私立東海大學

資訊工程與科學研究所

碩士論文

指導教授：楊朝棟　博士

於網格計算環境之基於網域資訊模型的啟發式服務質量量測

A Heuristic QoS Measurement with Domain-based Network Information Model on Grid Computing Environments

研 究 生：林志豪

中華民國九十八年六月

摘要

近年來，網格計算愈來愈普及，網格計算帶來大幅提昇的運算效能及分散式資源配置的便利性。但隨之而來的是普遍存在於網格計算環境的共同議題，即如何有效地管理及監控這些龐大的資源。我們經常搭配部署兩套 Open Source 軟體-「Ganglia」及「Network Weather Service (NWS)」，以求有效監控及管理網格計算環境的可用資源，完善的監控與有效率的管理是提昇整體網格運算效能的不二法則。

Ganglia 經常用以監控網格計算環境的資源狀態，如主機運行狀態、CPU、記憶體使用率等，當然 Ganglia 也可用以監控網路相關資訊，但我們經常以 NWS 來量測網格計算環境中計算節點，點對點之間的網路頻寬。相較於 Ganglia，NWS 可提供更大的應用彈性與較多選擇性的量測機制，且非侵入式的安裝方式，使我們更容易將 NWS 部署到各計算節點，迅速、便利地取得計算節點間的網路相關資訊。NWS 同時提供量測 CPU 或記憶體的機制，但在這方面功能性就無法與 Ganglia 比擬，也因此我們經常將這兩套軟體搭配使用，互補長短，以達成有效監控與管理網格計算環境的相關資源。

然而在實務應用上，Ganglia 與 NWS 並無法完全滿足不同應用情境下的使用者需求，特別是應用程式開發者。例如，用戶端希望藉由 Web Services 或應用程式，透過適當介面或管道取得網格計算環境的資源狀態及配置情形，目前 Ganglia 與 NWS 並無法直接提供這樣的服務。而基於網域資訊模式的服務部署，可有效減少量測次數及資源損耗。因此，在這篇論文中我們提出基於網域資訊模式的啟發式服務質量量測方法，從服務質量的觀點評估網格計算環境所能提供的資源與效能，以滿足不同使用者對網格計算環境的服務要求。

此外，我們經由實驗設計，架構出一個精簡的部署模式與模型，期能使後續研究人員可以引用參考，減少技術障礙，而能以使用者需求導向為訴求，發展格網計算應用服務。

我們希望不論是網格計算環境的使用者或管理者,都能藉由本論文提供的量測方法，更有效地管理、配置網格資源。

關鍵字：網格計算、啟發式、QoS、服務質量、網域資訊模型

Abstract

Recently, Grid computing is more and more common and widespread. Therefore, there exists a common issue, i.e., how to manage and monitor numerous resources of grid computing environments. In most cases, we use Ganglia and NWS to monitor Grid nodes' status and network-related information, respectively. With supports of Ganglia and NWS services, we could effectively monitor and manage available resources of our grid environments. Comprehensive monitoring and effective management are criterions to archiving higher performance of grid computation.

Ganglia is often adopted to monitor resources' status, like hosts' live status, CPU or memory utilizations, in grid environments. Certainly, Ganglia also has the ability to monitor network relative information. Instead of Ganglia, more often than that, we use NWS services to measure network relative information, like end to end TCP/IP performance. Compare to Ganglia, NWS services provide more flexibility and choices for measurement mechanism. Besides, NWS services could be deployed with non-intruding manner which could help us to deploy services to each grid nodes rapidly and easily. We could obtain network relative information in a short term following deployment. NWS services also provide measurements for CPU or memory utilizations. But NWS provides less functionality than Ganglia in this dimension. Therefore, we combine services provided by both Ganglia and NWS mostly to meet our requirements to effectively monitor and manage available resources of our grid environments.

Unfortunately, owing to diverse user requirements, information provided by Ganglia and NWS services is not sufficient in real cases, especially for application developers. For example, users couldn't directly retrieve utilizations or allocations of resources in grid environments through proper "interface" or "channel" with help of Ganglia or

NWS. In addition, NWS services that deployed based on “Domain-based Network Information Model” could greatly reduce overheads caused by unnecessary measurements.

Therefore, in this thesis, we propose a heuristic QoS measurement which is constructed with domain-based information model. This measurement has ability to provide more effective information to meet user requirements, especially for application developer. We hope users could manage and monitor numerous resources of grid environments more effectively and efficiently.

Keywords: Grid Computing, Heuristic, QoS, Network Information Model

Acknowledgements

回憶當初報考研究所只有一個念頭就是為了再深造，充實自己的理論基礎。上完學分班課程後順利考上研究所，距離上個學歷，回首已是十年後。兩年時間過得飛快，總覺得有做不完的題目可以研究，箇中酸楚只有經歷過碩士這個學習歷程的人可以瞭解吧！

有幸地能請楊朝棟 博士擔任指導教授，並加入老師指導的高效能計算實驗室，在此真誠感謝老師的悉心指導，予我在學期間諸多耐心與關懷、支持與鼓勵，使我能順利渡過這個學習階段。期間無論是參與國際研討會的論文發表或者是在課堂之中，碩士的學習歷程都充滿挑戰及成就感。誠如老師在我們加入實驗室前的勉勵，在職班學生要在兩年內畢業是很辛苦的，要有充份的心理準備。雖然工作繁忙，兼以研究學習的雙重壓力常常讓我加班熬夜，甚至無法入眠，但辛苦的成果是甜美的，且造就出的韌性與培養的專業知識相信有助於因應未來更多挑戰。

其次感謝家人的支持與體諒，讓我有堅實的後盾挺著我持續前進、奮鬥。

最後，感謝研究所的同學們，明峰、秋雄與實驗室的學長姊、學弟妹們，你們的無私付出，讓我的研究工作順利進行，感謝你們，並再次深深感謝楊老師的指導。

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

As we known, Grid computing technique is more and more popularly adopted by organizations to obtain high performance computing and heterogeneous resources sharing. Since all computing nodes in grid environments are connected by means of network, all tasks that executed in grid environments will be influenced by network status due to complicated and numerous communications between computing resources. While we design algorithms for specific usages or assign tasks into grid environments, we are forced to evaluate the performance of network from related information and adjust algorithms or specific parameters to try to attain optimal performance in real-time execution. The best scenario is that our grid environments have some mechanisms to retrieve network status and evaluate performance automatically. Thus, applications or web service agents could provide higher performance due to dynamic parameters adjustment and algorithms optimization.

While grid computing becomes widespread gradually, it brings about a common issue, i.e., how to manage and monitor numerous resources of grid computing environments. In most cases, we use Ganglia and NWS to monitor machines' status and network-related information, respectively. Owing to diverse user requirements, information provided by these services is not sufficient in some scenarios.

According to the mechanism that we designed in previous work, we could retrieve relative network information in real-time manner; even advanced customization for special purpose is available. With the customized shell scripts that we wrote for NWS services' deployment, we could easily and quickly deploy NWS services to each grid nodes and fetch network-related information in a regular time interval. Besides, we could obtain extra statistics for job-scheduling in our grid environments. Except job-scheduling, statistics is also helpful in many respects.

In our previous work, we found that the service provided by NWS will be affected if grid environment changed and then we have to frequently re-deploy NWS services manually. "Manual" is equivalent to "inefficiency" in network management. A typical example is illustrated in Figure 1.1. If we have registered a NWS clique into grid nodes A1, A2, A3 and A4 and the header is set to A1, i.e., A1 will be stored network measurements from these nodes. While hardware failure occurs to A1 or A1 has just forced to reboot due to software updating operations, then the NWS clique terminates, too. Network administrators have forced to restart cliques manually again and again. Besides, we won't be notified if any nodes fail by default. Therefore, we will lead in a network management system that using SNMP technique to co-work with NWS service to resolve this issue in the near future.

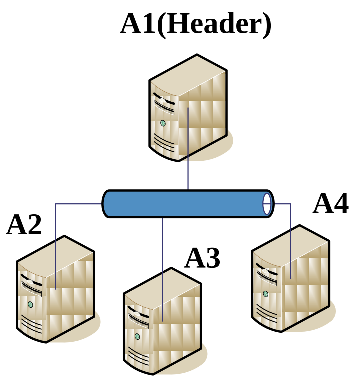

Figure 1.1: A typical NWS clique deployment in grid nodes

Before achieving dynamic detection and recovery of NWS services, we still need to find some approaches to manage our grid resources effectively and efficiently. Hence, we propose a heuristic QoS measurement constructed with domain-based information model that providing more effective information to meet user requirements. Furthermore, we hope that users could manage and monitor numerous resources of grid environments more effectively and efficiently. And for application developers, we expect to provide a generic interface to evaluate network performance before accessing any grid resources.

1.2 Contribution

In this thesis, we propose a heuristic QoS measurement constructed with domain-based information model. We have simplified deployment of NWS services to easy steps, and then NWS services could be quickly and easily deployed to grid machines for fetching network information regularly without intruding existed systems. And by using Relational Database Management System (RDBMS) [5], we could keep historical network information and calculate statistics in advance for QoS evaluation. Statistics is helpful in many fields, for example, job dispatching or replicas selection. In this thesis, the QoS measurement we proposed helps to reduce complexity of prediction of network status and provide a generic interface to evaluate network performance before accessing any grid resources.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. Background review and studies are presented in Chapter 2. Heuristic QoS measurement is given in Chapter 3. Experimental environment and results are presented in Chapter 4. Chapter 5 concludes this research article.

Chapter 2

Background review and related work

2.1 Machine information provider

The Ganglia [4] is an open source project grew out of the University of California, Berkeley's Millennium initiative. The Ganglia is a scalable distributed system for monitoring status of nodes (processor collections) in wide-area systems based on Grid or clusters. It adopts a hierarchical, tree-like communication structure among its components in order to accommodate information from large arbitrary collections of multiple Grid or clusters. The information collected by the Ganglia monitor includes hardware and system information, such as processor type, CPU load, memory usage, disk usage, operating system information, and other static/dynamic scheduler-specific details. It also provides a web portal for users to observe all machines via web interface. Our grid environments are currently overseen by ganglia and we could oversee several clusters or grids environments via web portal provided by Ganglia.

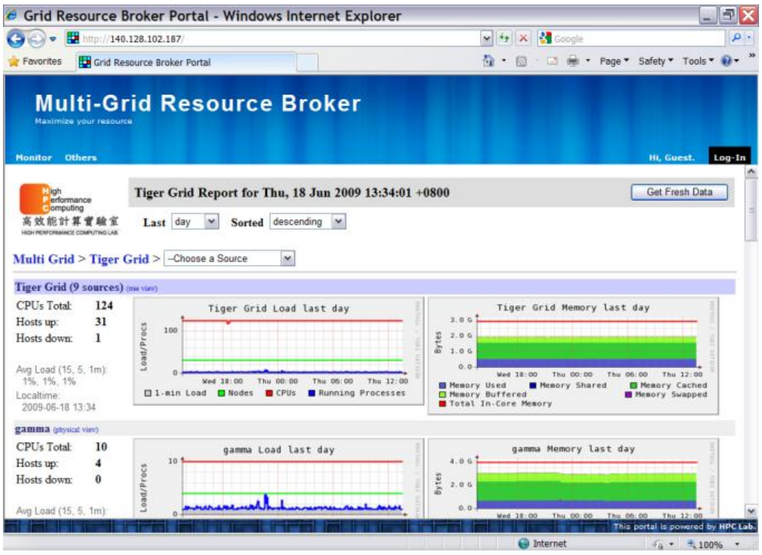

Figure 2.1: Multi-Grid Resource Broker with Ganglia web portal

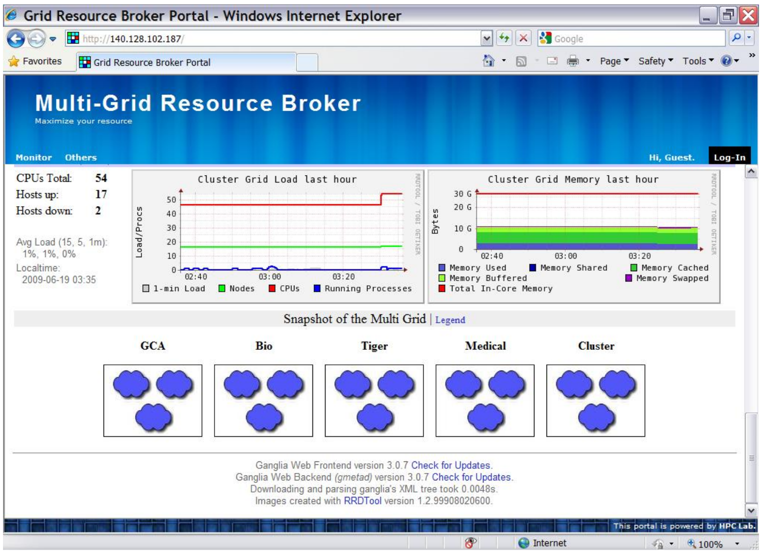

Figure 2.2: Multi-Grid has integrated clusters and grids environments into a single Ganglia web portal

2.2 Network information provider

The NWS (Network Weather Service) [9] [13] is a distributed system that detects network status by periodically monitoring and dynamically forecasting over a given time interval. The service operates a distributed set of performance sensors (network monitors, CPU monitors, etc.) from which it gathers system condition information. It then uses numerical models to generate forecasts of what the conditions will be for a given time period. The NWS system includes sensors for end-to-end 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 primarily use NWS for end-to-end TCP/IP measurements.

As Rich Wolski said [13], NWS is designed to maximize four possible conflicting functional characteristics. It must meet these goals despite the highly dynamic execution environment and evolving software infrastructure provided by shared meta-computing systems.

■ Predictive Accuracy

■ Non-intrusiveness

■ Execution longevity

■ Ubiquity

We have successfully developed a number of shell scripts for automatic NWS deployment. And these scripts form a basis for NWS services' management. And we have successfully integrated NWS services with Ganglia web portal which is shown as Figure 2.3 and Figure 2.4 below.

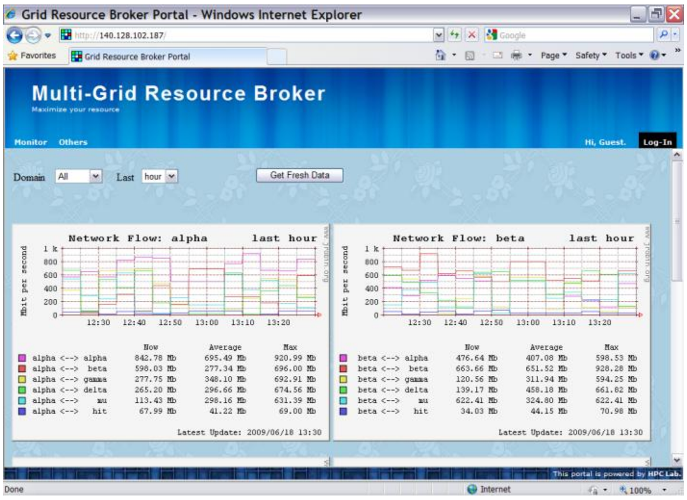

Figure 2.3: NWS services integrated with Ganglia web portal

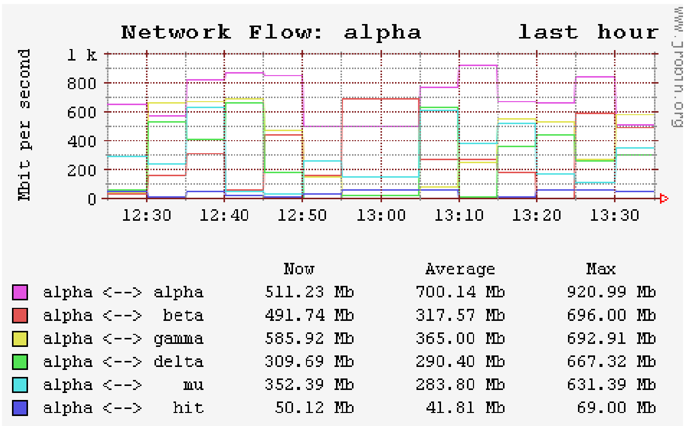

Figure 2.4: Network statistics produced by NWS measurements demonstrated in web portal

2.3 Quality of service

QoS (Quality of service) [21] [22] is the ability to provide different priority to different applications, users, or data flows, or to guarantee a certain level of performance to a data flow. It was widespread adopted in the field of computing networking, and we use it as a quality measurement of grid environments. Quality of service sometimes refers to the level of quality of service, i.e. the guaranteed service quality. High quality of service is an expectable crucial factor of highly reliable and high performance grid environments.

Some characteristics, like "Availability", "Accessibility" or "Maintainability", will also influence user experiences about the services provided by our system or services. To meet user requirements in diverse scenarios with sufficient quality, we are expected to have the ability evaluating performance in advance. If not, how could we guarantee a certain level of QoS? Some researcher have proposed network performance evaluation model [22], [11] to help network administrators to effectively analyze network performance and then adjust network devices properly. In this thesis, we try to propose a heuristic QoS measurement that could evaluate network performance in advance.

2.4 Network Management System

In our previous work, we have constructed a web portal composed of Ganglia and NWS services for overseeing grid environments. As time goes on, we find that it's inefficient to manage these resources passively. We had better make use of NMS (Network Management System) which could help us to manage and monitor numerous resources of grid environments actively. The primary communication mechanism between NMS and network devices and grid nodes is based on SNMP (Simple Network Management Protocol). We have chose NINO [15] [25] as our experimental NMS. NINO is not the most powerful NMS, but it's sufficient to form a prototype of our model. We believe that we could integrate NMS, like NINO, with our previous work and help us to manage grid environments more actively. Meanwhile, we are working on integration of Ganglia, NWS, and NINO.

In short, we use NMS like NINO to detect NWS services and operation status of grid nodes. If any failure occurs, we are expected to be notified by NMS like NINO. And then we could adopt some proper means to recover services semi-automatically or full-automatically. A couple of service functions written with PERL [8] are under development.

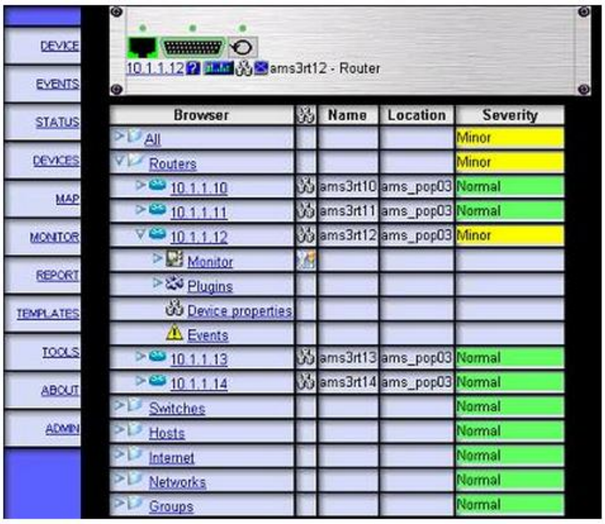

Figure 2.5: A snapshot of NINO, which has the ability to scan many types of devices of fabric layer, like hosts, switches, routers, etc.

Chapter 3

Heuristic QoS Measurement

In our previous project, we have built a integrated grid environments including a web portal composed of Ganglia and NWS service. Afterward, we start another project about PACS (Picture Archive and Communication System) [14] and most experiments were done in the same platform. The primary mission in this project is to exchange medical images efficiently with specific application developed by our team. The application, named "Cyber [20]", has successfully integrated eight algorithms. For exchanging medical images efficiently with these algorithms integrated in Cyber, we have to configure a lot of parameters before tasks submitted. Unfortunately, we have no idea what's best combination of parameters we should take in advance. Therefore, we adopt "trial and error method" unavoidably. But it's definitely not practical for most conditions [12]. For this reason, we expect to establish an automation of parameters self-optimization. To guarantee a degree of QoS, we regard user requirements as constraints of tasks. With these constraints and heuristic QoS measurements we proposed in this thesis, we could provide more QoS to meet user requirements.

3.1 Domain-based network information model

In this thesis, we adopt Domain-based Network Information Model [16], [18], [19] for NWS services deployment. The Domain-based Network Information Model is designed for solving a complete point-to-point bandwidth measurement problem. After investigating by experiments in physical environments, we can sure that Domain-based Network Information Model is helpful for reducing network measurements. The measurement model and design of Domain-based Network Information Model are shown as Figure 3.1 and Figure 3.2 below.

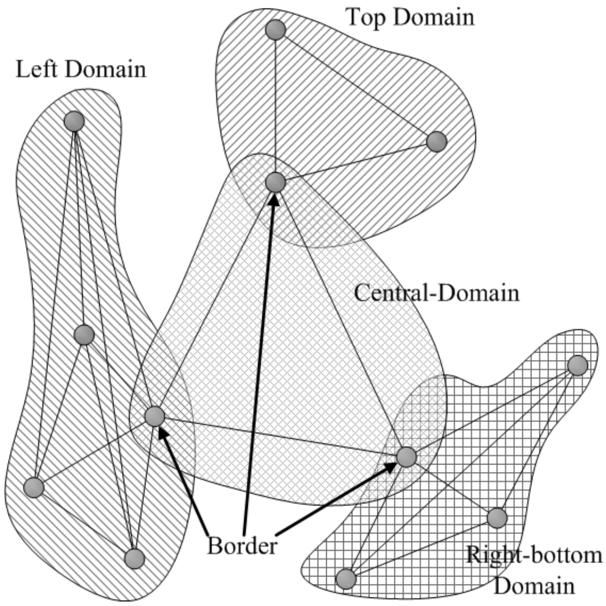

Figure 3.1: The domain-based network measurement model

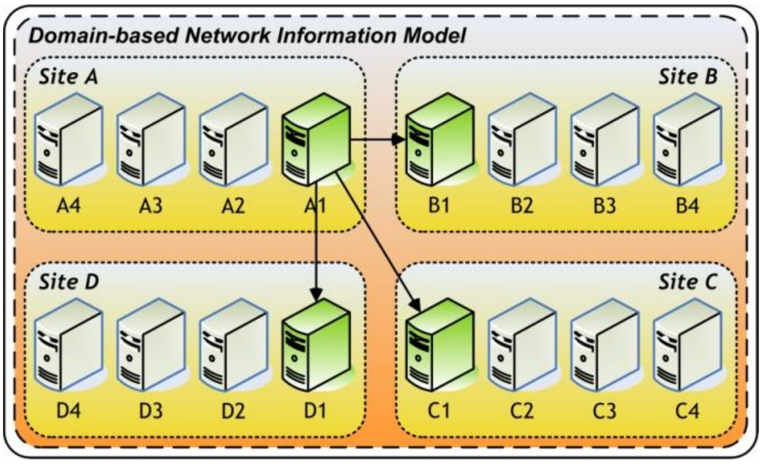

Figure 3.2: The design of domain-based network information model

For example, assume a Grid with nodes. Each node measures the links between itself and all other nodes every T seconds (e.g., T=1~3 sec) for a total of *NMN* (*n*) network measurements.

$$NMN(n) = n \times (n-1)\ . \quad (1)$$

In large-scale Grid environments, the number of network measurements grows quickly. In our test-bed, we have 20 hosts which could generate *NMN* (20) = 380 measurements. Thus, network traffic will be very heavy, particularly when underlying Grid intra-traffic is originally busy.

Our previous work [18] used the domain-based network information model shown in Figure 3.2. This figure shows four sites, each containing four nodes. The sites each have a head node, e.g., A1, B1, C1 and D1, are, respectively, the head nodes of sites A, B, C and D. Each head node in this model periodically measures the links between itself and the other three head nodes. Each head node also periodically measures the links between itself and all other nodes in its site. Hence, using the domain-based network information model, the measurement number will be dramatically reduced to

$$NMS(n,[n_i]) = NMN(n) + \sum NMN(n_i) , \qquad (2)$$

Where *ni* is the total number of nodes in site *i*. In our test-bed, the numbers of network measurements will decrease to *NMS*(5, [9,4,4,3]) =102. The reduction rate *R* is defined as:

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

Compared to *NMN* (20), the *Rs* are 73.16%, which shows the obvious efficiency of the model.

Even though this model can eliminate huge amounts of measurement effort and bandwidth use, it lacks network information between pairs of nodes belonging to different sites (unless both are borders). For example, the link (target) between nodes A2 and B1 shown in Figure 3.2 is not measured.

In this model, it reduces a large number of connections, but it lacks network information of Nodes except head Nodes in two different Sites. This model carries out an estimation model that provides network information of Nodes in two different Sites, but one of two Nodes should be a head Node of Site. For example, the link between Node A2 and B1 is not performed in this model which is shown in Figure 3.2.

The Domain-based Network Information Model reduced the number of bandwidth measurement between all Grid Nodes, but it lacks network information between two Nodes other than the head Node located in two different Sites other than the head Node. For example, the bandwidth measurement between Nodes A2 and B3 is not performed in this model.

We further enhanced the static model by improving the switching mechanism in the dynamic domain-based network information model. Figure 3.3 shows an example. The principal improvement is switching the site head node to the next free node. For example, when node A1 is busy, the next free node, node A2, becomes the head node of site A, and measures the bandwidth between itself and nodes B3, C2, and D4, if they are the respective free nodes in sites B, C, and D. The purpose is to avoid having a busy node still act as a border, which would decrease system performance. There are three obvious advantages in using this model.

- First, the number of bandwidth measurements is the same as that for a static model; the measurement time complexity is not worsened.
- Second, bandwidth measurements between pairs of arbitrary nodes belonging to different sites are easily obtained.
- Finally, network bandwidth measurements obtain real values instead of estimated values, thus enabling the Resource Broker to effectively schedule jobs allocated to multiple sites.

If we could dynamic change each header of all Grid Nodes, we could obtain the advantages described above. These is an issue derived from this requirement consequently, i.e. how to choose headers dynamically instead of manually operation. Hence, we regard Heuristic QoS Measurement as a solution for this issue. After integration with NINO, we expect to attain this goal as soon as possible.

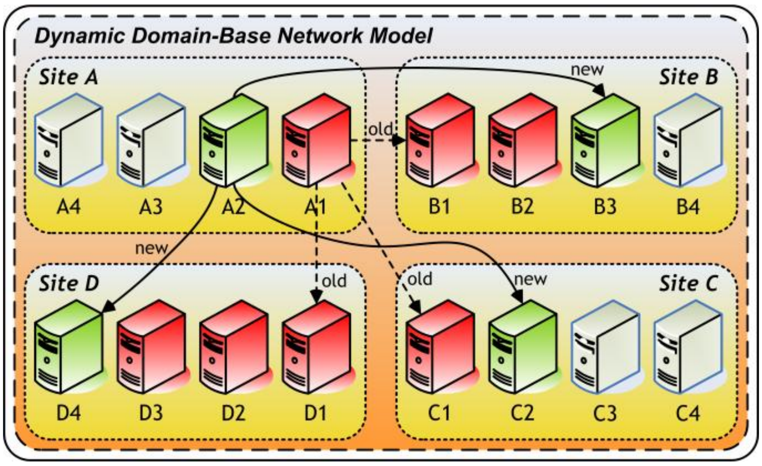

Figure 3.3: The design of domain-based network information model

3.2 NWS deployment and flowchart

While deploying NWS services, we paid attention to tried to get rid of intruding existed services on each grid nodes. In most cases, we deploy only one nameserver and multiple sensors on each computing resources. Besides, arbitrary "Persistence State" may be set up in different locations. In this thesis, we simply designate one nameserver, one memory server, and one clique for a group of grid nodes.

We regard several grid nodes as a group, and each group has a header to deploy nameserver and memoryserver. A simple NWS services deployment procedure that we used is divided into 3 steps:

1. Clean all NWS process.
2. Load NWS services.
3. Register NWS clique.

And the standard procedure we wrote in shell scripts is shown as Figure 3.4 below. Owing to the non-intrusiveness characteristic of NWS, these shell scripts we wrote could be executed without root privilege.

Figure 3.4: Procedure of NWS services deployment

Figure 3.5 has shown a simple flowchart we used. In this thesis, we have edit crontab to schedule some routines for loading NWS information into database automatically and backing up raw data as plain text files locally.

While routines that we scheduled in crontab are invoked, customized shell scripts that we wrote are executed. The first step of the shell script is to get host groups from database for NWS information gathering. Each host groups is pre-defined in database and will be assign a clique for measuring network status. After the clique is created, it will measure network information in an equal time interval, for example, 30 seconds. Then the script will extract bandwidth and latency from NWS clique respectively. If successes, it will load bandwidth and latency information into database.

The second routine that we defined to keep raw data as plain text files locally is designed for future use. Currently, it just provides a different storage than database to keep raw information of NWS services.

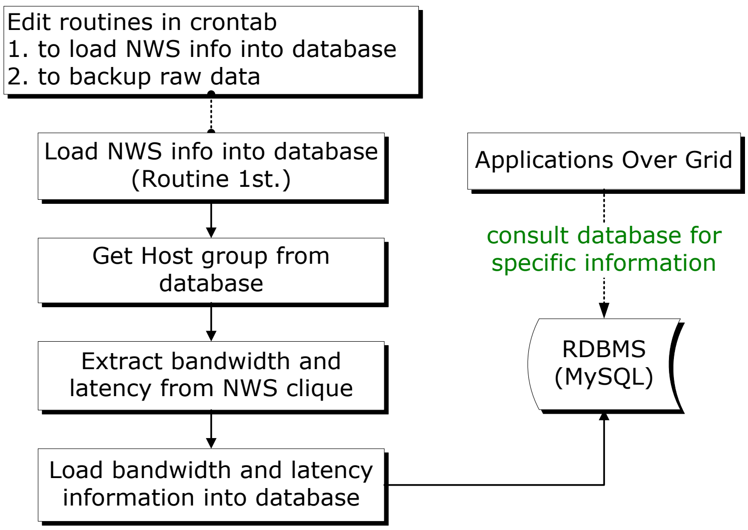

Figure 3.5: The flowchart of gathering network information

What we described above helps us successfully keep network information into database in a regular time interval. But for migrating to dynamic NWS services detection and recovery, we have to simplify not only deployment but also practical usages. After several trials, we have successfully simplified NWS services operations to 3 shell scripts [23] [24]:

Table 3.1: Simplified NWS 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

Proper parameters for NWS clique is key to accuracy of NWS measurements, Table 3.2 shows a sample set of parameters that we used for measurement.

Table 3.2: A sample set of parameters of NWS clique

Clique: A **control** coordinates experiments between a set of NWS sensors, called the members of the clique **Guideline:** To achieving higher accuracy of measurement, the attributes "**size**", "**message**", and "**buffer**" should adjust for different grid environments. We used "**nws_ping**" as a tool to obtain proper configuration before deploying NWS clique. This technique requires experiences and concept about NWS core services. You had better read NWS manual [9] in advance before trial.	name:cross-domain controlName:clique skillName:tcpMessageMonitor period:60 member:zeta1 member:beta2 member:delta2 member:eta4 **size:24000** **message:256** **buffer:512**

3.3 Heuristic approach

Statistics is helpful in many fields, especially for prediction. Some researchers have used statistical method to monitor and predict bandwidth for QoS sensitive task [21]. In this thesis, we collected historical network information of grid environments and found an approach to evaluate QoS. We could give applications dedicated parameters in a simple manner by means of database operations. Couples of functions have been designed for analyzing historical information of network performance. All network relative information was periodically categorized to most used statistics. Applications could dynamically adjust their parameters about network for better performance and have no need sending request to estimate network status between all grid nodes in real-time manner.

Besides, we have planned an innovative method to obtain real-time network state that worked with Dynamic Domain-based Network Information Model, i.e.

dynamically deploying clique into dedicated node, measuring network state, and then reporting results to database, users, or applications. The enhanced version of current work which supports Dynamic Domain-based Network Information Model is currently under development.

We have design a simple model for integration of Ganglia, NWS, and NINO (as shown in Figure 3.6). Ganglia and NINO provide UI for users to manage and monitor grid environments. NWS and Ganglia collect related information from hosts and network regularly. And "Smart Broker" provides parameters to applications like Cyber.

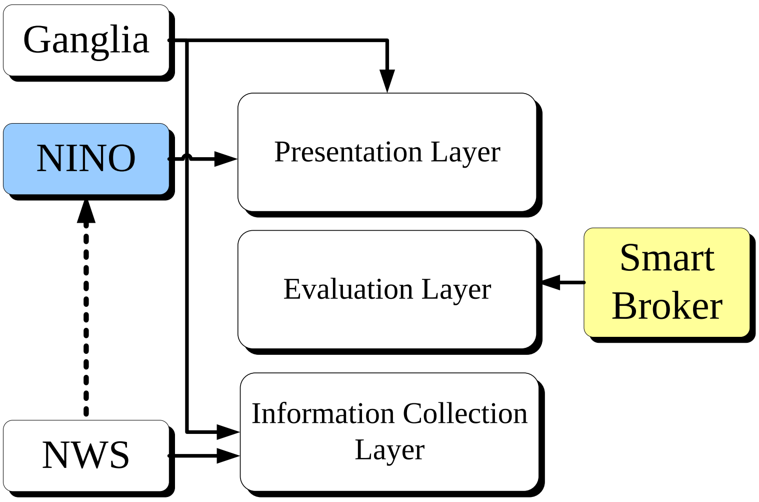

Figure 3.6: A simple model that integrate Ganglia, NWS, NINO, and Smart Broker

Smart Broker is the key component for us to evaluate QoS. Our previous work [14], [20] has provided users an interface for tuning up parameters which is shown as Figure 3.7 below. But most parameters used by this application, Cyber, must be set manually and it's very inconvenient. We developed "Smart Broker" to help us to achieve automation of parameters self-optimization in diverse scenarios. Smart Broker works as evaluation layer between applications and information collection layer. We have pre-defined 4 task types that perform QoS measurement in various ways.

- Download
- Upload
- Computational
- Hybrid

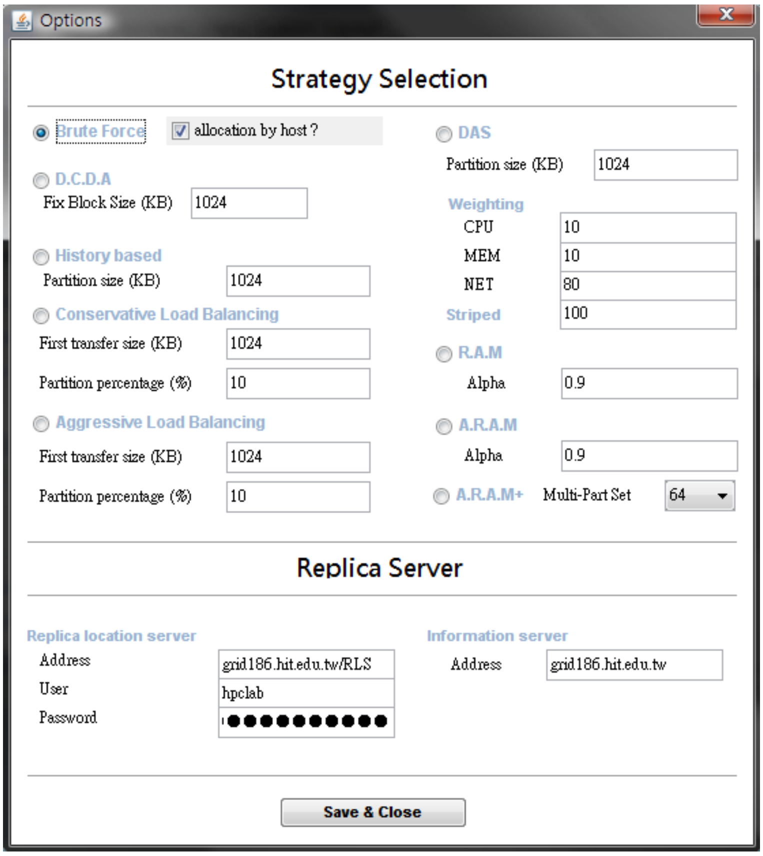

Figure 3.7: Strategy selection – UI provided by Cyber for parameters input

Cyber is a typical application of "Download" type. Figure 3.8 shows the scenario we used in Cyber. Figure 3.9 shows the QoS evaluation model we used in this thesis. And this evaluation model could be tuned at any time to approach higher accuracy in different grid environments.

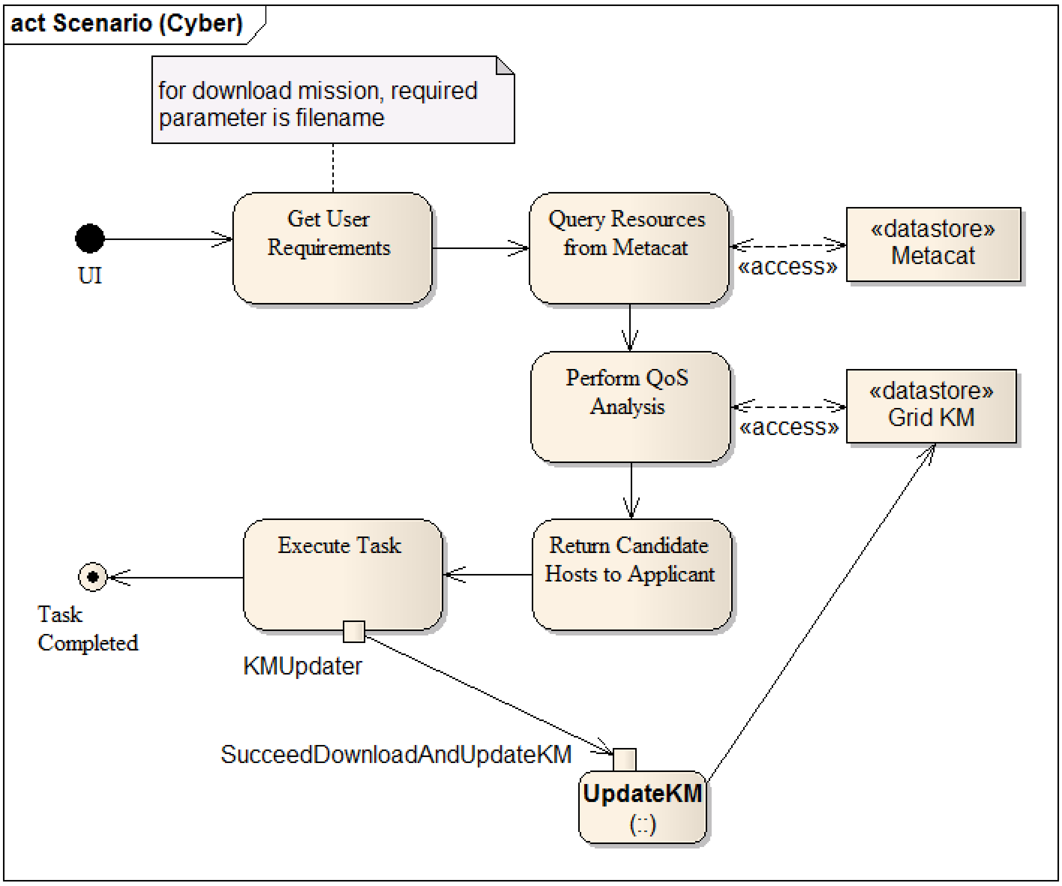

Figure 3.8: The scenario we used for evaluating QoS in this thesis

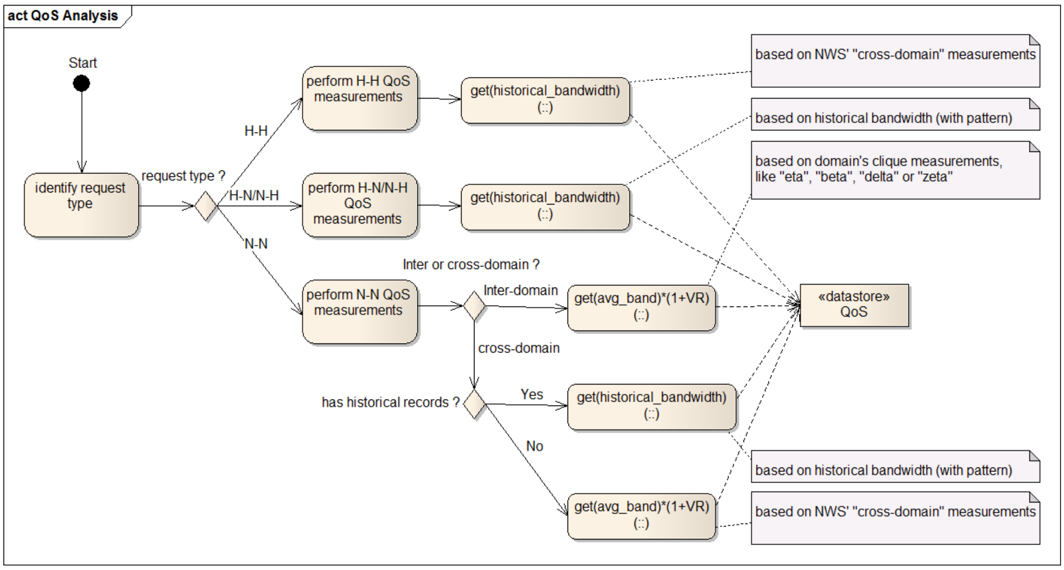

Figure 3.9: The QoS evaluation model we used in this thesis

Chapter 4

Experimental Environments and Result

In order to verify the architect we proposed in this thesis, we have performed couples of experiments. Our test-bed has 20 grid nodes and all these hosts were divided into 4 groups. Physical deployment is shown in Figure 4.1. NWS services and database deployment is shown in Figure 4.2. We have adopted a pull-based model to collect network information measured by NWS services as shown in Figure 4.3.

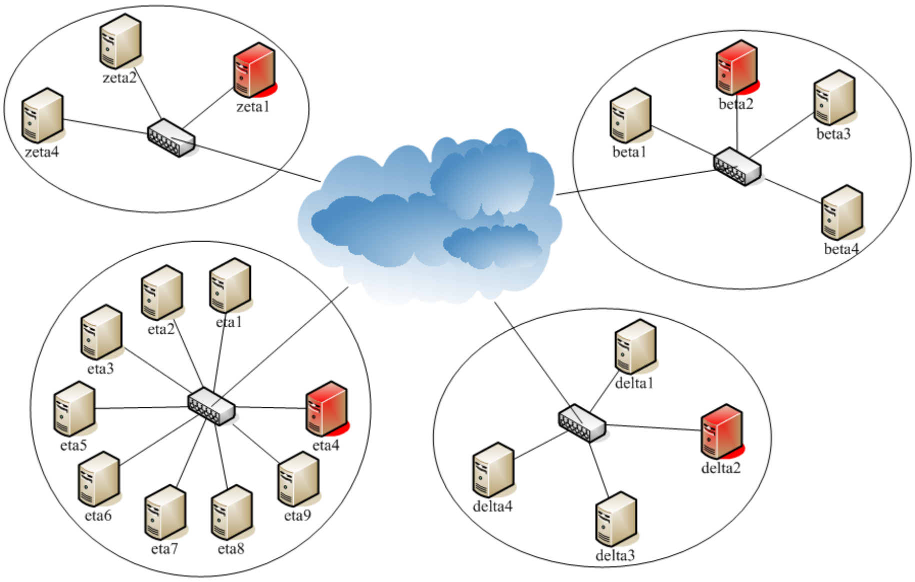

Figure 4.1: Physical deployment of grid nodes that we used for test-bed

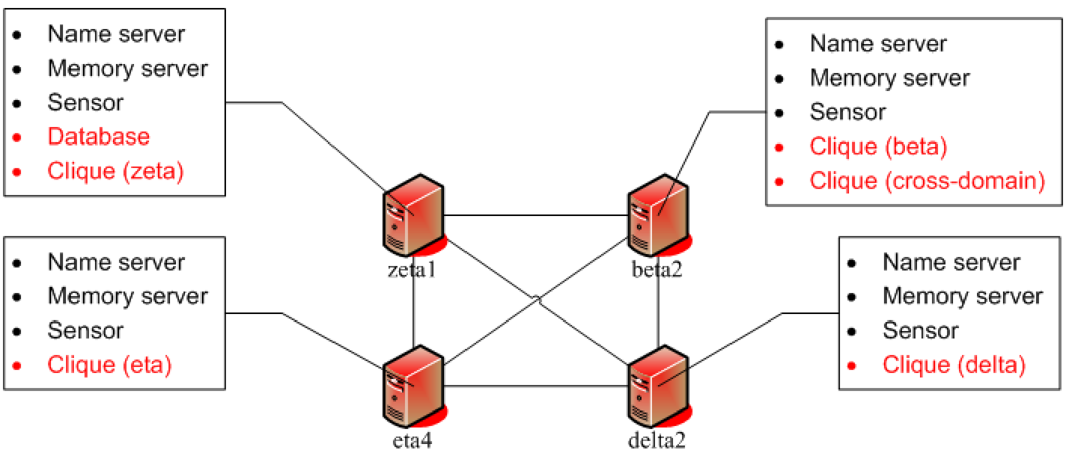

Figure 4.2: NWS services and database deployment

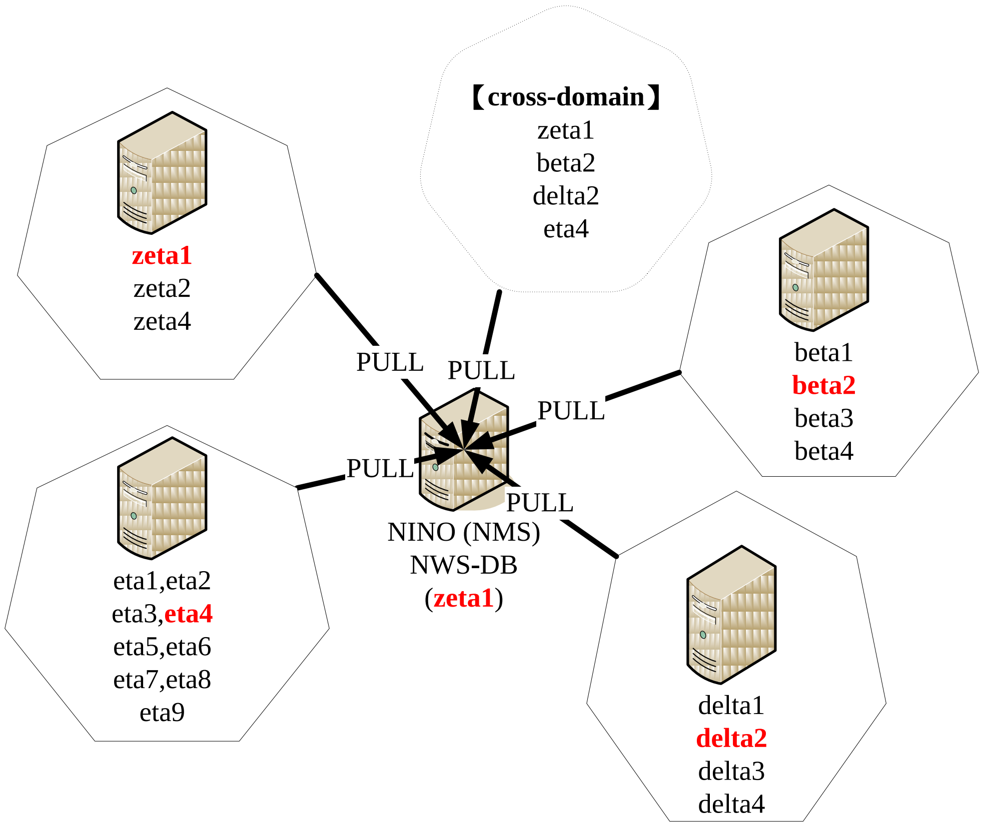

Figure 4.3: NWS cliques deployed in our Grid environments

Table 4.1 shows experimental grid nodes deployed with NWS services. All grid nodes were deployed with NWS sensor, and zeta1, beta2, delta2, and eta4 were deployed both NWS nameserver and memoryserver. Zeta1 was deployed a routine to collect (pull) all network information measured by NWS services and load these raw data into database locally. Operation system versions of these grid nodes are different, but this doesn't influence our work.

Table 4.1: Experimental Grid nodes deployed with NWS services

Clique	Name	OS	Clique	Name	OS
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 4.2: Experimental Grid groups deployed with NWS clique and their member lists, measurement period

clique name	**period**	**member**
cross-domain	30 sec	zeta1, beta2, eta4, delta2
zeta	30 sec	zeta1, zeta3, zeta4
beta	30 sec	beta1, beta2, beta3, beta4
delta	30 sec	delta1, delta2, delta3, delta4
eta	30 sec	eta1, eta2, eta3, eta4, eta5, eta6, eta7, eta8, eta9

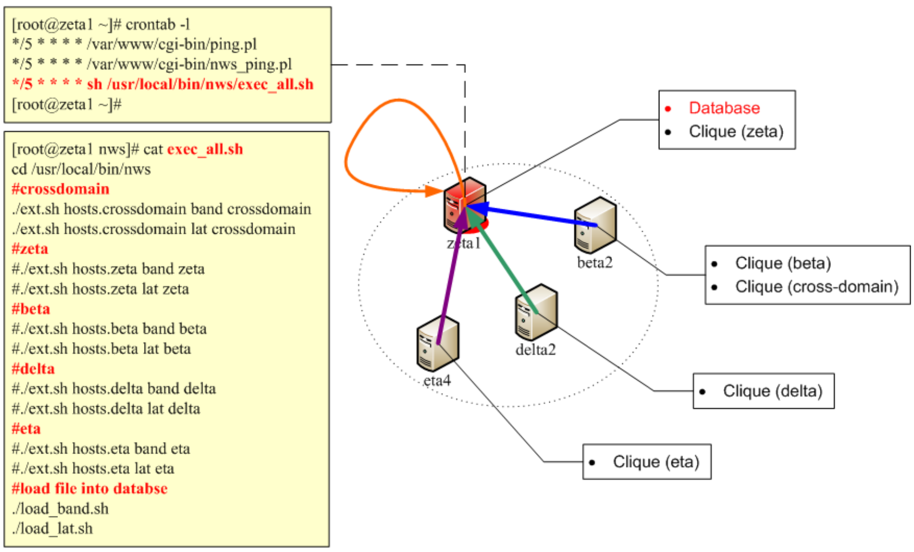

Figure 4.4: A pull-based model to collect network information measured by NWS services

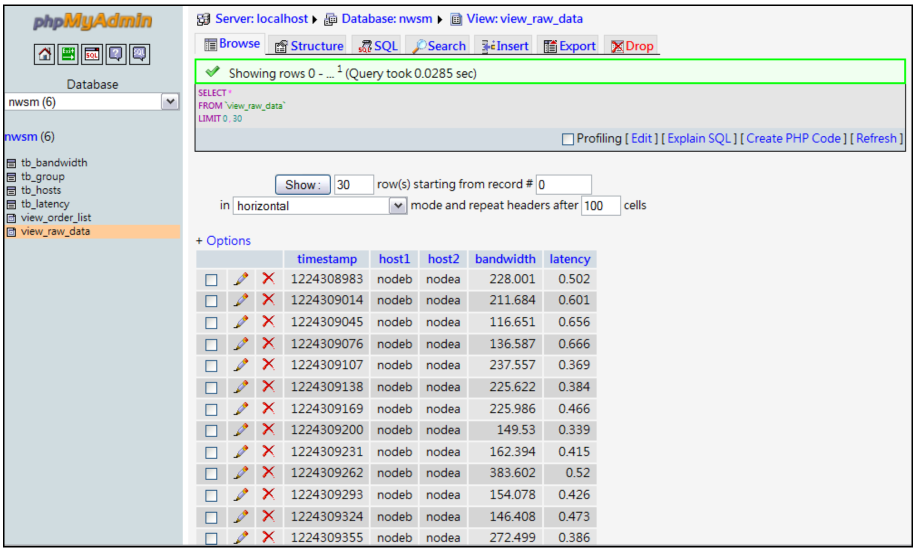

Figure 4.5: Network information from NWS measurements, both bandwidth and latency, were loaded into MySQL Database

In this thesis, we have chose 4 grid nodes as "Header", which is called "border" in 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. Hence, we have to ensure that the NWS services deployment we performed is applicable.

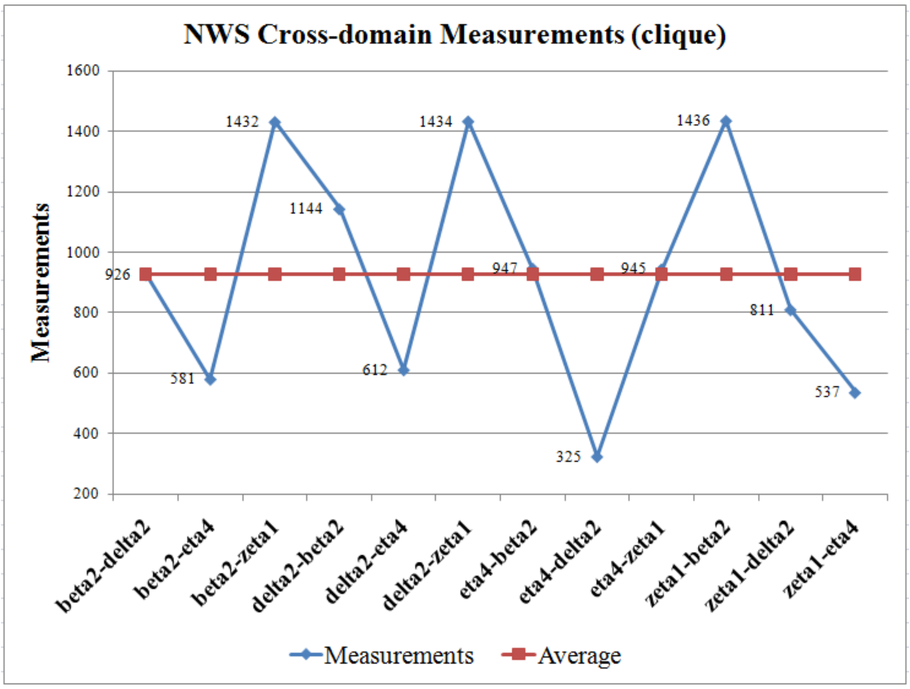

Figure 4.6: The measurements of cross-domain NWS clique

As shown in Figure 4.6, we could easily found that the measurements of NWS clique may be uneven. For example, eta4-delta2 has minimum measurements, 325, while zeta1-beta2 has maximum measurements, 1436. Uneven measurements may influence accuracy of our model while evaluating QoS with statistical approaches.

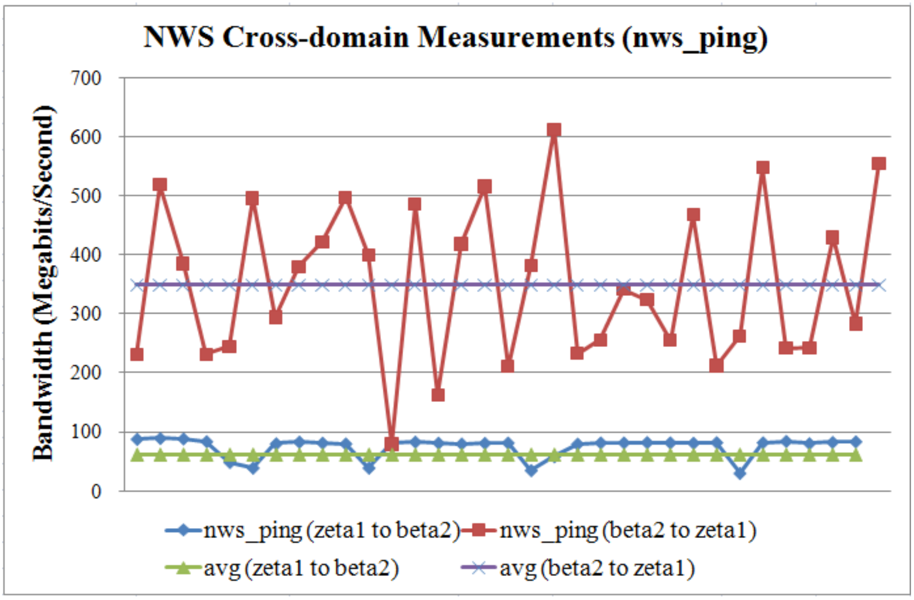

Figure 4.7: NWS services collision test

The NWS services have the ability to avoid collision which may cause inaccuracy of measurement, and this advantage is restricted in the same nameserver. In our test-bed, we found that collision influence accuracy frequently. Figure 4.7 is our collision test for NWS services. We could found that network performance has a great variation due to collision of NWS measurement.

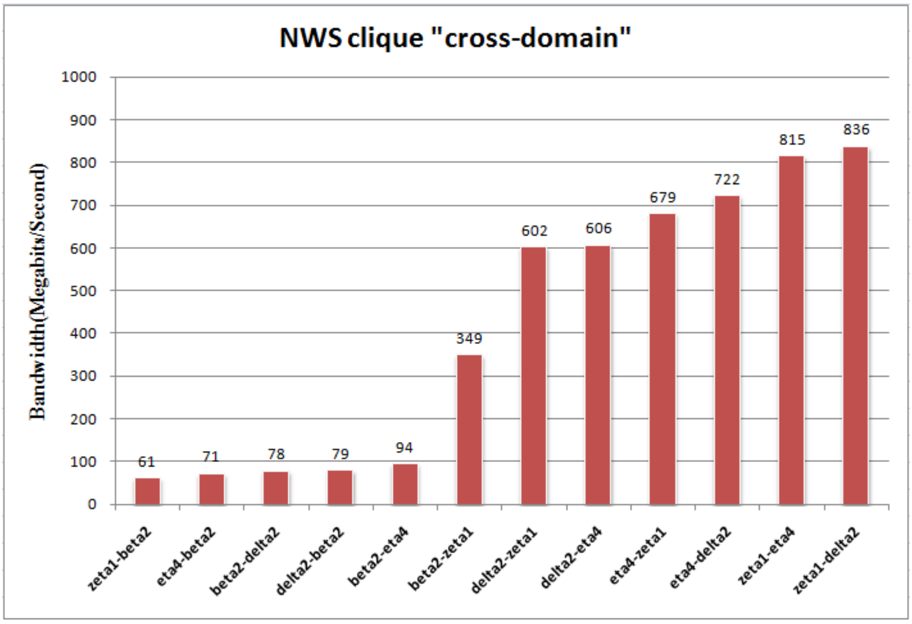

Figure 4.8: NWS measurements as our basis for QoS evaluation

Figure 4.8 has shown NWS measurements of our test-bed. Although QoS evaluation model we adopted in this thesis could not absolutely predict real performance for real-time tasks execution. We still could pick out best selection of resources by means of QoS evaluation model.

To verify usability of this QoS evaluation approach, we have also performed a simple experiment of file transmission. And the result is identical to our predication using QoS evaluation model as we mentioned in Figure 3.9.

Chapter 5

Conclusions

In this thesis, we use Domain-based Network Information Model for experiments, but it's not a proper model for dynamic grid environments. If any grid nodes that occur hardware failure or just have been reassigned to another IP, we have to manually reconstruct NWS cliques. This has already mentioned as drawback of NWS [7]. In large scale grid environments, it's a complicated task to manage these cliques and hosts' relations. Our future work will be adopting Dynamic Domain-based Network Information Model for next deployment so as to reduce overheads come from complicated management tasks.

And in this thesis, we have simplified NWS deployment with a standard procedure for managing grid nodes semi-automatically. To guarantee a certain degree of QoS, we propose a heuristic method to predict QoS from diverse grid environments for Download-oriented tasks. According to our experimental results, we 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, we have some suggestions about NWS services deployment:

- Only assign a single clique for every 10 grid nodes.
- Perform QoS evaluation for each inter-domain grid nodes first.
- To ensure certain 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.

We are planning to refer some approaches proposed by other researcher [3] to reduce measurements in the near future. And this evaluation approach should be adjusted to meet the requirements of other 3 task types, i.e., Upload-oriented, Computational and Hybrid. We'll make a study of these kinds of tasks before long.

Bibliography

[1] A taxonomy and survey of grid resource management systems for distributed computing. (2002). Softw. Pract. Exper., 32(2), 135-164.

[2] A, S., P, D., N, H., Junwei, Subhash, & R, G. (2006). Performance prediction and its use in parallel and distributed computing systems. Future Generation Computer Systems, 22(7), 745-754. doi: 10.1016/j.future.2006.02.008.

[3] Chung, W., & Chang, R. (2009). A new mechanism for resource monitoring in Grid computing. Future Generation Computer Systems, 25(1), 1-7. DOI: 10.1016/j.future.2008.04.008.

[4] Ganglia. Ganglia. Retrieved from http://ganglia.info/.

[5] Kofler, M. (2006). MySQL 5 徹底研究. 博碩.

[6] Krefting, D., Vossberg, M., & Tolxdorff, T. (2008). Simplified Grid Implementation of Medical Image Processing Algorithms using a Workflow Managment System. Presented at the MICCAI-Grid Workshop, New York. Retrieved from http://www.i3s.unice.fr/~johan/MICCAI-Grid/website.html.

[7] Legrand, A., & Quinson, M. (2004). Automatic deployment of the Network Weather Service using the Effective Network View. In Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International.

[8] Lu, J. Perl 教學. Retrieved from http://web.nchu.edu.tw/~jlu/cyut/perl.shtml.

[9] Network Weather Service. Network Weather Service. Retrieved from http://nws.cs.ucsb.edu/ewiki/.

[10] NINO - Network Manager. . Retrieved from http://nino.sourceforge.net/nino/index.html.

[11] QUE, W., ZHANG, G., & WEI, Z. Model for IP Network Synthetical Performance Evaluation. Computer Engineering, 34(8).

[12] Vazhkudai, S., Schopf, J., & Foster, I. (2002). Predicting the performance of wide area data transfers. In Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM (pp. 34-43). Presented at the Parallel and Distributed Processing Symposium., Proceedings International, IPDPS 2002, Abstracts and CD-ROM. doi: 10.1109/IPDPS.2002.1015510.

[13] Wolski, R., Spring, N., & Hayes, J. (1999). The network weather service: A distributed resource performance forecasting service for metacomputing. Future Generation Computer Systems, 15(5-6), 757-768.

[14] Yang, C. T., Chen, C. H., Yang, M. F., & Chiang, W. C. (2008). MIFAS: Medical Image File Accessing System in Co-allocation Data Grids. In IEEE Asia-Pacific Services Computing Conference, 2008. APSCC'08 (pp. 769-774).

[15] Yang, C., & Chen, S. (2008). A Multi-site Resource Allocation Strategy in Computational Grids. 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.

[16] Yang, C., Chen, S., & Chen, T. (2007). A Grid Resource Broker with Network Bandwidth-Aware Job Scheduling for Computational Grids. 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.

[17] Yang, C., Chen, T., & Tung, H. (2007). A Dynamic Domain-Based Network Information Model for Computational Grids. 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.

[18] Yang, C., Shih, P., Chen, S., & Shih, W. (2005). An Efficient Network Information Model Using NWS for Grid Computing Environments. In Grid and Cooperative Computing - GCC 2005 (pp. 287-299). Retrieved July 29, 2009, from http://dx.doi.org/10.1007/11590354_40.

[19] Yang, C., Shih, P., Lin, C., & Chen, S. (2007). A resource broker with an efficient network information model on grid environments. The Journal of Supercomputing, 40(3), 249-267. doi: 10.1007/s11227-006-0025-0.

[20] Yang, C., Yang, M., & Chiang, W. (2008). Implementation of a Cyber Transformer for Parallel Download in Co-Allocation Data Grid Environments. 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.

[21] Yinzhe Yu, Irene Cheng, & Basu, A. (2003). Optimal adaptive bandwidth monitoring for QoS based retrieval. Multimedia, IEEE Transactions on, 5(3), 466-472. doi: 10.1109/TMM.2003.814725.

[22] Zhili Cheng, Zhihui Du, & Suihui Zhu. (2007). A Service Level QoS Mechanism and Algorithm for Data Distribution and Backup in an Grid Based Astronomy Data Management System. 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.

[23] 鳥哥. (2007). 鳥哥的 Linux 私房菜基礎學習篇, 2/e. 上奇科技.

[24] 鳥哥. (2007). 鳥哥的 Linux 私房菜伺服器架設篇, 2/e. 上奇科技.

[25] 歐萊禮. (2007). SNMP 網管實務, 2/e.