years, people began to mutually share video files with larger space; in addition to the popularity of cloud computing and storage, video websites become fairly popular. The goal of this paper is to build video services on cloud infrastructure as a service environment, which integrates Kernel-based Virtual Machine and OpenNebula open sources to provide a cloud virtual environment for end users. This paper realizes video services that are easy for users to understand, access, and operate with it on cloud. Most overlapping texts are related to background review parts.

3. HEURISTIC QOS MEASUREMENT

In our previous project, we built an integrated environment including a web portal composed of Ganglia and NWS services. Afterward, we started another project called Picture Archive and Communication System [18], and many experiments were performed on the same platform. The primary mission of Picture Archive and Communication System is to efficiently exchange medical images in a specific application developed by our team. The application, called 'Cyber [24]', has successfully integrated eight algorithms. To speed up exchange of medical images in algorithms contained in Cyber, we had to configure many parameters before submitting tasks. Nevertheless, we had no idea of the most suitable combination of parameters in advance. Therefore, we used the trial and error method to find it. Apparently, it is definitely not very practical for most situations [16].

DOI: 10.1002/cpe

For this reason, we established an automatic mechanism for self-optimization of parameters. To guarantee QoS, we included user requirements as constraints of tasks. Armed with these constraints and the proposed heuristic QoS measurements, we were able to provide better QoS to meet user requirements.

3.1. Domain-based network information model

On the basis of the proposed approach, we used a domain-based network information model [19], [21], [23] to deploy NWS services. The domain-based network information model is designed to tackle point-to-point bandwidth measurement problems. After investigating it by tests in physical environments, we found that the domain-based network information model is very useful for reducing network measurement costs. Figures 7 and 8 show the schematic and physical design of the domain-based network information model, respectively.

For instance, assume a cloud composed of a few nodes. If every T seconds (e.g., $T = 1 \sim 3$ s), each node measures links between itself and all the other nodes, thus a total of NMN (n) network measurements are made, where

$$NMN(n) = n\tilde{n}(n-1) \tag{1}$$

Apparently, the total number of network measurements will grow fast in large-scale cloud environments. For example, 20 hosts in a test-bed would generate NMN (20) = 380 measurements. Hence, network traffic will become very heavy, especially if the background cloud intra-traffic is initially busy.

Figure 8 illustrates the domain-based network information model used in our previous work [22], which consists of four sites, each with four nodes. Each of the four sites has a head node (or called as a header), that is, A1, B1, C1, and D1 as shown. Each head node regularly measures links between itself and the other three head nodes, and links between itself and all the other nodes in its site. Consequently, by the domain-based network information model, the number of measurements is reduced to

$$NMS(n, [n_i]) = NMN(n) + \sum NMN(n_i) \tag{2}$$

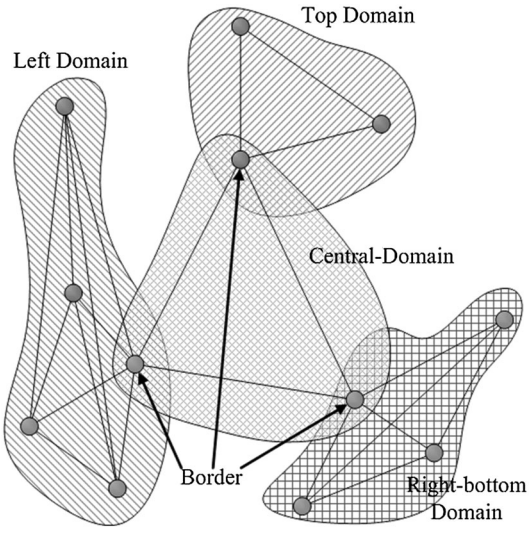

Figure 7. Schematic of the domain-based network measurement model.

DOI: 10.1002/cpe

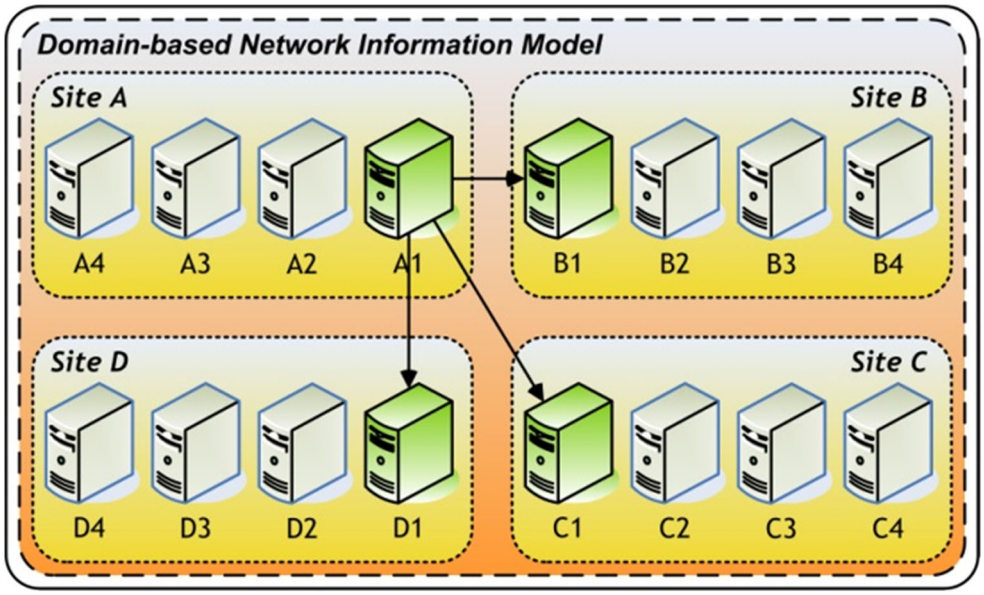

Figure 8. The design of domain-based network information model.

Where n_i is the total number of nodes in site i. In our test-bed, the number of network measurements might be decreased to $NMS(4, [9,4,3]) = 114$. The reduction rate R is defined as

$$R = \frac{NMN(n) - NMS(n, [n_i])}{NMN(n)} \tag{3}$$

In the previous example, the reduction rate R is found to be about 70%, which shows an impressive improvement by using the domain-based model.

Although this model saves many measurements and bandwidths, it lacks network information between nodes belonging to different sites (except nodes on borders). This model reduces a large number of connections; hence, it lacks network information of nodes except head nodes in two different sites. This model could carry out an estimation to provide network information of nodes in two different sites, but one of these two nodes should be a head node, for example, the link between node A2 and B1. No network information is measured between two nodes located in two different sites, if both not a head node. For example, no bandwidth measurement is performed between nodes A2 and B3.

We further improved the static model by adopting an exchanging scheme to build a dynamic domain-based network information model. Figure 9 illustrates how it works. The principle is to exchange the busy head node to a free node in the same site. For instance, if node A1 is busy, the next free node, node A2, may become the head node of site A and measure the bandwidth between itself and say, nodes B2, C3, and D4, the free nodes in sites B, C, and D, respectively. The purpose is to prevent using a busy head node acting as a border node, which might reduce system performance. The proposed dynamic model has three essential advantages:

1. The number of bandwidth measurements is the same as a static model, so the time complexity is not worsened.
2. Bandwidth measurements between pairs of arbitrary nodes of different sites are easily performed.
3. True values instead of estimated ones are obtained in network bandwidth measurements; thus, the Resource Broker is able to efficiently schedule tasks assigned to multiple sites.

DOI: 10.1002/cpe

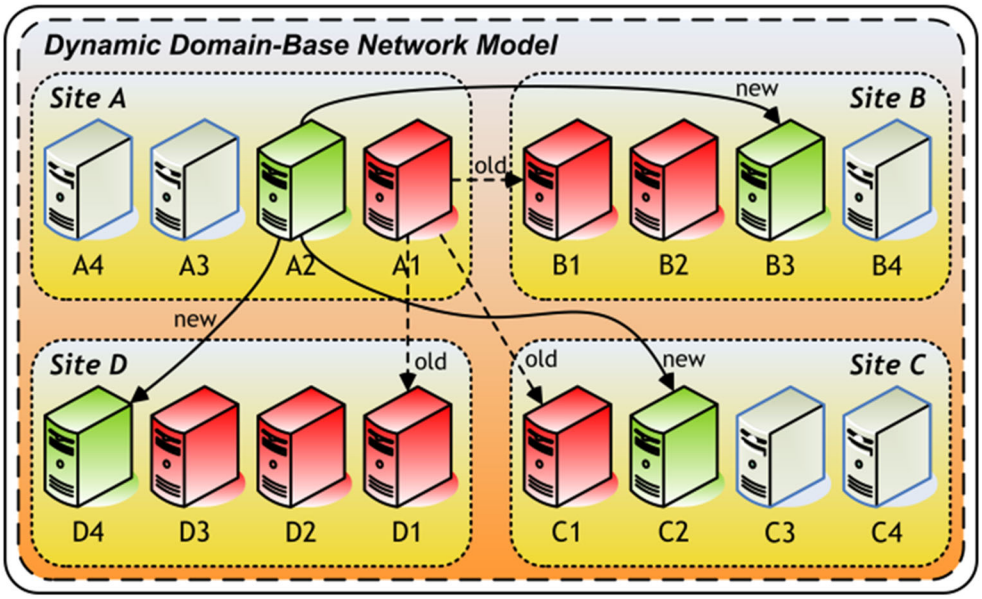

Figure 9. Design of a dynamic domain-based network information model.

The previous advantages are obtained if head nodes of all nodes could be dynamically replaced. Hence, it is very important that the head nodes can be dynamically adapted without the need of manual operations. The heuristic QoS measurement is a feasible solution in this aspect.

3.2. Deployment and flowchart of Network Weather Service

When deploying NWS services, we focused on eliminating interferences of existed services on each cloud nodes. Each computing resources consists of one nameserver and multiple sensors. In addition, arbitrary 'persistence state' may be set up in different locations. Without loss of generality, we only assigned one nameserver, one memory server, and one clique for each group of cloud nodes. We took a few cloud nodes as a group, each with a header to deploy nameserver and memoryserver. We adopted a simple NWS services deployment procedure with three steps:

Step 1 Clean all NWS processes.
Step 2 Load NWS services.
Step 3 Register NWS clique.

Figure 10 illustrates the standard procedure written in shell scripts. Due to the nonintrusiveness property of NWS, these shell scripts can be executed without root privilege.

Figure 11 depicts the flowchart to gather network information. In this project, we wrote crontab scripts to schedule routines for automatically loading NWS information into database and backing up raw data as plain text files in the local.

Whereas the scheduled routines in crontab are triggered, customized shell scripts are executed. First, host groups are taken from database to gather NWS information. Each host groups is predefined in

Figure 10. Steps for Network Weather Services deployment.

DOI: 10.1002/cpe

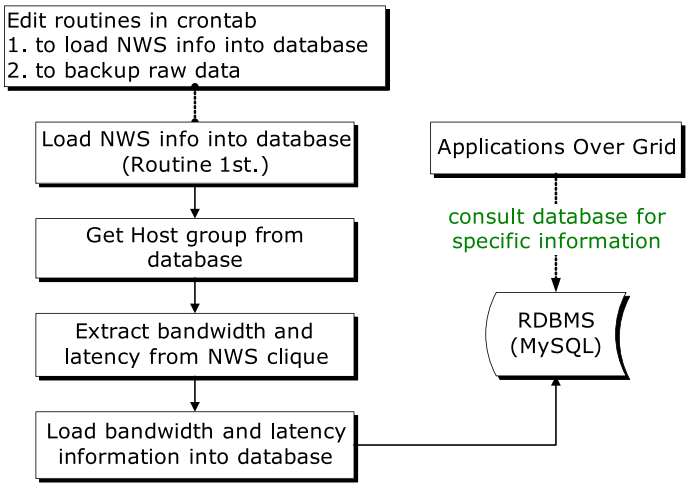

Figure 11. Flowchart to gather network information.

database and will be allotted a clique to measure network status. After creation of the clique, it will sample network information in a regular time, for example, every 10 s. The clique bandwidth and latency can be extracted by the script, and then loaded into database.

The routine defined to keep raw data as plain text files in the local is designed for future use. At present, it just provides an alternative storage other than the database to store raw data.

The method described earlier helps us successfully save network information into database in a regular time interval. But for migrating to dynamic NWS services detection and recovery, we have to simplify operations of deployment and practical usages as well. After a few tests, we have successfully simplified NWS services operations to three shell scripts, as listed in Table I:

Table I. Simplified Network Weather Services operations, which is helpful for dynamic NWS services detection and recovery.

To deploy NWS services	**#install_nws.sh for NWS deployment** cd ~ mkdir src cd src wget http://140.128.98.39/src/nws.tar.gz wget http://140.128.98.39/src/run.sh cp run.sh ~ chmod 755 ~/run.sh tar xvf nws.tar.gz -C /usr/local/bin **#Example** ./install_nws.sh
To start NWS services	**#run.sh for starting NWS** services/usr/local/bin/nws/bin/nws_nameserver -e /usr/local/bin/nws/log/nameserver.err -f /usr/local/bin/nws/log/registrations -l /usr/local/bin/nws/log/nameserver.log > /dev/null & /usr/local/bin/nws/bin/nws_memory -d /usr/local/bin/nws/log -e /usr/local/bin/nws/log/memory.err -l /usr/local/bin/nws/log/memory.log -N $1 > /dev/null & /usr/local/bin/nws/bin/nws_sensor -M $1 -N $1 > /dev/null & # **Example ('zeta1' is Header)**./run.sh zeta1
To start activity (clique)	**#act.sh for starting NWS activities (clique)** [root@zeta1 home_nws]# cat act.sh #$1 = nameserver/memoryserver /usr/local/bin/nws/bin/start_activity -F -f myClique $1 **#Sample** ./act.sh beta2

DOI: 10.1002/cpe

To have proper parameters for NWS, clique is the key to measurement accuracy of NWS. Table II lists the sample set of parameters used for measurement.

3.3. Heuristic approach

Statistics is helpful in many fields, especially for estimation and prediction. Some researchers have used statistical method to monitor and predict bandwidth for QoS sensitive tasks [25]. In this paper, we collected historical network information of cloud environments and found an approach to evaluate QoS. We could provide applications dedicated parameters in a simple manner by means of database operations. Several functions have been designed for analyzing historical information of network performance. All network-related data were periodically categorized to most common statistics; therefore, applications could dynamically adjust parameters of networks for better performance, without the need to send requests to estimate network status between all cloud nodes in a real-time manner.

Besides, we have organized an innovative method to obtain real-time network states that could work in the dynamic domain-based network information model, which dynamically deploys the dedicated node in the clique, measures network states, and then reports results to database, users, or applications. The enhanced version of work to support the dynamic domain-based network information model is currently under development.

We have designed a simple model for integration of Ganglia, NWS, and NINO, as shown in Figure 12. Ganglia and NINO provide UI for users to manage and monitor cloud environments. NWS and Ganglia collect related information from hosts and network regularly; whereas 'Smart Broker' provides parameters to applications such as Cyber.

Table II. A sample set of parameters of Network Weather Service (NWS) clique.

Clique:	name:cross-domain
A *control* coordinates experiments between a set of NWS sensors,	controlName:clique
called the members of the clique	skillName:tcpMessageMonitor
Guideline:	period:60
To achieving higher accuracy of measurement,	member:zeta1
the attributes 'size', 'message', and 'buffer'	member:beta2
should adjust for different cloud environments.	member:delta2
We used 'nws_ping' as a tool to obtain proper	member:eta4
configuration before deploying NWS clique.	**size**:**24000**
This technique requires experiences and concept about	**message**:**256**
NWS core services. You had better read NWS	**buffer**:**512**
manual [13] in advance before trial.	

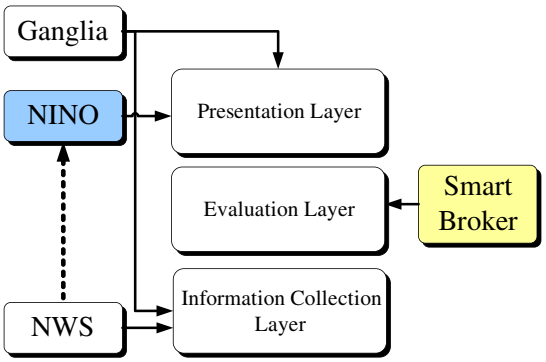

Figure 12. A simple model that integrates Ganglia, Network Weather Service (NWS), NINO, and Smart Broker.

DOI: 10.1002/cpe

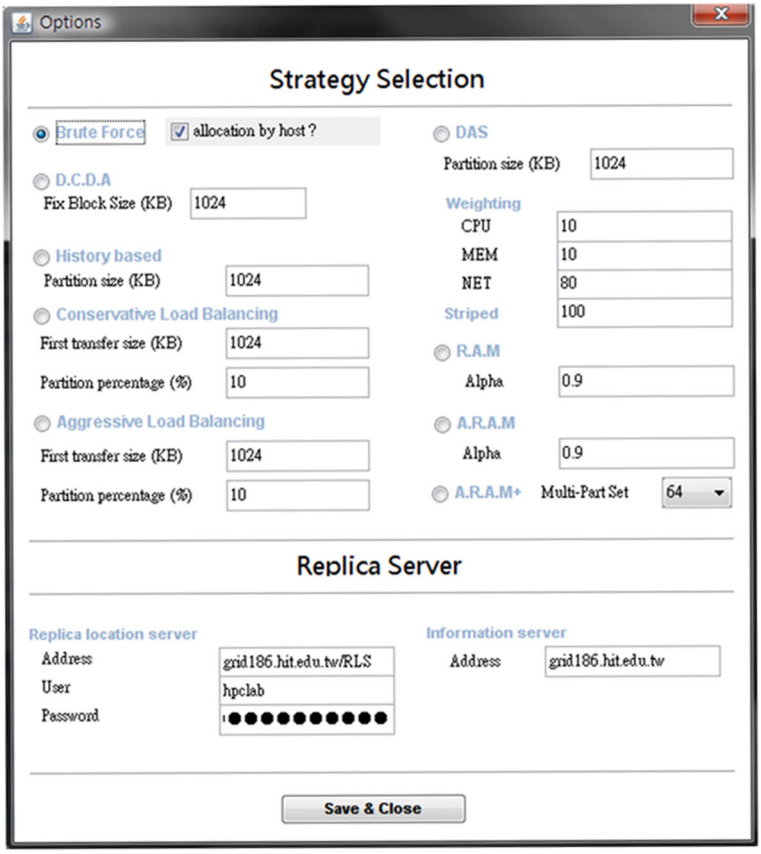

Figure 13. Strategy selection—UI provided by Cyber for parameters input.

Smart Broker is the key component to evaluate QoS. Our previous work [18], [24] provided users with an interface to tune up parameters, as shown in Figure 13. But most parameters used by the application, Cyber, must be set manually and it was very inconvenient for network administrators. Therefore, we developed 'Smart Broker' to achieve automation of self-optimization of parameters in various scenarios. Smart Broker works as the evaluation layer between applications and information collection layers. We have predefined four task types to perform QoS measurement in various ways.

- Download
- Upload
- Computational
- Hybrid

Cyber is a typical application of 'download' type. Figure 14 shows the scenario we used in Cyber. Figure 15 shows the QoS evaluation model we used in this paper. And this evaluation model could be tuned at any time for higher accuracy in different cloud environments.

4. EXPERIMENTAL ENVIRONMENT AND RESULTS

In order to verify the proposed approach, a number of experiments are conducted. Our test-bed consists of 20 cloud host nodes divided into four groups. The physical deployment is shown in Figure 16. NWS services and database deployment is shown in Figure 17. We have adopted a pull-based model to collect network information measured by NWS services, as shown in Figure 18.

DOI: 10.1002/cpe

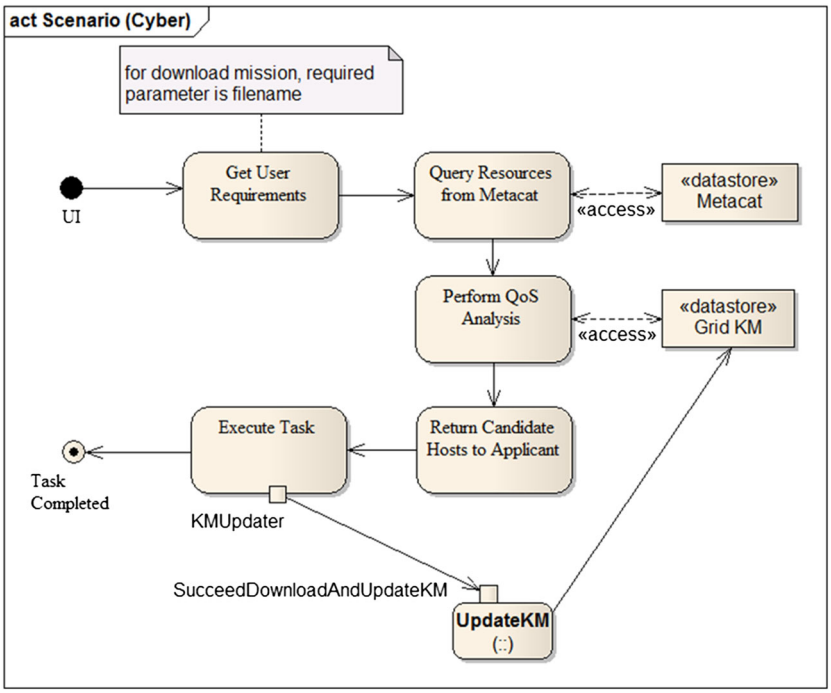

Figure 14. The scenario we used for evaluating QoS.

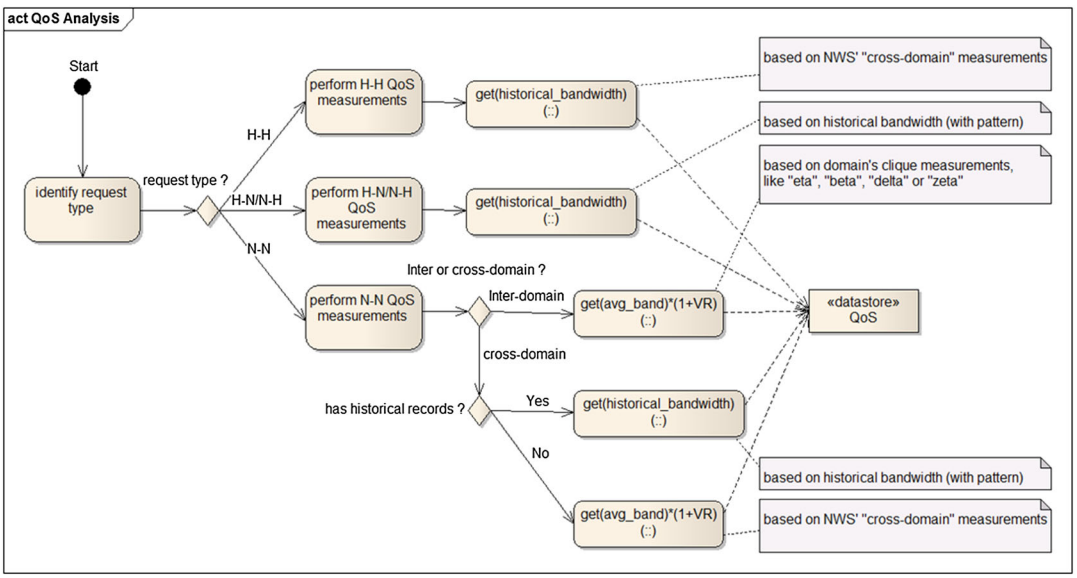

Figure 15. The QoS evaluation model.

Tables III and IV list experimental cloud nodes deployed with NWS services. All cloud nodes were deployed with NWS sensors, and zeta1, beta2, delta2, and eta4 were deployed with both NWS nameserver and memoryserver. Zeta1 was deployed with a routine to collect (pull) all network information measured by NWS services, and to load these raw data into database locally, as shown in Figure 19. Versions of operating systems installed on these cloud nodes are not the same, but it does not influence our work.

DOI: 10.1002/cpe

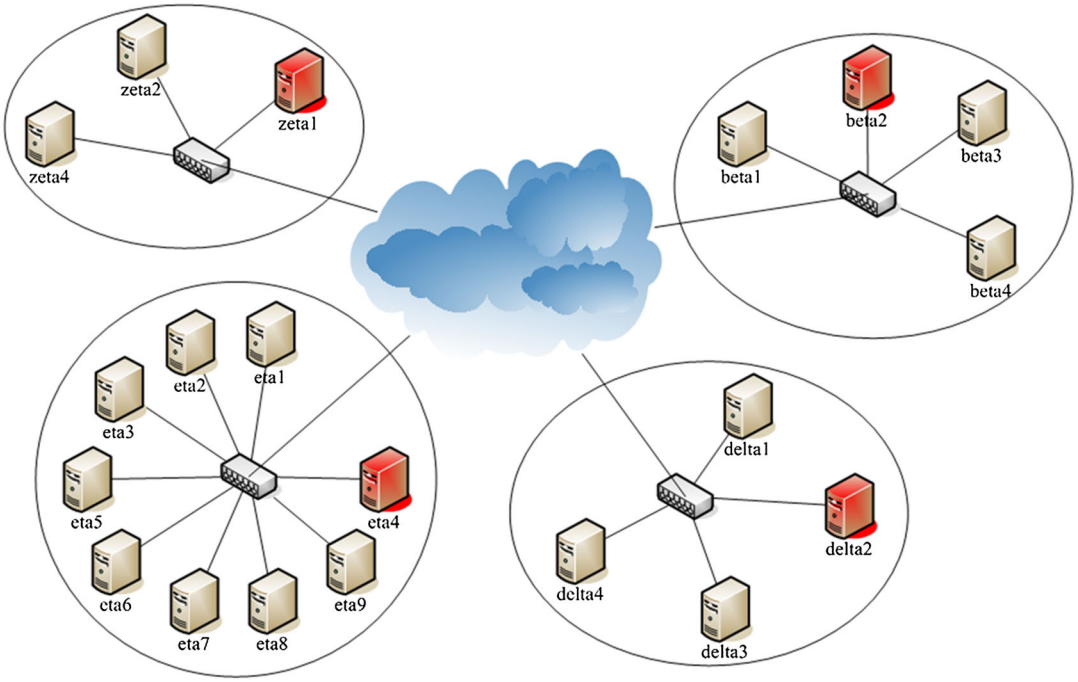

Figure 16. Physical deployment of cloud nodes that we used for test-bed.

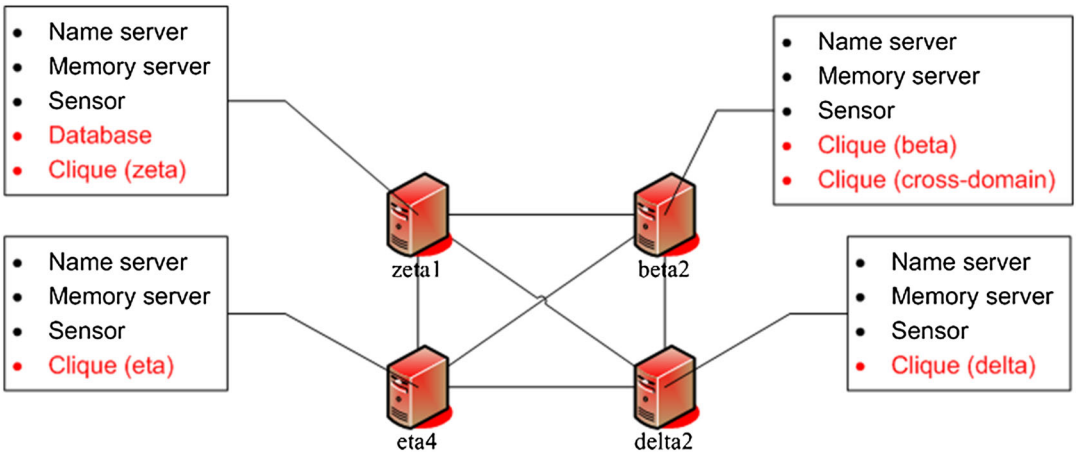

Figure 17. Network Weather Services and database deployment.

In this paper, we chose four cloud nodes as 'header', which is called 'border' in the domain-based network information model, to register specific NWS service—clique for gathering inter-domain network performance. Except these headers, we also registered a NWS clique named 'cross-domain' to measure network performance between these headers. Information collected by NWS services is our basis to evaluate QoS. The experimental scenario shown in Figure 20 is exemplified to ensure that NWS deployment is feasible.

DOI: 10.1002/cpe

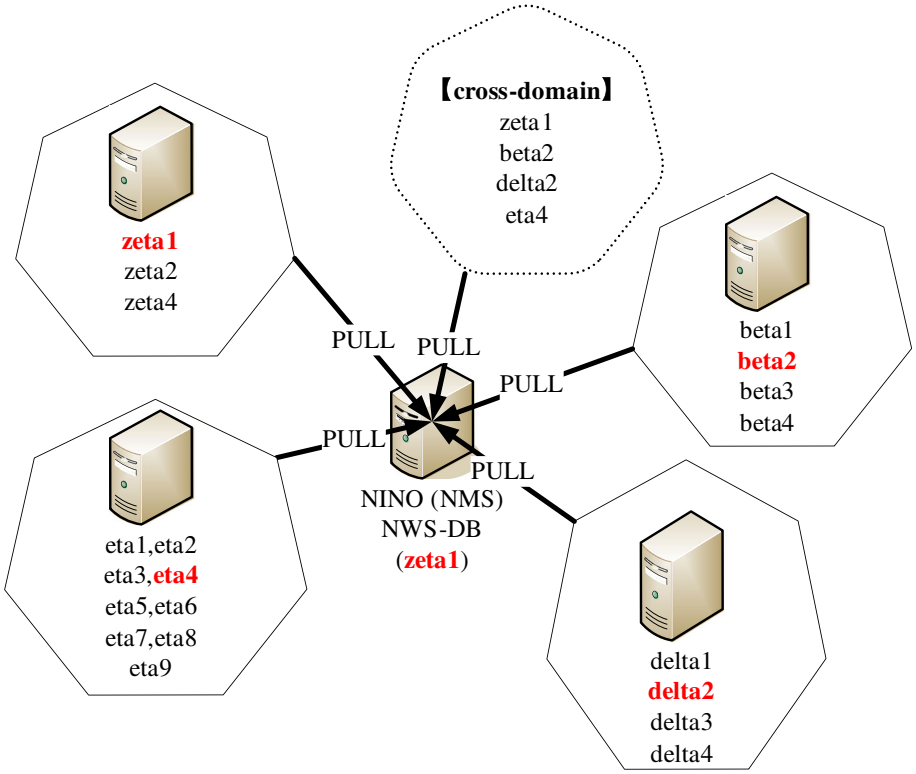

Figure 18. Network Weather Service cliques deployed in our cloud environments.

Table III. Experimental cloud nodes deployed with Network Weather Services.

Clique	Name	Operating System	Clique	Name	Operating System
Zeta	Zeta1	Fedora 8	Eta	Eta1	Fedora 6
	Zeta2	Fedora 8		Eta2	Fedora 6
	Zeta4	Fedora 8		Eta3	Fedora 6
Beta	Beta1	Fedora 5		Eta4	Fedora 6
	Beta2	Fedora 6		Eta5	Fedora 6
	Beta3	Fedora 6		Eta6	Fedora 6
	Beta4	Fedora 6		Eta7	Fedora 6
Delta	Delta1	Fedora 8		Eta8	Fedora 6
	Delta2	Fedora 8		Eta9	Fedora 9
	Delta3	Fedora 8			
	Delta4	Fedora 8			

Table IV. Experimental cloud groups deployed with Network Weather Service clique and their member lists, measurement period.

Clique name	Period (s)	Member
Cross-domain	30	Zeta1, beta2, eta4, delta2
Zeta	30	Zeta1, zeta3, zeta4
Beta	30	Beta1, beta2, beta3, beta4
Delta	30	Delta1, delta2, delta3, delta4
Eta	30	Eta1, eta2, eta3, eta4, eta5, eta6, eta7, eta8, eta9

DOI: 10.1002/cpe

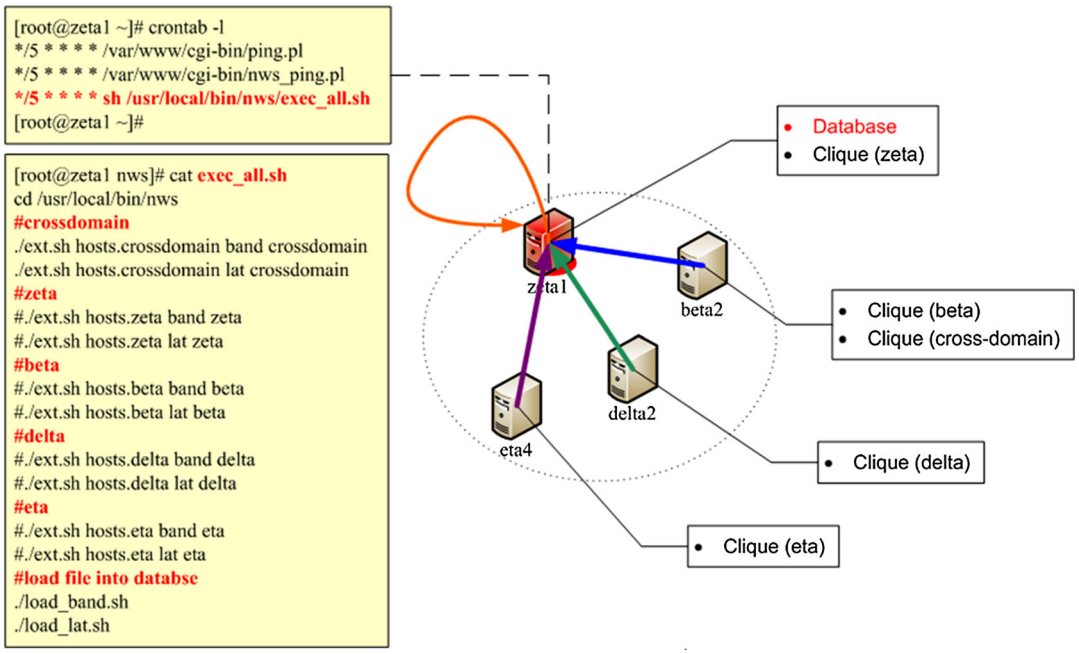

Figure 19. A pull-based model to collect network information measured by Network Weather Service (NWS).

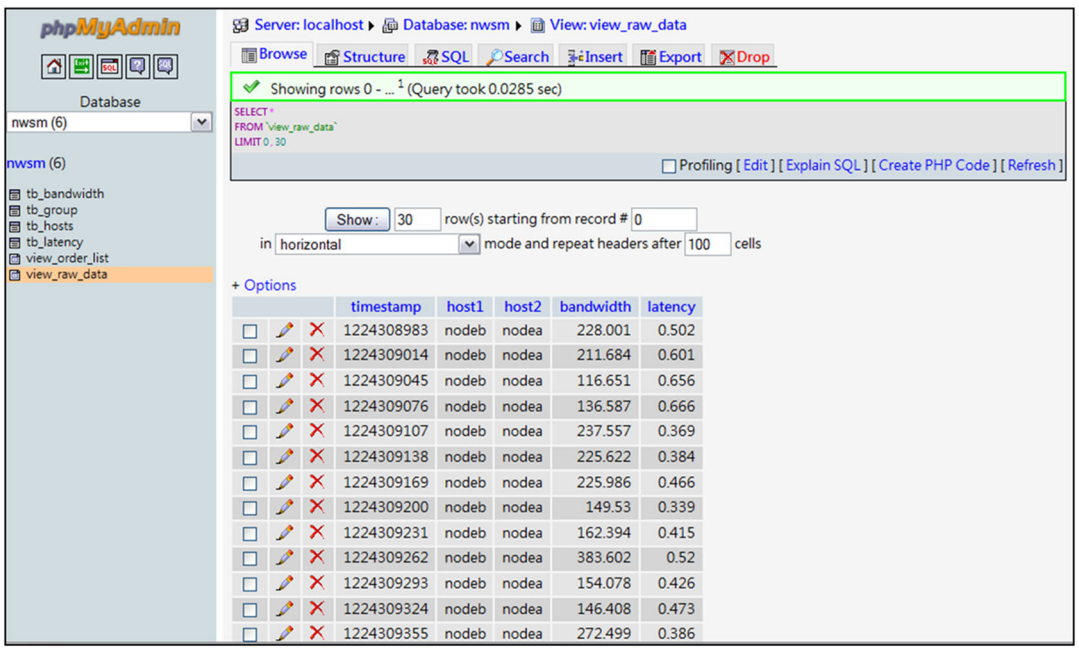

Figure 20. Network information from Network Weather Service (NWS) measurements, both bandwidth and latency, were loaded into MySQL Database.

As shown in Figure 21, we could easily find that the measurements of NWS clique are uneven. For example, eta4-delta2 has minimum measurements, 325, whereas zeta1-beta2 has maximum measurements, 1436. Uneven measurements may influence accuracy of our model while evaluating QoS with statistical approaches.

DOI: 10.1002/cpe

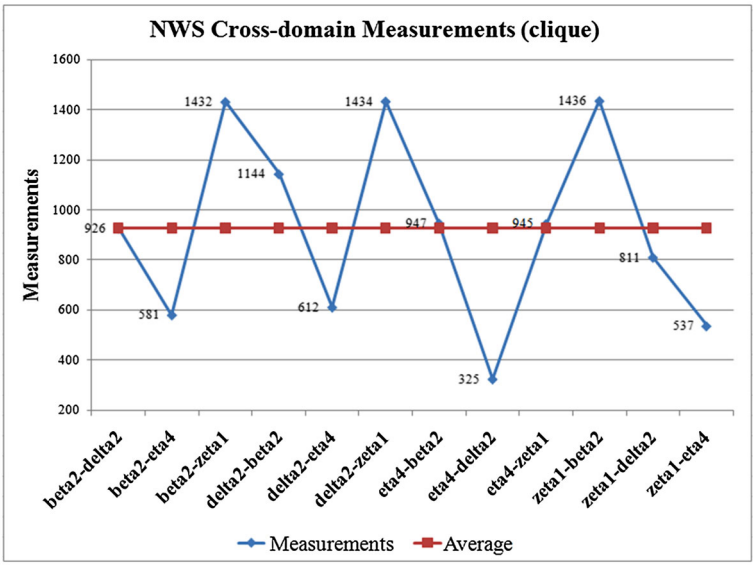

Figure 21. The measurements of cross-domain Network Weather Service clique.

The fact that the NWS services have the ability to avoid collision may cause inaccuracy of measurement, although this feature is restricted in the same nameserver. In our test-bed, we found that collision frequently influence accuracy of measurement. Figure 22 shows results of the collision test for NWS services. We observe that network performance suffers a great variation due to collision of NWS measurement.

Figure 23 shows NWS measurements of our test-bed. Although the adopted QoS evaluation model could not exactly predict real performance for real-time tasks execution, we can still pick best selection of resources by means of the QoS evaluation model. To verify feasibility of this QoS evaluation approach, we also performed file simple transmission experiments. And the result corroborates the predication by using the QoS evaluation model.

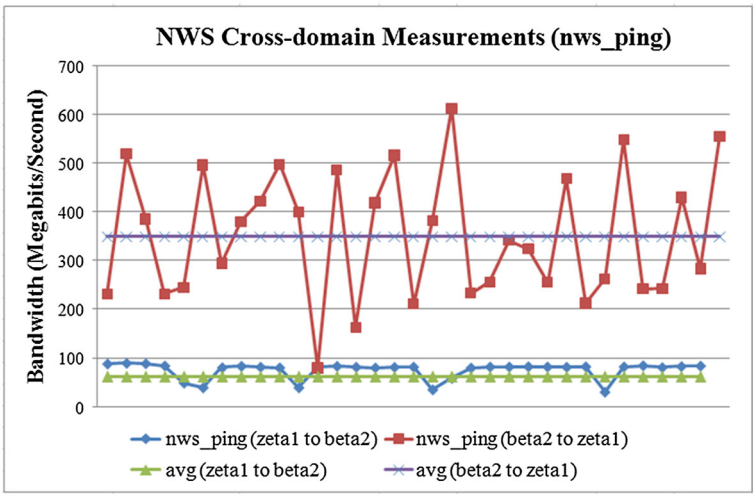

Figure 22. Network Weather Service collision test.

DOI: 10.1002/cpe

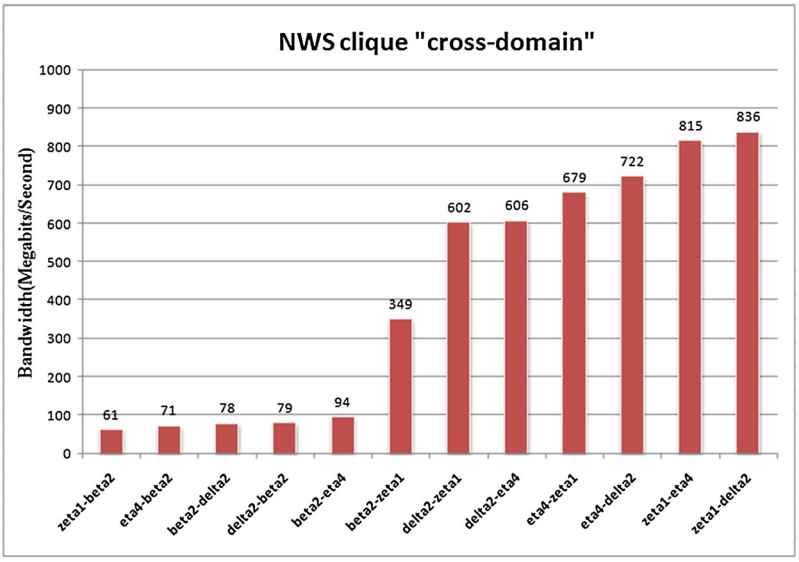

Figure 23. Network Weather Service (NWS) measurements as our basis for QoS evaluation.

5. CONCLUSIONS

In this paper, we first investigated a domain-based network information model, which is not a proper model for dynamic cloud environments. If any cloud node has hardware failure or just been reassigned to another IP, network administrators have to manually reconstruct NWS cliques. This has already been mentioned as drawbacks of NWS [11]. In large-scale cloud environments, it is a complicated task to manage relations between these cliques and hosts. We then adopted dynamic domain-based network information model so as to reduce overheads come from complicated management tasks.

To make network management task more efficiently, we have simplified NWS deployment with a standard procedure for managing cloud nodes semi-automatically. To guarantee a certain degree of QoS, we propose a heuristic method to predict QoS from diverse cloud environments for download-oriented tasks. According to our experimental results, it is found that improper NWS services deployment may cause serious collisions. How to avoid collisions of NWS measurements is a crucial factor to achieve higher accuracy of QoS prediction.

Hence, some practical comments are contributed and concluded as follows:

- Only assign a single clique for every 10 cloud nodes.
- Perform QoS evaluation for each inter-domain cloud nodes first.
- To ensure certain level of accuracy of NWS measurements, stop inter-domain cliques while measuring cross-domain network performance.
- Using the 'nws_ping' utility provided by NWS services to verify the data measured by NWS cliques.

The methods proposed in [7] may be explored in the future research to mitigate measurements workloads. Moreover, the evaluation approach may be improved and adjusted to meet the requirements of other three task types including upload-oriented, computational, and hybrid.

REFERENCES

1. Cloud computing, http://en.wikipedia.org/wiki/Cloud_computing#Infrastructure
2. Milojičić D, Llorente IM, Montero RS. OpenNebula: a cloud management tool. *Internet Computing, IEEE* 2011; **15**(2).
3. Sempolinski P, Thain D. A comparison and critique of eucalyptus, OpenNebula and nimbus. 2010. Cloud Computing Technology and Science (CloudCom), 2010 IEEE Second International Conference, pp.417–426.
4. Cordeiro T, Damalio D, Pereira N, Endo P, Palhares A, Gonçalves G, Sadok D, Kelner J, Melander B, Souza V, Mångs JE. Open source cloud computing platforms. 2010. Grid and Cooperative Computing (GCC) 2010 9th International Conference, pp. 366–371.

DOI: 10.1002/cpe

5. Krauter K, Buyya R, Maheswaran M. A taxonomy and survey of grid resource management systems for distributed computing. *Softw. Pract. Exper.* 2002; **32**(2):135–164.
6. Jarvis S, Spooner D, Keung H, Cao J, Saini S, Nudd G. Performance prediction and its use in parallel and distributed computing systems. *Future Generation Computer Systems* 2006; **22**(7):745–754. doi: 10.1016/j.future.2006.02.008
7. Chung W, Chang R. A new mechanism for resource monitoring in grid computing. *Future Generation Computer Systems* 2009; **25**(1):1–7. DOI: 10.1016/j.future.2008.04.008
8. Ganglia. Ganglia. Retrieved from http://ganglia.info/.
9. MySQL, http://www.mysql.com/
10. Krefting D, Vossberg M, Tolxdorff T. Simplified grid implementation of medical image processing algorithms using a workflow managment system. Presented at the MICCAI-grid workshop, New York. 2008. Retrieved from http://www.i3s.unice.fr/~johan/MICCAI-Grid/website.html.
11. Legrand A, Quinson M. Automatic deployment of the network weather service using the effective network view. 2004. In Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International.
12. Lu J. Perl. Retrieved from http://web.nchu.edu.tw/~jlu/cyut/perl.shtml.
13. Network Weather Service. Network Weather Service. Retrieved from http://nws.cs.ucsb.edu/ewiki/.
14. NINO - Network Manager. Retrieved from http://nino.sourceforge.net/nino/index.html.
15. Que W, Zhang G, Wei Z. Model for IP network synthetical performance evaluation. *Computer Engineering*; **34**(8).
16. Vazhkudai S, Schopf JM, Foster IT. Predicting the Performance of Wide Area Data Transfers. In Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS '02). IEEE Computer Society, Washington, DC, USA, 2002; 270–.
17. Wolski R, Spring N, Hayes J. The network weather service: a distributed resource performance forecasting service for metacomputing. *Future Generation Computer Systems* 1999; **15**(5-6):757–768.
18. Yang CT, Chen CH, Yang MF, Chiang WC. MIFAS: medical image file accessing system in co-allocation data grids. 2008. In IEEE Asia-Pacific Services Computing Conference, 2008. APSCC'08 (pp. 769–774.
19. Yang C, Chen S. A multi-site resource allocation strategy in computational grids. 2008. In Advances in Grid and Pervasive Computing (pp. 199–210). Retrieved July 29, 2009, from http://dx.doi.org/10.1007/978-3-540-68083-3_21.
20. Yang C, Chen S, Chen T. A grid resource broker with network bandwidth-aware Job scheduling for computational grids. 2007. In Advances in Grid and Pervasive Computing (pp. 1–12). Retrieved July 29, 2009, from http://dx.doi.org/10.1007/978-3-540-72360-8_1.
21. Yang C, Chen T, Tung H. A dynamic domain-based network information model for computational grids. 2007. In Future Generation Communication and Networking (Vol. 1, pp. 575–578). Los Alamitos, CA, USA: IEEE Computer Society. doi: http://doi.ieeecomputersociety.org/10.1109/FGCN.2007.9.
22. Yang C, Shih P, Chen S,Shih W. An efficient network information model using NWS for grid computing environments. 2005. In Grid and Cooperative Computing - GCC 2005 (pp. 287–299). Retrieved July 29, 2009, from http://dx.doi.org/10.1007/11590354_40.
23. Yang C, Shih P, Lin C, Chen S. A resource broker with an efficient network information model on grid environments. *The Journal of Supercomputing* 2007; **40**(3):249-267. doi: 10.1007/s11227-006-0025-0
24. Yang C, Yang M, Chiang W. Implementation of a cyber transformer for parallel download in Co-allocation data grid environments. 2008. In Proceedings of the 2008 Seventh International Conference on Grid and Cooperative Computing (pp. 242–253). IEEE Computer Society. Retrieved July 29, 2009, from http://portal.acm.org/citation.cfm?id=1471431.
25. Yinzhe Yu, Irene Cheng, Basu, A. Optimal adaptive bandwidth monitoring for QoS based retrieval. *Multimedia, IEEE Transactions on* 2003; **5**(3):466–472. doi: 10.1109/TMM.2003.814725.
26. Cheng Z, Du Z, Zhu S. A service level QoS mechanism and algorithm for data distribution and backup in an grid based astronomy data management system. 2007. In Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on (pp. 430–436). Presented at the Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on. doi: 10.1109/GCC.2007.25.
27. Fox G, Pallickara S. Recent work in utility and cloud computing. *Future Generation Comp. Syst.* 2013; **29**(4):986–987.
28. The NIST definition of cloud computing. National Institute of Standards and Technology. Retrieved 24 July 2011.
29. Armbrust M, Fox A, Griffith R, Joseph AD, Katz RH, Konwinski A, Lee G, Patterson DA, Rabkin Ariel, Stoica I, Zaharia M. Above the clouds: a Berkeley view of cloud computing, Technical Report, EECS-2009-28, UC Berkeley.
30. Wang L, von Laszewski G, Younge AJ, He X, Kunze M, Tao J, Fu C. Cloud computing: a perspective study. *New Generation Comput* 2010; **28**(2):137–146.
31. Diaz J, von Laszewski G, Wang F, Fox G. Abstract image management and universal image registration for cloud and HPC infrastructures. IEEE CLOUD 2012: 463–470.
32. Diaz J, von Laszewski G, Wang F, Younge AJ, Fox G. FutureGrid image repository: a generic catalog and storage system for heterogeneous virtual machine images. CloudCom 2011:560–564.
33. Wang L, Chen D, Zhao J, Tao J. Resource management of distributed virtual machines. *IJAHUC* 2012; **10**(2):96–111.
34. Wang L, Chen D, Yangyang Hu, Ma Y. Wang J. Towards enabling cyberinfrastructure as a service in clouds. *Computers and Electrical Engineering* 2013; **39**(1):3–14.
35. Youseff L, Butrico M, Da Silva D. Toward a unified ontology of cloud computing, In the Proceedings of the Grid Computing Environment Workshop, 2008, IEE Computer Society Press.
36. Wang L, Fu C. Research advances in modern cyberinfrastructure. *New Generation Comput* 2010; **28**(2):111–112.

DOI: 10.1002/cpe

37. Campegiani P, Lo Presti F. A general model for virtual machines resources allocation in multi-tier distributed systems, autonomic and autonomous systems. 2009, International Conference on, pp. 162–167. Fifth International Conference on Autonomic and Autonomous Systems, 2009.
38. Tung Yang C, Lin CH, Yang MF. A heuristic QoS measurement with domain-based network information model for grid computing environments. 2010, International Journal of Ad Hoc and Ubiquitous Computing.
39. Yang CT. Implementation of a heuristic network bandwidth measurement for grid computing environments. 2010. Lecture Notes in Computer Science.
40. Yang CT, Lin CH, Yu SC. A heuristic network bandwidth measurement with domain-based model on grids. 2009. Communications in Computer and Information Science.
41. Yang CT, Huang KL, Liu JC, Chen WS, Hsu WH. On construction of cloud IaaS using KVM and open nebula for video services. 2012 41st International Conference on Parallel Processing Workshops, 2012
42. Wang L, Tao J, Ranjan R, Marten H, Streit A, Chen J, Chen D. G-Hadoop: MapReduce across distributed data centers for data-intensive computing. *Future Generation Comp. Syst* 2013; **29**(3):739–750.
43. Wang L, Kunze M, Tao J, von Laszewski G. Towards building a cloud for scientific applications. *Advances in Engineering Software* 2011; **42**(9):714–722.
44. Wang L, von Laszewski G, Chen D, Tao J, Kunze M. Provide virtual machine information for grid computing. *IEEE Transactions on Systems, Man, and Cybernetics, Part A* 2010; **40**(6):1362–1374.

DOI: 10.1002/cpe