Architecture for Grid Application Performance Monitoring System(GAPerMS) On The OGSA-based

Moon-Hee Lim*, Tai-Kyoung Kim*, Ok-Hwan Byeon**, Tai-Myoung Chung*
Internet Management Technology Lab.
Dept. of Information Communication
SungKyunKwan University*
Korea Institute of Science and Technology Information**
Email: {mhlim, tkkim, tmchung}@rtlab.skku.ac.kr*
Ohbyeon@garam.kreonet.re.kr**
http://imtl.skku.ac.kr/

Abstract

Grids, Metacomputing and Peer-to-Peer computing deal with utilizing available resources on the Internet (such as supercomputers, computing clusters, PCs, storage systems, data resources) to carry out various interesting tasks (such as searching for Information technique, high energy physics(HEP), extraterrestrial intelligence, bioengineering research, etc.). Combining globally distributed computers and information sources into a universal source of computing power and information, Grid Application typically need to share the resources available in the Grid. In order to solve Grand Challenge Problem in Grid Environment, we must establish grid infra structure, and perform development of grid middleware, grid application and study of a portal and development. In this paper, first of all, we present performance factors creating the grid applications. And with combining by a feedback in grid application running performance information and for specification to follow requirements for the grid network construction that the most suitable can support this application, in grid environment on the OGSA-based effectively it is purpose to design the system that can monitor and manage applications.

Keywords : Application, End-to-End, Grid, GRAM, Monitoring, OGSA. Performance, Service

Contact Person : Prof. Tai M. Chung (tmchung@rtlab.skku.ac.kr)

Introduction

- Sharing that Grid environments are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources,
- Required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering.
- *Open Grid Services Architecture*(OGSA) : alignment of Grid and Web services technologies an
- Efforts within the Grid and high performance computing communities to improve end-to-end network performance for applications
- Introduce GAPerMS for Grid Application Performance Monitoring

APNOM 2003

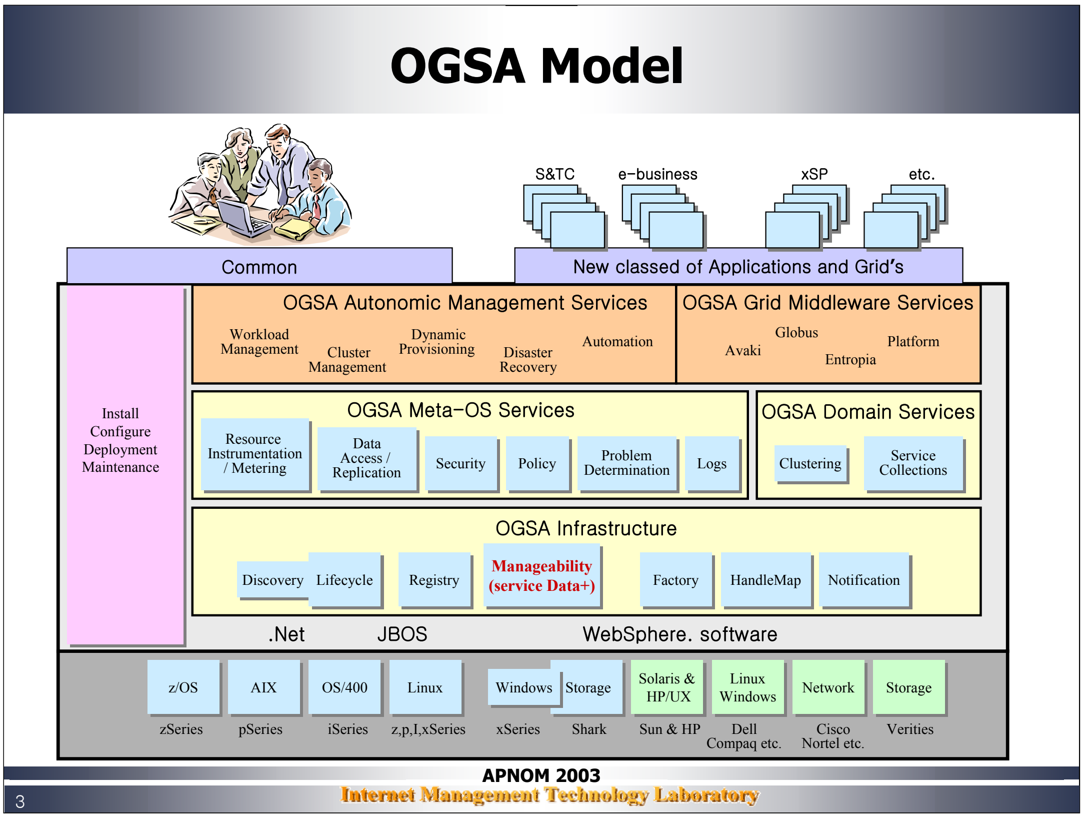
2 Internet Management Technology Laboratory

1. Introduction

Grid technologies support the sharing that Grid environments are concerned with is not primarily file exchange but rather direct access to computers, software, data, and other resources, as is required by a range of collaborative problem-solving and resource-brokering strategies emerging in industry, science, and engineering. This sharing is, necessarily, highly controlled, with resource providers and consumers defining clearly and carefully just what is shared, who is allowed to share, and the conditions under which sharing occurs. Recently, Grid technologies can be aligned with Web services technologies[1] to capitalize on desirable Web services properties, such as service description and discovery; automatic generation of client and server code from service descriptions; binding of service descriptions to interoperable network protocols; compatibility with emerging higher-level open standards, services and tools; and broad commercial support. It is called this alignment of Grid and Web services technologies an *Open Grid Services Architecture*(OGSA)[2].

Grid is composed of the network that grid infrastructure is composed of grid middleware and grid toolkit and application on a large scale. Construction and an operation of grid network include assistance about grid application and a network technology support, and assistance can hold computing dispersed regionally, shares of tip equipments and variance cooperation between users. There are considerable efforts within the Grid and high performance computing communities to improve end-to-end network performance for applications that require substantial amounts of network bandwidth. Recent experience[3] has demonstrated that actual aggregate TCP throughput realized by high performance applications is persistently much less than the end-to-end structural and load characteristics of the network would indicate is available.

In this paper, First, we will analyze End-to-End system and network performance measuring factors of the application level that it was generated of applications executed in a grid environment. For this, We introduce GAPerMS for Grid Application Performance Monitoring. We present an approach to predicting application performance and enabling the detection of unexpected execution behavior, typically caused by unanticipated load on shared grid resources.



2. OGSA Model

2.1 Requirement

The Open Grid Services Architecture(OGSA) has been proposed as an enabling infrastructure for systems and applications that require the integration and management of services within distributed, heterogeneous, dynamic "Virtual Organizations(VO)"[4]. Whether confined to a single enterprise or extending to encompass external resource sharing and service provider relationships, service integration, management in these contexts can be technically challenging because of the need to achieve various end-to-end qualities of services and applications when running on top of different native platforms. Building on Web services and Grid technologies, OGSA proposes to define a core Grid service semantics and an integrated set of service definitions that address critical application and system management concerns.

While the OGSA vision is broad, work to date has focused on the definition of a small set of core semantic elements. Specifically, the Grid service specification[5] being developed within the Open Grid Services Infrastructure(OGSI) working group of the Global Grid Forum defines, in terms of Web Services Description Language (WSDL) interfaces and associated conventions, the mechanisms that any OGSA-compliant service must use to describe and discover service attributes, create service instances, manage service lifetime, and subscribe to and deliver notifications. Characteristics of OGSA are as follows

•Distributed Resource Management across heterogeneous platforms

•Seamless QoS delivery

•Common Base for Autonomic Management Solutions (OGSA provides an open, integrating infrastructure; Grid computing then addresses issues relating to accessing and sharing the infrastructure, while autonomic functions make it possible to manage the infrastructure and thus create self-configuring, self-optimizing systems.

•Common infrastructure building blocks to avoid "stovepipe solution towers"

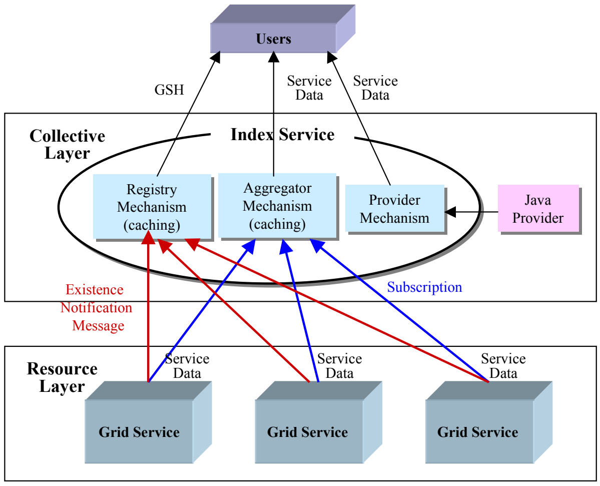
•Open and Published Interfaces

•Industry- standard integration technologies: Web Services, SOAP, XML, etc.

•Seamless integration with existing IT resources

Extension of GRID & GIIS : Grid Index Service

- A generic framework for aggregation of Service data
- A specific instance of an Index Service representing multiple resources (services and applications)



APNOM 2003

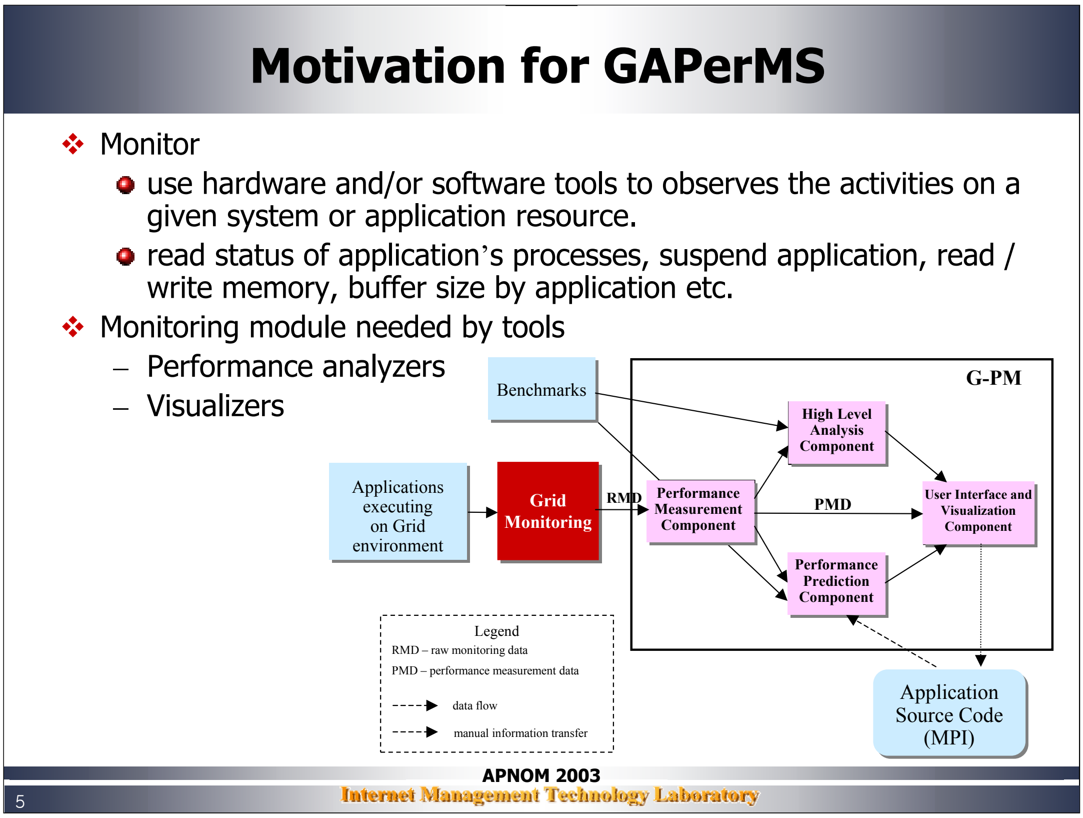
4 Internet Management Technology Laboratory

2.2 Extension of GRID & GIIS : Grid Index Service

We describe a generic framework for aggregation of Service Data and a specific instance of an Index Service representing multiple grid resources. Grid environment's Index Service is the Grid Service which provides information about Grid resources. As modular Java component framework for OGSA, service developers can use to implement various information management solutions for GT3 (Globus Toolkit 3)-compatible OGSA Services and Service Data. A Grid service instance maintains a set of service data element(SDE). It is declared via an extended XSD element declaration, placed in a WSDL portType. And It includes basic introspection information, interface-specific data, and application state.

Discovery often requires instance-specific, perhaps dynamic information. Therefore Service Data offers a general solution. Every service must support some common service data, and may support any additional service data desired. Such Service Data supports service model with properties; fine-grained view of resource functionality and data scoped by service instance, supports information domain-dependent state; service discovery or applications and services monitoring information, and stateful properties of services and applications. Also it supports infrastructure state information; introspection for client tooling and parameters of generic features

Index Service included in Collective Lager combines Service Data Provider, Service Data Aggregation and Registry Components. It creates a dynamic data-generating and indexing node, similar in concept to Globus Toolkit 2 (GT2) MDS GRIS & GIIS. Service Data Provider Components provide a standard mechanism for dynamic generation of service data via external programs. Here, External provider programs can be the core providers that are part of GT3 or be user-created, custom providers. Service Data Aggregation Components provide a reusable mechanism for handling subscription, notification, and updating of. Service Data from both locally executing information providers and other OGSA service instances can be aggregated into any given services or applications.Finally, Registry Components maintain a set of available peer Grid Service Handles(GSH)[6]. It provides soft-state cataloging of a set of Grid Services



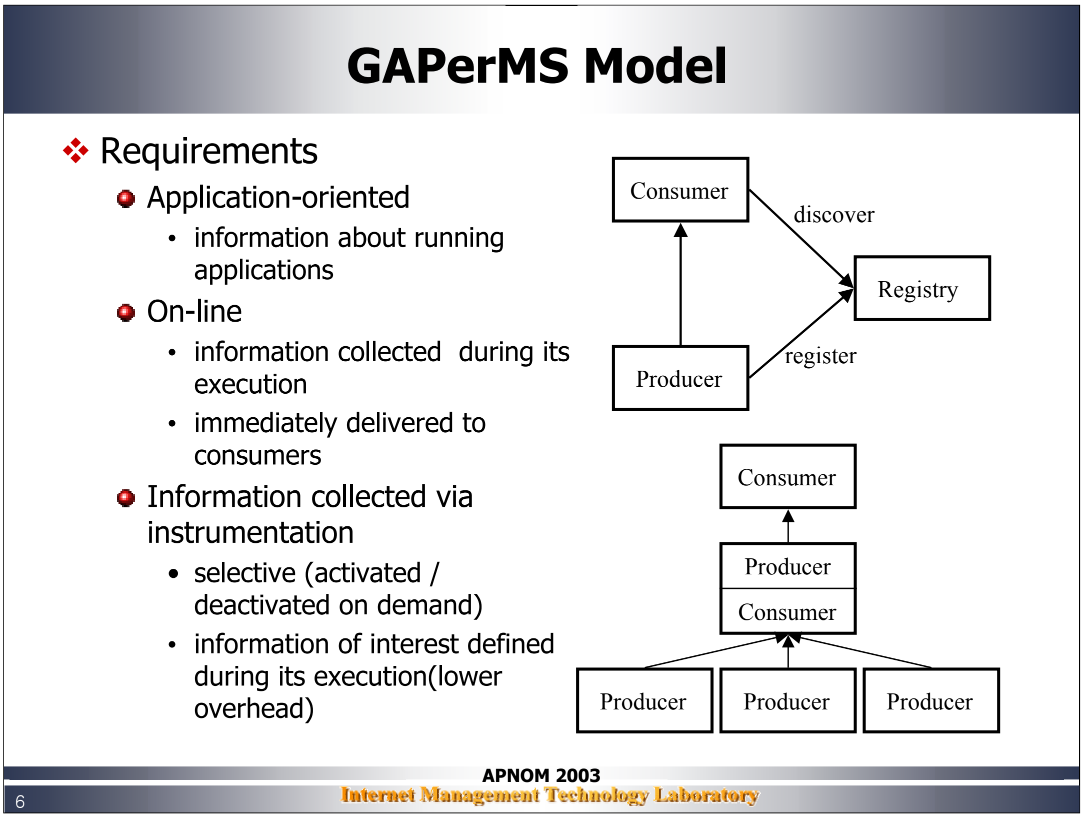
3. Application Monitoring : Motivation for GAPerMS

There are considerable efforts within the Grid and high performance computing communities to improve end-to-end network performance for applications that require substantial amounts of network bandwidth. Recent experience [7] demonstrates that actual aggregate TCP throughput realized by high performance applications is persistently much less than the end-to-end structural and load characteristics of the network would indicate is available [8].

How do we monitor and manage collections of services and applications executing performance information? Traditional performance monitoring and prediction techniques often render performance models that are specific to a single architecture or a static set of resources. However, Grid environments consist of a collection of dynamic, heterogeneous resources. Our approach seeks to separate the influences of the runtime system from the behavior intrinsic to the applications executing on Grid environment.

Application Performance Monitors have many possible usages; efficient data gathering and stori ng; take over some of the local monitor's tasks; be used to dynamically load monitoring extension s; even more for multithreaded applications. And They perform analysis performance, detection fault, identifying bottleneck, tuning performance, prediction performance, scheduling "the best resources".

Efficient data gathering produce data much more frequent than retrieval. Application performanc e data first stored locally, in the context of application processes. And performance data initially p reprocessed.



4. Designed Architecture for GAPerMS

4.1 Requirement

We design an innovative Grid Information and Application Performance Monitoring system which is based on a powerful data model, is compatible with the Grid Monitoring Architecture (GMA) of the Global Grid Forum (GGF) and as it is being implemented and will deploy. We have designed and implemented a system with a number of components following the GMA [9].

Registry supports information publication and discovery. Producer makes performance data available (performance event source). Consumer receives performance data(performance event sink). Producers of information announce themselves to a registry, Consumers (wanting information) consult the registry to find suitable resources of information, and having done so connect directly to the Consumer either to transfer data for a single request or to have date streamed to them.

An application runs on a set of resource(processors, host, network, storage or I/O etc.), each of which has certain capabilities (maximum floating point rate or available bandwidth). The application is executed with certain problem parameters, and it achieves some measurable and desired performance during its execution.

GAPerMS model is based on requirements, that is, application-oriented, on-line and information collected via instrumentation. Application-oriented exposes information about running grid applications. On-line requirement implies that information collected during application execution is immediately delivered to consumers on the on-line. And ‘information collected via instrumentation’ requirement exposes information activated or deactivated on demand and information of interest defined during application execution.

Currently GMA uses servlet technology. Each distributable component is a servlet. Each servlet receives http requests and makes an XML coded response. Code has been written for each servlet to deal with the http and XML and offer a simple client API. The API code has been constructed in Java, C, C++, Perl, and Python to suit all users.

Performance Factors for GAPerMS

Classification	MDS Naming	Description
Network Performance Information	Mds-Net-TX	Total Traffic Capacity
	Mds-Net-RX	Total Receiving Capacity
	Mds-Net-Bandwidth	Network Bandwidth
	Mds-Net-Delay	Network Average Delay
	Mds-Net-Jitter	Network Jitter
	Mds-Net-StartTime	Session Start Time
	Mds-Net-EndTime	Session End Time
	Mds-Net-BufSize	Session Buffer Size
	Mds-Net-WndwSize	Window Size
	Mds-Net-SessionTX	Session Transfer Packet Capacity
	Mds-Net-SessionRX	Session Transfer Packet Capacity
	Mds-Net-PktRoss	Packet loss or Error Capacity
	Mds-Net-Request	MDS Query Request Time
	Mds-Net-Reply	MDS Query Reply Time
	Mds-Net-RN	Router Hop Count
System Performance Information	Mds-Sys-OS	Operating System Information
	Mds-Sys-Disk	Disk Quantity Consumed
	Mds-Sys-Memory	Memory Quantity Consumed
	Mds-Sys-CPU	Average CPU Transaction Capacity
	Mds-Sys–Uptime	Application Transaction Capacity
	Mds-Sys-BufSize	Default Buffer Size
	Mds-Sys-SockBufSize	Buffer Size by Sock

APNOM 2003

7 Internet Management Technology Laboratory

4.2 Performance Factors for GAPerMS

State characteristic information and network information of current available resources are required in systems to stand in order to use these applications. Generally, applications have the characteristic that is a characteristic and the execution dependence enemy that execution is independent. The following shows classification about important property to almost break out in all dispersed parallel applications.

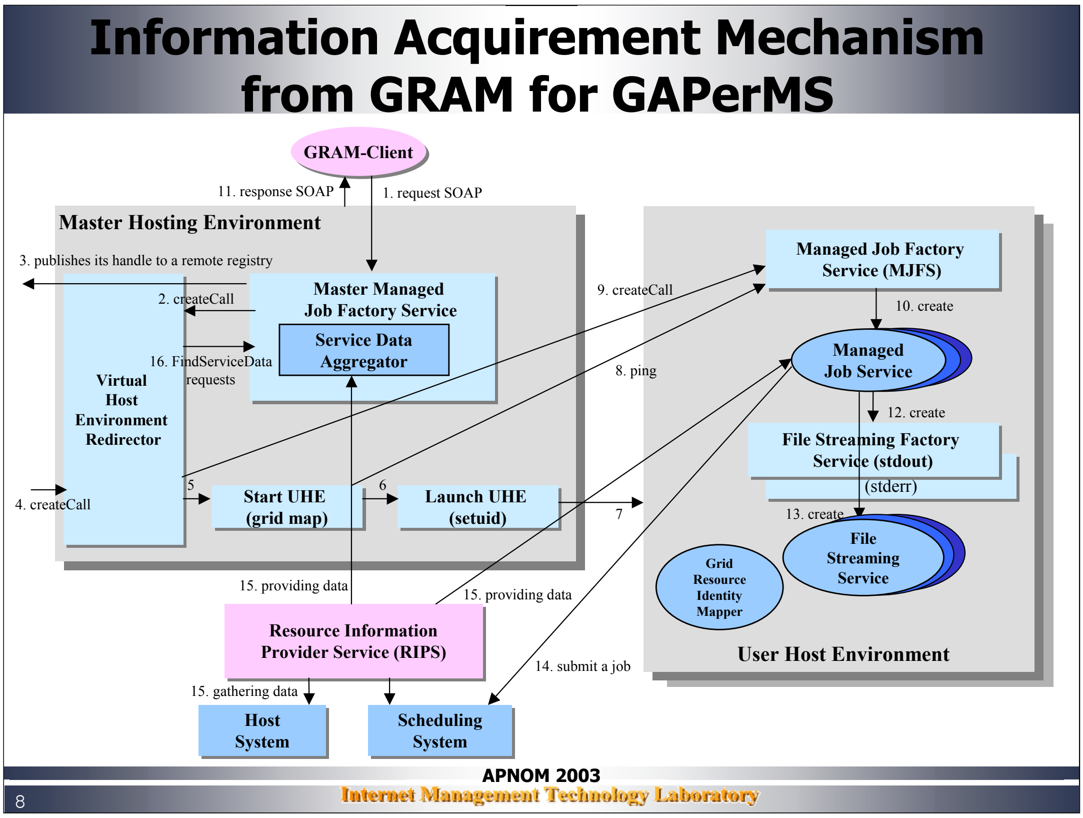
•Task-specific implementation characteristics: including computational paradigm, number and size of data structures, application's data communication patterns, memory requirements, CPU utilization, etc.

•Inter-task communication characteristics: including data format for individual execution tasks, pipeline size, communication regularity and frequency, etc.

•Application execution information: including I/O requirements, TCP buffer [10], computation, unifying data structures, iteration patterns [11], etc.

It is thing this which generalizes application-centered performance requirements and resources capacity necessary in an important element to do that so that an enormous grid application is executed in grid environment effectively [12]. Also, it is an important element to look up a general mechanism to include application - specific information in order to generalize application-centered scheduling study. As for the grid application, infinitesimal of error and reliability must be ensured when that plays delay time along sharing of tip equipment and data transmission and a process [13]. Grid network supports high-speed data networking through large-scale substitution width and requires direct mass data transmission between peer-to-peer[14]. Performance metric is necessary in order to measure this execution state. Generally, there will be multiple ways to measure a given metric.

The above performance metric partitioned off the information that a collection was possible by network performance information and system performance information. It is to have applied a naming rule of the time when it is stored a 'Name' item in MDS that each performance elements support actual grid information service. It provides only a system management system or memory information, a network address, interface name and information of the maximum data transmission unit degree used with information service information offer server of the existing Globus [15].



4.3 Information Acquirement Mechanism from GRAM for GAPerMS

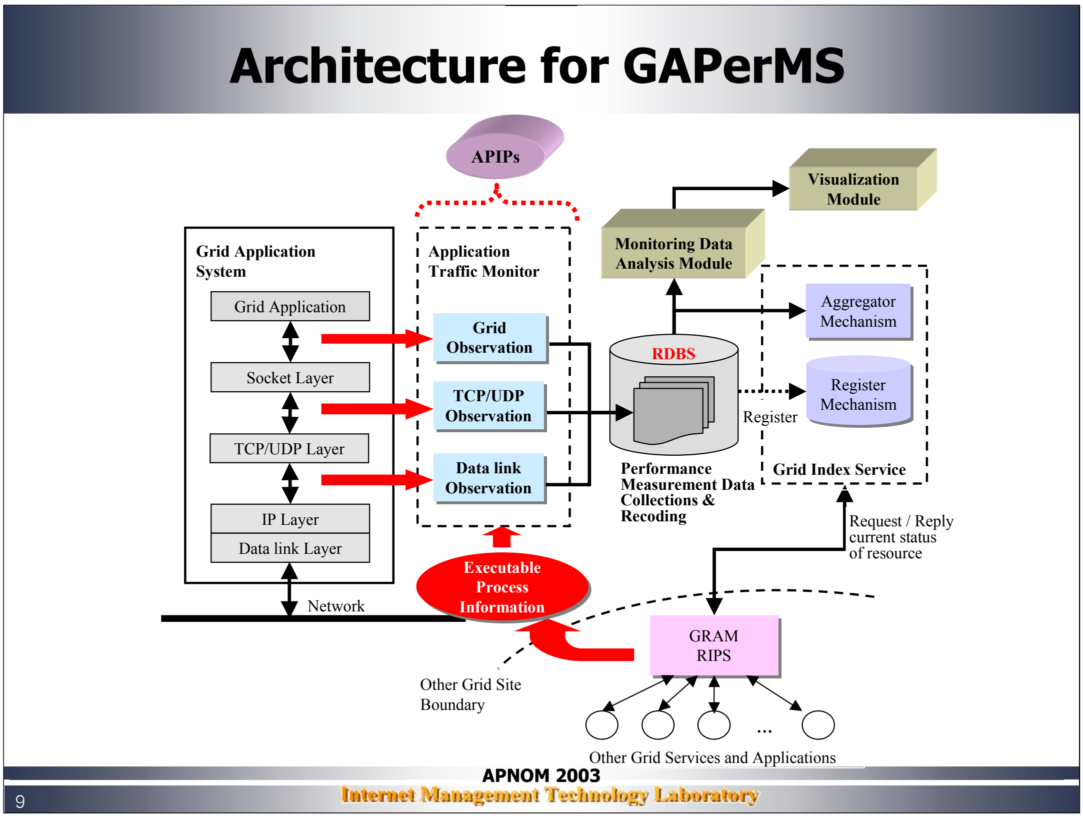
Resource Specification Language (RSL-2) is used to communicate requirements on the new Globus Toolkit 3(GT3). A set of WSDL/OGSI client interfaces allows programs to be started on remote resources, despite local heterogeneity. Much of the power of GRAM(Grid Resource Allocation Manage) is in the RSL. This is presented as XML schema defined language for specifying job requests.

Managed Job Service(MJS) translates this common language into scheduler specific language. MJS on the GRAM defines an OGSI/WSDL interface for submitting, monitoring and controlling a job. MJS uses the File Stream Factory Service to manage the job's stdout and stderr file streaming. Also, It exposes the stdout and stderr File Stream Factory Grid Service Handles(GSH) in Service Data Element. It is used by GRAM-Client. GRAM-Clients is anticipated the community will contribute robust clients.

Master Managed Job Factory Service(MMJFS) is the Redirect Provider. MMJFS expect RSL-2 for its createService input parameters, there the RSL-2describes a program(here, Grid application and services) to executed. It is returning ManagedJobService ports. Therefore, ManagedJobPort can start remote job .

As File Stream Factory Service, CreateService prepare to stream job's stdout or stderr to a destination URL. It expects RSL-2 for its CreateService input parameters, where the RSL-2 describes a destination for the file, and the offset at which to begin streaming. Output is GSH to a FileStreaming service.

In the above figure, numbered flows expose the process GRAM client request the service, and it is dealt with. Resource Information Provider Service(RIPS) is a specialized notification service. It maintains job information from the scheduler. Scheduler information provider is essentially the GR2(Globus Toolkit 2) queue script used by the GRAM-reporter, but it outputs XML instead of LDIF. The MJS instances will subscribe for RIPS for notification on job state changes. Interactions with file system and scheduler are done by MJS calling scheduler perl module. This allows the JM host to be different from the scheduler host.



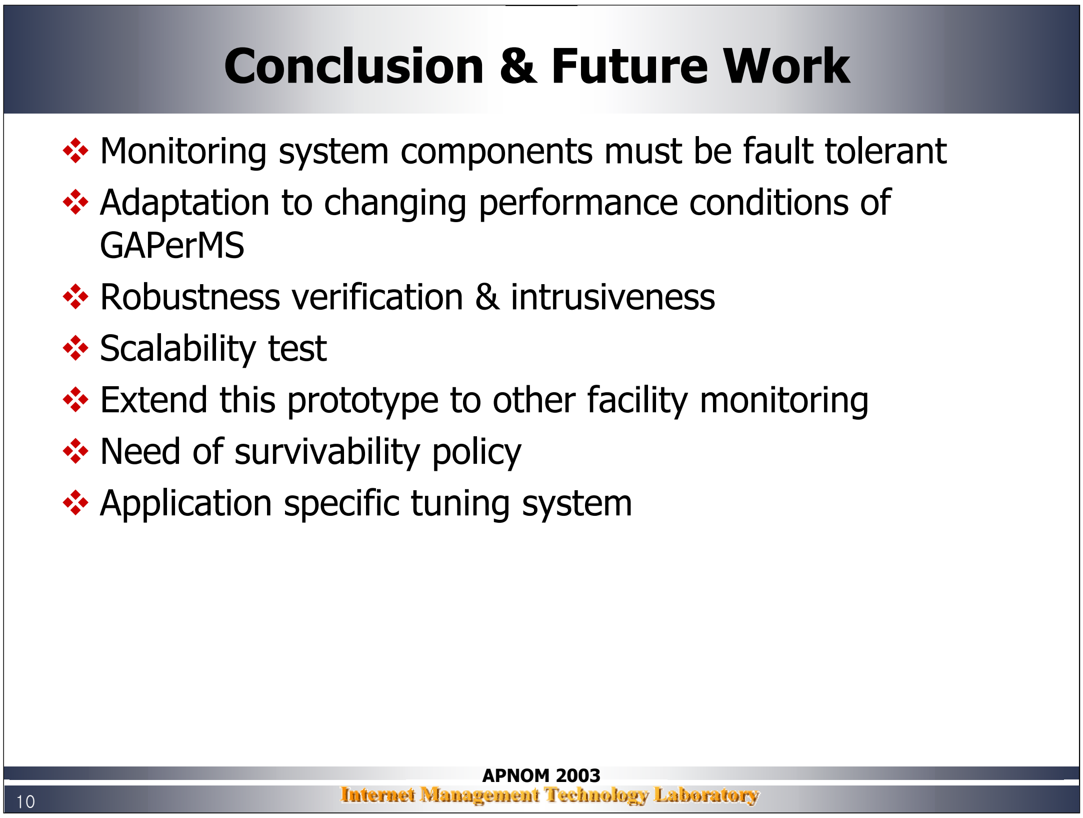
4.4 Architecture for GAPerMS

We have chosen a relational implementation that we have been able to apply both to information and to monitoring. The system creates th impression that you have one RDBMS per Virtual Organization(VO) as far as inserting and retrieving information is concerned.

The proposed system, GAPerMS is a system to collect necessary information in order to compose a network to support an application and service executed in grid environment effectively. Creating a contract in the GAPerMS consists of information that provided given resources information, certain capabilities (disk capacity, CPU capacity, buffer size, window size etc.), measurable network performance factors(traffic capacity, bandwidth, latency, transfer rate, loss or error capacity etc.) for an grid application. These factors was expressed with MDS Naming before 2 pages. GAPerMS measures performance information of the application system that Globus was installed in, and we refer to these performance providers 'APIP(Application Performance Information Provider)' here.

We will develop a Application Performance Information Provider(APIP) on the network link that can forecast network information used in the future. It is stored and delivery in these network performance information providers by condition and forecast information with GRIS of the system that adjoined most. We added a system and a network information provider in an information provider element to provide in the existing GRIS in this system. It is to have indicated all process that gets these grid applications that increased and information about used resources performance condition to support service.

The Resource performance information provider (APIP) that can measure each performance is implemented in the system kernel space that an individual application is practiced. As the APIPs, it is located in TCP layer in order to measure traffic generated in an actual application execution process. Such as platform type of system, host OS specification, system load, file system back related to system operation is performance information provided in Grid Index Service of Globus basically. It is transmitted to MDS event information measured in APIP are stored in performance RDB with raw data, and to support information service. When it checks own state periodically, and a failure or performance drops, please cope with role of performance RDB according to the event information that an application system was generated with an application being located in a executed on the system quickly and do.



5. Conclusion and Future work

In this paper we presented performance factors that it was generated when it was run grid application. Demonstrated the use of these with our management infrastructure. Besides, we measured these application performance elements exactly and expected these and maximized efficiency of grid application behavior and composed grid application execution environment effective in and designed an End-to-End application performance management system to manage these. The grid supports multi-platform in order to overcome that the operating system is different from a platform of heterogeneous resources scattering to a lot of area. Also, resources management service and an application must provide the flexibility that can respond to what time or a dynamic environment for this because possibility stopped at any time exists among a great number of peers connected to grid network. Because possibility stopped at any time exists among a great number of peer connected to grid network, resources management service and an application must provide the flexibility that can respond to at any time a dynamic environment for this. A numerous peers cannot but take notice in the rod balancing which considered a case usable at the same time when they considered the high-performance that is a characteristic of grid. Grid grows larger more and more and it is enormous, and measures about a situation for grid performance to be able to become a decrease in a network problem or a problem about rod balancing because that will have produced various peer are necessary.

Measurement and simulation will do the performance elements that they defined in this paper in test-bed and an actual grid network, and we add an actual program of APIP to a system too, and we can construct a correct performance analysis base. An important consideration matter must not have an influence on the traffic which measurement and traffic used by a performance analysis were generated by an application. To provide robustness, measuring systems are remotely started and killed for each measurement. Also, Monitoring system components, APIPs must be fault tolerant, adapt to changing performance conditions of services and application, and can extend this prototype to other facility monitoring.In addition to, they need of survivability policy. For the future, it is important study to develop an application specific tuning system in order we manages a highly efficient grid effectively, and to be able to acquire the best results in application execution.

References

[1] Foster, I., Kesselman, C., Nick, J. and Tuecke, S. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration, Globus Project, 2002.

[2] "Open Grid Services Architecture: A Roadmap", GGF, 2002.

[3] Harimath Sivakumar, Stuart Bailey, Robert L. Grossman., "PSockets : The Case for Application-level Network Striping for Data Intensive Applications using High Speed Wide Area Networks", SC2000:High-Performance Network and Computing Conference, Dallas, TX, 11/00

[4] B. Lowekamp, B. Tierney, L. Cottrell, R. Hughes-Jones, T. Kielmann, M. Swany, "A Hierarchy of Network Measurements for Grid Applications and Services.", GWD-I. Network Measurements Working Group DRAFT,

[5] Tuecke, S., Czajkowski, K., Foster, I., Frey, J., Graham, S. and Kesselman, C. Grid Service Specification, 2002 .

[6] S. Tuecke, K. Czaijkowski, I.Foster, J. Frey, S. Graham, C. Kesselman, T. Maquire, T. Sandholm, D. Snelling, P. Vanderbilt, "Open Grid Services Infrastructure(OGSI): Version 1.0 (draft)", GWD-R, draft-ggf-ogsi-gridservice-29,http://www.ggf.org/ogsi-wg, April 5, 2003.

[7] J. Lee, D. Gunter, B. Tierney, W. Allock, J. Bester, J. Bresnahna, S. Tecke, "Applied Techniques for high Bandwidth Data Transfers across Wide Area Networks." Dec. 2000, LBNL-46269.

[8] Thnmas J. hacker, Brian D. Athey, "The End-to-End Performance Effects of Parallel TCP Sockets on a Lossy Wide-Area Network".

[9] Brian Tierney, Ruth Aydt, Dan Gunterm, Warren Smith, Valerie Taylor, Rich Wolski and Martin Swany, "A Grid Monitoring Architecture", GGF, 2001.

[10] TCP Extension for High Performance, RFC 1323

[11] F. Vraalsen, Ruth A. Aydt, Celso L. Mendes, and Daniel A. Reed, "Performance Contracts: Predicting and Monitoring Grid Application Behavior.", Proceedings of the 2nd International Workshop on Grid Computing/LNCS(GRID 2001), Volume 2242, pp. 154-165, 2001.

[12] Francine Berman, Richard Wolski, "Scheduling From the Perspective of the Application." Copyright 1996 IEEE. Published in the Proceeding of the High Performance Distributed Computing Conference, 1996.

[13] B. Lowekamp, B. Tierney, "Network Metrics for Grid Applications and Services", Network Measurements Working Group, 2002.

[14] Oram A., "Peer-to-Peer: harnessing the Power of Disruptive Technologies", O'Reilly Press, 2001.

[15] The Globus Project Web Page. http://www.globus.org