FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY UNIVERSITY OF MALAYA

2 maroon

GRID APPLICATION

Submitted By

TAN KIEN PENG
WEK020254

Under the Supervision of

MR ANG TAN FONG

Moderator

MR LIEW CHEE SUN

SESSION 2004/2005

ABSTRACT

After Grid computing introduced since 1995, many research done for the Grid computing. Most of the grid computing research project are focusing on the science. Some vendors such as IBM, Oracle and others start introduce Grid computing to the industry.

Basically, my project focused on the data grid. A web based FSKTM data portal is developed for users to access into the data portal. The lecturers and students can gather the resources from the databases. They can find out the related resources from distributed and multiple databases which are located in the heterogeneous platforms. Besides that, lecturers can analyze data from the databases. They can analyze student's result and distribution of students.

The middleware that used to develop FSKTM data portal is OGSI.NET. It supports the development of web services. Visual C# is used to develop this project.

Besides that, I hope this project to help lecturers and students in gathering related resources in the databases.

ACKNOWLEDGEMENT

I would like to take this opportunity to dedicate my sincere appreciation to those people who had helped me throughout this project. Their precious advises and sound understanding throughout the project is deeply appreciated.

First, I would like to express my gratitude to my respected supervisor, Mr. Ang Tan Fong for his support, valuable guidance, encouragement and constructive comments during this project.

Secondly, special thanks to my moderator, Mr. Liew Chee Sun who has contributed suggestions and ideas in this project.

I would like to take this opportunity to thank my group member, Mr. Yap Chin Wai for their generous sharing of knowledge and support. These support contribute to the completion of my project.

TABLE OF CONTENT

ABSTRACT i
ACKNOWLEDHEENT ii
TABLE OF CONTENT iii
LIST OF FIGURES v
LIST OF TABLES vi

CHAPTER 1 INTRODUCTION 1
1.1 INTRODUCTION TO GRID COMPUTING 1
1.2 PROECT OBJECTIVE 5
1.3 PROJECT SCOPE 6
1.4 REPORT ORGANIZATION 6

CHAPTER 2 LITERATURE REVIEW 7
2.1 INTRODUCTION TO OPEN GRID SERVICES ARCHITECTURE (OGSA) 7
2.2 GRID ARCHITECTURE 17
2.3 GRID APPLICATION 24
2.4 CASE STUDY 27
2.5 COMPUTING TECHNOLOGY 38
2.6 OPERATING SYSTEM 43
2.7 PROGRAMMING LANGUAGE 47
2.8 INTRODUCTION TO DISTRIBUTED DATABASE 51
2.9 INTRODUCTION TO REPLICATION 56
2.10 DATABASE MANAGEMENT SYSTEM 57
2.11 DATA ACCESS TECHNOLOGIES 61
2.12 INTRODUCTION TO THE WEB SERVICES 63
2.13 PROTOCOL 68
2.14 MIDDLEWARE 71

CHAPTER 3 METHODOLOGY 83
3.1 SOFTWARE DEVELOPMENT 83
3.2 FACT FINDING 84
3.3 ITERATION AND INCREMENTATION MODEL 86
3.4 TESTING 88

CHAPTER 4 ANALYSIS 89
4.1 INTRODUCTION TO .NET FRAMEWORK 89
4.2 PROGRAMMING LANGUAGES 91
4.3 OPERATING SYSTEM PLATFORM 93
4.4 DATABASE MANAGEMENT SYSTEM 94
4.5 FUNCTIONAL REQUIREMENT 95
4.6 NON-FUNCTIONAL REQUIREMENT 96

CHAPTER 5 SYSTEM DESIGN 98
5.1 SYSTEM FUNCTIONALITY DESIGN 99

5.2 DATABASE DESIGN 103

CHAPTER 6 IMPLEMENTATION 105
6.1 INTRODUCTION 105
6.2 DEVELOPMENT ENVIRONMENT 105
6.3 APPROACHES TO THE DEVELOPMENT OF THE SYSTEM 107
6.4 CODING IMPLEMENTATION 110

CHAPTER 7 TESTING 116
7.1 INTRODUCTION 116
7.2 TYPE OF TESTING 118

CHAPTER 8 SYSTEM EVALUTION 120
8.1 INTRODUCTION 120
8.2 PROBLEM ENCOUNTER AND SOLUTION TO OVERCOME THEM 120
8.3 SYSTEM STRENGTH 124
8.4 SYSTEM LIMITATIONS 125
8.5 FUTURE ENHANCEMENTS 125

REFERENCES 127

APPENDIX

LIST OF FIGURES

FIGURE 2.1 GRID ARCHITECTURE 17
FIGURE 2.2 SCIENCE PORTAL ARCHITECTURE 25
FIGURE 2.3 SEARCH MODULE 27
FIGURE 2.4 ADVANCED SEARCH 28
FIGURE 2.5 SEARCH RESULT 28
FIGURE 2.6 MAIN PAGE OF WIKI 31
FIGURE 2.7 USER ENTER KEYWORD IN THE SEARCH FIELD 32
FIGURE 2.8 SEARCH RESULT WHICH SEARCH BY KEYWORD 32
FIGURE 2.9 MAIN PAGE OF ASTROGRID 33
FIGURE 2.10 MAINPAGE OF NERC DATA GRID 34
FIGURE 2.11 SEARCH RESULT THAT DISPLAY 35
FIGURE 2.12 SEARCH ENGINE 35
FIGURE 2.13 ADVANCED SEARCH 36
FIGURE 2.14 CONFIGURE THE DISPLAY FORMAT OF SEARCH RESULTS 36
FIGURE 2.15 HIGH-PERFORMANCE COMPUTING HIERARCHY 40
FIGURE 2.16 TYPES OF PEER-TO-PEER COMPUTING 42
FIGURE 2.17 RELATIONSHIPS NETWEEN PROVIDER ENTITY AND REQUESTOR ENTITY 65
FIGURE 2.18 OVERVIEW OF GRAM 73
FIGURE 2.19 OVERVIEW OF MDS 75
FIGURE 2.20 OVERVIEW OF GLOBUS TOOLKIT 76
FIGURE 2.21 OVERVIEW OF OGSI.NETCOMPONENTS 80
FIGURE 2.22 ARCHITECTURE OF MS.NETGRID 82
FIGURE 3.1 SOFTWARE DEVELOPMENT 84
FIGURE 3.2 ITERATION AND UNCREMENTATION 87
FIGURE 5.1 CONTEXT DIAGRAM 99
FIGURE 5.2 DATA FLOW DIAGRAM 102
FIGURE 5.3 ER DIAGRAM 104

LIST OF TABLES

TABLE 5.1 SYMBOLS OF DATA FLOW DIAGRAM 101
TABLE 5.2 TABLE OF mcat_logical_file 103
TABLE 5.3 TABLE OF indexing 104
TABLE 6.1 SOFTWARE SPECIFICATIONS FOR IMPLEMENTATION PROCESS 106

CHAPTER 1 INTRODUCTION

1.1 INTRODUCTION TO GRID COMPUTING

Grid computing concept was explored since 1995. I-WAY experiment was the first large-scale grid experiment. It consisted of high-end resources at 17 sites across North America. There are over 60 applications which ran on the I-WAY.

Grid computing is a distributed computing which allows resource-sharing among heterogeneous platforms. Grid computing uses open standards to coordinate the resources which located at different places and belonging to different administrative domain over a network. Resources are the core component in the grid computing. It allows users to solve computational problems by sharing resources from other computers in the grid computing.

Besides that, multi-user environment is provided in the grid computing. Authentication techniques are performed before users need to log on to the grid system. This is to ensure the security of grid computing.

There are some tasks that can apply in the grid computing:

- If the machine which an application normally runs is busy due to peak in activity, the application could be run on other computer which is in grid computing. There are two prerequisites that required fulfilling. First, the application must be executable remotely and without undue overhead. Second, any special hardware, software or resource requirements that need by application must meet with the selected machine.
- An application could be split into several of independent sub tasks and could run into different machine. The application need to use algorithm to partitioned it

into independently running part. After all running part finish their application, the result will be collected.

1.1.1 VIRTUAL RESOURCES AND VIRTUAL ORGANIZATIONS

The Grid computing system seems like a large computer system because it enabled very heterogeneous system to work together. A large virtual resources is created when every machines in the grid computing system share resources. Sharing is not only including file but also hardware, software, services, licenses and others.

To organize users dynamically, virtual organizations are created based on the different policy requirement. These virtual organizations can share their resources collectively as a larger grid. Users can be members of several real or virtual organizations. The Grid enforces security rules among them and implemented policies which can resolve priorities for both resources and users.

1.1.2 TYPE OF GRID

There are three types of grid which is computational grid, data grid and scavenging. These types of grid are dividing according application perspective. The boundaries between these grid types are not very clear. Grid often is a combination of two or three of grid types.

1.1.2.1 COMPUTATION GRID

Computation grid is focus on setting aside resources which are specifically for compute power. Most of the machines in the computation grid are high-performance servers. The benefits of computation grid are faster, more efficient processing of computer-intensive jobs while utilizing existing resources.

1.1.2.2 SCAVENGING

Scavenging grid most common used with a huge numbers of desktop machines. Machines that in scavenging machines are share their CPU cycles and other resources. The owners of the desktop machines are given control of when their resources are available participating in the grid.

1.1.2.3 DATA GRID

Data grid is providing a way for users to access to data across multiple organizations. Users can gain the required data but they do not need to concern where the actual location of the data stored. Hence, data grid enables users and applications to manage and efficiently use database information from distributed location. The data grid works on high speed disk caches and on tertiary storage and cache management provides the basic building blocks. Relevant metadata standards and metadata-driven retrieval mechanisms are developed within the library community.

1.1.2.3.1 DATA GRID DESIGN

To operate wide area, multi-institutional and heterogeneous environments, there are several principles that need to follow in the data grid design.

- Mechanism neutrality

  The data grid architecture that designed should be as independent as possible. This feature is important for the low-level mechanisms used to store data, store metadata, transfer data and so forth.

- Policy neutrality

  While data movement and replica cataloging are provided as basic operations, replication policies are implemented via higher-level procedures for which defaults are provided but that can easily be substituted with application-specific code.

- Computability with grid infrastructure

The data grid architecture should be structured to ensure data grid tools are compatible with lower-level Grid mechanisms. The implementation of strategies that integrate can be simplified.

- Uniformity of information infrastructure

In the underlying grid, uniform and convenient access to information about resource structure and state is emphasized.

1.1.2.3.2 FUNDAMENTAL SERVICES

There are two types of fundamental services which provided by data grid. The fundamental services are data access and metadata access. Accessing, managing and initiating third-party transfers of data stored in storage systems are the mechanisms which are provided by data access. Data access should support for remote requests to read and/or write named file instances and determine file instances attributes such as size. The metadata access service provides the mechanisms for accessing and managing information about data stored in storage systems.

Besides that, the data grid can provide other services for the users. The services are including authentication and authorization, resource reservation and co-allocation mechanisms and others.

1.1.3 GRID TOPOLOGY

Grid network can designed from sharing a few computers to enterprise network. Grid topology is another method to differentiate the grid. There are three types of grid topology which are Intra-grid, Extra-grid and Inter-grid.

1.1.3.1 INTRA-GRID

Intra-grid is created within an enterprise and inter-departmental girds. This is the smallest and simplest grid topology. The middleware which used must have a better understanding of resources allocation. It is caused by processor-sharing relationships which will increase additional complexity.

1.1.3.2 EXTRA-GRID

Extra-grid is a grid network which connected between geographically distributed sites within or between enterprise organizations. There are two important concepts in extra-grid which are geographic distribution and inter-enterprise. Some features are become critical to the middleware to ensure the processing or data can be shared. There are authentication, policy management and security.

1.1.3.3 INTER-GRID

Inter-grid is the largest size in the grid topology. It embodies two of the primary vision of grid computing. There are utility computing infrastructure and grid services or grid services providers.

1.2 PROJECT OBJECTIVE

The following points are the main objectives that need to be achieved in this project.

- Develop a Data Grid Portal for Faculty of Computer Science and Information Technology, University of Malaya
- This application can run through Linux and Windows platforms
- Users can access this application through a standard web browser.
- Provide a reliable search engine for users to gather related resources

1.3 PROJECT SCOPE

To make the project success, there are several scopes that need to define.

- Provide a standard and user-friendly interface that allows user easy to gather related resources
- Allow users download resources from multiple sites
- Allow users share resources through uploading resources to databases

1.4 REPORT ORGANIZATION

There are five chapters included in this project. Those chapters are organized as follow:

Chapter 1 provides an introduction to the grid computing, project objectives, project scopes and report organization.

Chapter 2 covers the information of grid architecture, grid application, case studies for the existing grid applications, platform for operating system, programming tools, grid middleware, database management system and data access technologies.

Chapter 3 explains the types of fact finding that used and methodology methods that used in this project

Chapter 4 discusses the programming tool, platforms and database management system that used to develop this project. Besides that, this chapter also discusses functional requirements and non-functional requirements of this project.

Chapter 5 explains the design of the data grid portal and its databases design. Context diagram, data flow diagram, data dictionary and entity-relationship diagram are explained in this chapter.

CHAPTER 2 LITERATURE REVIEW

2.1 INTRODUCTION TO OPEN GRID SERVICES ARCHITETURE (OGSA)

A modern computing environment which links up diverse components can be viewed as a single virtual system. To discover access, allocate, monitor, account and manage the virtual system, standardization is needed. This also caused by the components which provided by different vendors and/or operated by different organization. A service-oriented architecture, a set of core capabilities and behaviors already defined in the Open Grid Services Architecture (OGSA). This architecture is defining the standardization that used in Grid computing.

2.1.1 REQUIREMENT OF GRID COMPUTING

There are several requirements which need to be concerned. This is important when the developers develop grid application. The requirements will be defined as below.

2.1.1.1 DYNAMIC AND HETEROGENEOUS ENVIRONMENT SUPPORT

The grid computing needs to support heterogeneous environment. Variety of operating systems, hosting environments, devices and services made up the heterogeneous environment. To reduce the complexity of administrating heterogeneous system, the interoperability between these diverse, heterogeneous and distributed resources and services must be enable.

Some requirements that need to obtain for heterogeneous systems support. The requirements are:

- Resource virtualization

  Reduce the complexity of managing heterogeneous systems and to handle diverse resources in unified way.

- Common management capabilities

  To uniform and consistent management of resources, the administration of a heterogeneous system needs to be simplified. A minimum set of common manageability capabilities is required.

- Resource discovery and query

  Mechanisms which should handle a highly dynamic system well are required for discovering resources with desired attributes and for retrieving their properties.

- Standard protocols and schemas

  This is important for interoperability. In addition, standard protocols are also particularly important as their use can simplify the transition to using grid.

2.1.1.2 RESOURCE SHARING ACROSS ORGANIZATION

The resources in the Grids are owned and controlled by various organizations. Hence, OGSA is to support resource sharing and utilization which across administrative domains. A context which can be used to associate users, requests, resources, policies and agreements should be provided by mechanisms. There are some requirements for resource sharing.

- Global name space

  To ease data and resource access, OGSA entities should be able to able to access other OGSA entities transparently, subject to security constraints, without regard to location or replication.

- Metadata services

  This is important for finding, invoking and tracking entities. It should be possible to define views on entities that allow for access to and propagation, aggregation and management of entity metadata across administrative domains.

- Site autonomy

  Mechanisms are required for accessing resources across sites while respecting local control and policy

- Collecting and exchanging usage information across organizations

  Some mechanisms and standard schemas are performed for the purpose of accounting, billing, etc.

2.1.1.3 OPTIMIZATION

Optimization applies to both suppliers and consumers of resources and services. Flexible resource allocation policies which include advance reservation of resources with a bounded time period, the pooling of backup resources and others can improve the resource utilization.

Metering, monitoring, logging, changing resource allocation and provisioning resources are the mechanisms for monitoring resource utilization. These mechanisms should apply dynamically and on demand. Demand-side optimization must include different ways of managing various workloads. An important requirement for workload management is the ability to dynamically adjust priorities in order to meet service level objectives.

2.1.1.4 QUALITY of SERVICE (QoS) ASSURANCE

Quality of Service is an important requirement in the Grid environment. Services such as job execution and data services must provide the agreed-upon QoS. The measurable terms which can be captured in the Service Level Agreements (SLAs) are used to express the QoS. The QoS assurance requirements include:

- Service level agreement

  Service requestor and provider need to negotiate for establishing agreements. These agreements are QoS. Standards mechanisms should be provided to create and manage agreements.

- Service level attainment

  If the agreement requires attainment of service level, the resources used by the service should be adjusted so that the required QoS is maintained. Hence, some mechanisms are required to monitor services quality, estimate resource utilization and plan for and adjust resource usage.

- Migration

  It should be possible to migrate executing services or applications to adjust workloads for performance or availability.

2.1.1.5 JOB EXECUTION

Manageability for execution of user-defined tasks throughout their lifetime must be provided by OGSA. Scheduling, provisioning, job control and exception handling of jobs are the functionalities which must be supported. The requirements that needed in the job execution are:

- Support for various job types

  Various types of job such as simple jobs and complex jobs must be supported.

- Job management

  The jobs are managed during their entire lifetimes. Mechanisms are required for controlling the execution of individual job steps.

- Scheduling

  Tasks are scheduled and executed based on the information such as specified priority and current allocation of resources. Multiple schedulers are used to schedule the task across administrative domains.

- Resources provisioning

  The complicated process of resources allocation, deployment and configuration need to be automated. It must be possible to deploy the required applications and data to resources and configure them automatically, if necessary deploying and reconfiguring hosting environments such as OS and middleware to prepare the environment needed for job execution. It must be possible to provision any type of resource not just compute resources.

2.1.1.6 DATA SERVICES

More fields of science and technology are needed for the huge quantities of data to be accessed and moved in efficiently. Data sharing is one of the components that need to consider important here. Data services requirements include:

- Data access

  To abstract underlying data sources, the types of data should be accessed easy and efficiently. Besides that, the data's physical location should be independent. The access rights at different levels of granularity are controlled by mechanisms.

- Data consistency

  The consistency can be maintained when cached or replicated data is modified.

- Data persistency

  Data and its association with its metadata should be maintained for their entire lifetime. It should be possible to use multiple persistency models.

- Data integration

  The mechanisms should be provided to integrate heterogeneous, federated and distributed data. It is also required to be able to search data available in various formats in a uniform way.

- Data location management

  The required data should be made available at the requested location. Transfer, copying and caching is the selection that should be allowed by OGSA. All the selection is depend on the nature of data.

2.1.1.7 SECURITY

Through robust security protocols and the security policies that provided, the safe administration requires controlling access to services. To protect Grid systems, standard and secure mechanisms should be deployed. These mechanisms support safe resource-sharing across administrative domains. The requirements that need in security are:

- Authentication and authorization

  The identity of individuals should be authenticated before the services can be established. The authorization mechanisms should be implemented by service providers to enforce policy over how each service can be used.

Authorization should be accommodate various access control models and implementations.

- Multiple security infrastructure

  Multiple security infrastructures can integrate and interoperate with the distributed operations in Grid systems.

- Perimeter security solutions

  Applications may have to be deployed outside the client domain. Standard and secure mechanisms need to be deployed. These mechanisms enable cross-domain interaction without compromising local control of equipment for protecting domains, such as firewall policy and intrusion-detection policy.

- Isolation

  There are various kind of isolation must be ensured in the Grid systems. Examples of isolation are performance isolation, isolation of users and isolation between contents offerings within the same Grid system.

- Delegation

  Requestors have been delegate to access to services that required. To minimize the risks of misuse of delegated rights, the rights transferred through delegation are scoped only to the intended job and should have a limited lifetime.

- Security policy exchange

  Security policy information should be exchanged dynamically between service requestors and providers. A negotiation about security context can be established.

- Intrusion detection, protection and secure logging

  Intrusion detection and identification of misuses, malicious or otherwise should be monitored. The critical areas or functions should be protected by migrating attacks away from them.

2.1.1.8 ADMINISTRATIVE COST REDUCTION

Administration costs and the risk of human errors increased due to the complexity of administering of administering large-scale distributed and heterogeneous system. To automate Grid system control, policy-based management is required. The operations of policy-based management conform to the goals of the organization that operates and utilize the Grid system. The policies include monitoring and managing resources, business processes, availability, performance, security, scheduling and brokering.

The deployment, configuration and maintenance of complex systems should be facilitated and all application-related information are allowed to be specified and managed as a single logical unit. All these mechanisms are defined in the application contents management.

The administrators should recognize and cope quickly when problem occur. Hence, problem determination mechanisms are needed.

2.1.1.9 SCALABILITY

To utilize huge number of resources, the Grid system should be added new services or values. The potentially thousands of resources of varied nature need to be scaled by the management architecture.

To improve throughput of entire computational process, the parallel job execution should be adjusted and optimized. The mechanisms which performed these tasks are high-throughput computing mechanisms.

2.1.1.10 AVAILABILITY

Expensive fault-tolerant or complex cluster systems are often used to provide high availability. Since Grid systems are introduced, Grid system can be used to realize stable, highly-reliable execution environments. Policy-based autonomous control and dynamic provisioning are keys to realizing systems of high flexibility and recoverability.

To ensure Grid system can be recovered quickly and efficiently, discovery mechanisms are needed. These mechanisms can avoid the services that disrupted for a long duration. Fault management mechanisms can be required so that running jobs are not lost because of resource faults. Fault detection and diagnosis of causes or impacts on running jobs are required.

2.1.2 OGSA FRAMEWORKS

Service-oriented architecture (SOA) is the interaction between the services. The services include the individual and collective of resources. The services are built on web service standards, with semantics, additions, extensions and modifications that are relevant to Grids.

There are a few important points in the OGSA frameworks. The points are listed below:

- Composition paradigm or building block approach is a set of capabilities or functions is built or adapted as required from a minimalist set of initial capabilities. Adaptability, flexibility and robustness to change which are required in the architecture should be provided.

- The services, their interface and the semantics/behavior and interaction of these services are represented in the OGSA. The implementations of internals of the services which are driven by software architecture is outside the OGSA working group's scope.
- The architecture is not layered where the implementation of one service is built upon and can only interact with the layers upon which it is logically dependent.

The services which either single or as part of an interacting group of services in the Grid systems are loosely coupled peers. The capabilities of OGSA can be realized through implementation, composition or interaction with other services.

2.2 GRID ARCHITECTURE

Grid architecture identifies the fundamental component of grid computing. Besides that, it also specifies the purpose and function of these components. The components will follow the specification to interact with each other. Hence, grid architecture is foremost a protocol architecture. Virtual organization users should follow the protocol to establish, manage and exploit sharing relationships. Resource negotiate also include in the protocol. Figure 2.1 shows the diagram of grid architecture.

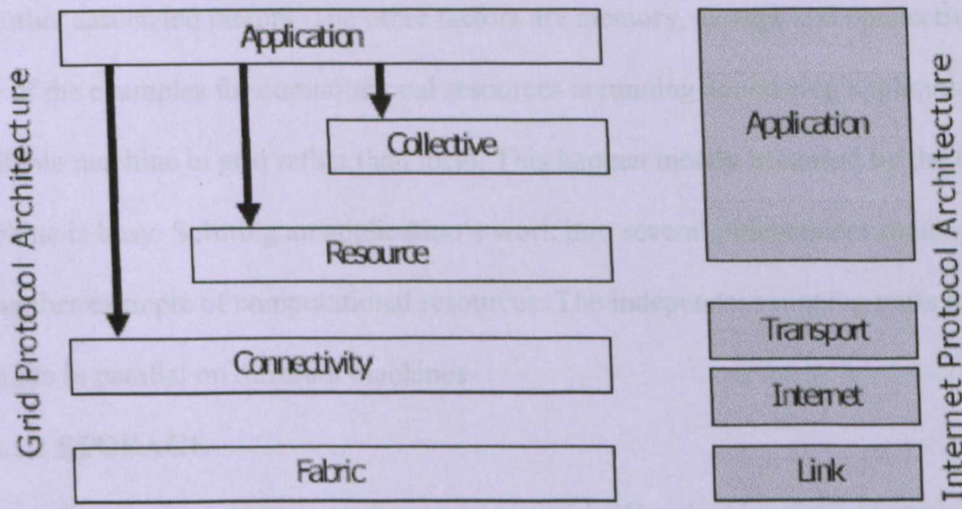


Figure 2.1 Grid Architecture

2.2.1 FABRIC

A fabric layer is the lowest layer in the grid architecture. All the resources which shared in the grid computing are provided by fabric layer. Resources may be logical entity. Enquiry mechanism is implemented in the fabric layer. It can let users to request the resources in the grid computing. The structure, state and capabilities of resources will also discovery by virtual organization users through this enquiry mechanism. Resource management is important in the fabric layer. It can provide some control of delivered of quality services.

2.2.1 TYPES OF RESOURCES

There are various types of resources which could share to Grid. The types of resources are computational, storage, communications, software and licenses, special equipment, capabilities, architecture and policies.

2.2.1.1 COMPUTATIONAL

This is the most common resource in grid computing. Computing cycles which provided by processors of the machine can vary in speed, architecture, software platform and other associated factors. The other factors are memory, storage and connectivity. One of the examples for computational resources is running an existing application on available machine in grid rather than local. This happen mostly is caused by the local machine is busy. Splitting an application's work into several independent running parts is another example of computational resources. The independent running parts can execute in parallel on different machines.

2.2.1.2 STORAGE

Storage is another type of resources in grid. The machines in grid network usually provide some storage. The purpose of providing the storage is for the use of grid. Besides that, the secondary storage also increases the capacity, performance, sharing and reliability of data. Mountable networked file systems are used by grid system. The example of mountable networked file system are Andrew File System (AFS®), Network File System (NFS), Distributed File System (DFS™), or General Parallel File System(GPFS). The differences among this type of mountable networked file systems can show through the performance, security features and reliability features. More advance file system may automatically duplicate set of data. This will increase reliability and performance of the grid system.

2.2.1.3 COMMUNICATION

Sending jobs and getting required data within a grid need a communication. The communication is important to grid system. Some jobs require large amount of data to process but they may not always reside on the machine to process the jobs. Communication in grid is not only within a grid but also can external access to internet. It is important to have redundant communication paths. It can prevent from network failures.

2.2.1.4 SOFTWARE AND LICENSES

Some software needs too high expanses to buy licenses to install in every machine in grid. To save the cost, the applications that need the software may transfer to the machine which already installed the required software. Besides that, concurrent copies of the software which are being used are kept track by licenses management software. Licenses management software prevents more than numbers from executing at any time.

2.2.1.5 SPECIAL EQUIPMENT, CAPACITIES, ARCHITECTURE AND POLICIES

Each machine in grid may have different operating systems, devices, capacities and equipment. These criteria may be a consideration for job scheduler to decide how to assign the jobs.

2.2.2 CONNECTIVITY

Communication and authentication are provided in the connectivity layers. Communication layer is responsible in data exchange between fabric layer resources. Hence, the communication requirements are include transport, routing and naming. The communication will transfer the resources to the actual location to execute applications.

Authentication protocol which used in grid computing ensures only the authorized virtual organization users log on to grid application. There is some of the characteristics of authentication solutions in grid:

- Single sign on

Users just need log on to the grid computing once. After they log on, they can access any resources which shared in the grid system. There is no further user intervention.

- Delegation

A grid application should access the resources that it requires. The resources are under user's authorization. Besides that, the application should be able to conditionally delegate a subset of its rights to another application.

- Integration with various local security solutions

Other security solutions like Kerberos and UNIX security also can integrate in the grid security. The resource providers can add on these local security solutions to authenticate users.

- User-based trust relationships

Users can use resources from multiple providers. The security system must not require each of the resource providers to cooperate or interact with each other in configuring the security environment.

2.2.3 RESOURCE

Protocols are defined for secure negotiation, initiation, monitoring, control, accounting and payment of sharing operations on individual resources. Resource layer implement these protocols to call fabric layer's functions. These functions can let users access and control local resources. Protocols in the resource layer are perform in local machine only.

There are two primary classes in the resource layer protocol. First, information protocols are used to obtain information about the structure and state of a resource. Example of information protocols is the configuration, current load and usage policy of a resource. Second, management protocols are used to negotiate access to shared resource. The negotiation includes these following fields such as resource requirements and operations to be perform. Advanced reservation and quality of service are included in the resource requirement.

2.2.4 COLLECTIVE

Unlike resource layer only interacts with single resource, collective layer contains protocols and services to capture interact across collections of resources in grid. Collective layers can implement variety of sharing behaviors without placing new requirements. The sharing behaviors which stated are listed below.

➢ Directory services

Virtual organization users can discover existing properties of the resource. Besides that, users can query resources through directory services by name, or attributes such as types, availability and load.

➢ Co-allocation, scheduling, and brokering services

Virtual organization users are allowed to request one or more resources for some specific purpose. They also can schedule task on appropriate the resources. Example for co-allocation, scheduling and brokering services include AppLeS, Condor-G, Nimrod-G, and the DRM broker.

➢ Monitoring and diagnostics services

To prevent failure, intrusion detection, overload and other, monitoring of virtual organization resources are important.

➢ Data replication services

These services are support of virtual organization storage resources. The purpose of these services is to maximize data access performance. The metrics which used to evaluate the data access performance are response time, reliability, and cost.

➢ Grid-enabled programming systems

These systems enable familiar programming models use in grid environment. These programming models use various grid services to address resource discovery, security, resource allocation and other concerns.

➢ Workload management systems and collaboration frameworks

These frameworks are also known as problem solving environments. The information which can gain from these frameworks is the description, use, and management of multi-step, asynchronous, multi-component workflows.

➢ Software discovery services

These services can discover and select the best software implementation and execution platform. The parameters that choose software implementation and execution platform are based on problem being solved. Examples for these services are NetSolve and Ninf.

- Community authorization servers

Community policies are enforced to govern resource access and generating capabilities. Community members use these policies to access the resources. These servers provide global policy enforcement service. This service build on resource information, and resource management protocols which in the Resource layer and security protocols in the Connectivity layer.

- Community accounting and payment services

These services gather resource usage information. The purpose of these services is do accounting, payment, and/or limiting of resource usage by community members.

- Collaboratory services

These services coordinated exchange of information within potentially large community members. The examples for these services are CAVERNsoft, Access Grid, and commodity groupware systems.

2.2.5 APPLICATION

User applications can execute within virtual organization environment. The user application can call or request services which define at any layers.

2.3 GRID APPLICATION

There are several grid applications that developed now. All grid application can categorize into five categories. Science portals, distributed computing, large-scale data analysis, computer-in-loop instrumentation and collaborative work are the common grid application.

2.3.1 SCIENCE PORTAL

Science portals are applications which allow application users access to the resources through a web page interface. Security in the websites is important. All users who access to the science portal should be authenticate. Besides that, science portal also help them make better decisions for scheduling jobs. When a job is submitted, a profile is created and stored for portal users. Users can use profiles to track and monitor jobs submitted and view results.

The science portal is a standard three tier model. A client browser can establish securely communications to web servers over https connection. HTTPS is a secure version of http. It can provide authentication and encrypted communication. Web server is capable of accessing various Grid services using Globus infrastructure. The science portal architecture is shown in figure 2.2.

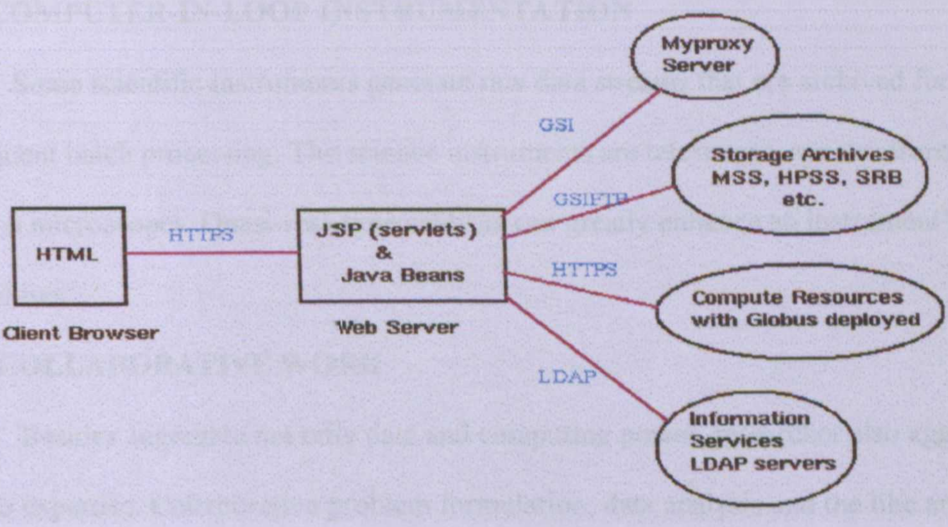

Myproxy Server
GSI
Storage Archives MSS, HPSS, SRB etc.
GSIFTP
HTML
HTTPS
JSP (servlets) & Java Beans
HTTPS
Compute Resources with Globus deployed
Client Browser
Web Server
LDAP
Information Services LDAP servers

FIGURE2.2 Science Portal Architecture

2.3.2 DISTRIBUTED COMPUTING

Various high-speed workstations and networks can be connected together to form a substantial computational resources. Besides that, some machines such as personal computer also can link together to process computational problems. This will reduce the cost of process which compare with using expensive workstations. FigthAIDS@Home is one of the examples of distributed computing. The range of problems that aggregated computing resources can tackle will be increased if network performance and grid technologies are improved.

2.3.3 LARGE-SCALE DATA ANALYSIS

Many scientific researches require analysis of large datasets to solve the scientific problems. In this application, the communities not only share computers and storage but also share analysis procedures and computational results. Distributed resources can be used efficiently because the natural parallelism inherent in many data analysis procedure.

2.3.4 COMPUTER-IN-LOOP INSTRUMENTATION

Some scientific instruments generate raw data streams that are archived for subsequent batch processing. The science instruments are telescopes, synchrotrons and electron microscopes. Quasi-real-time analysis can greatly enhance an instrument's capabilities.

2.3.5 COLLABORATIVE WORK

Besides aggregate not only data and computing power, researcher also aggregate human expertise. Collaborative problem formulation, data analysis and the like are important Grid application. Some research projects need the researchers view results and can discuss each other in real time.

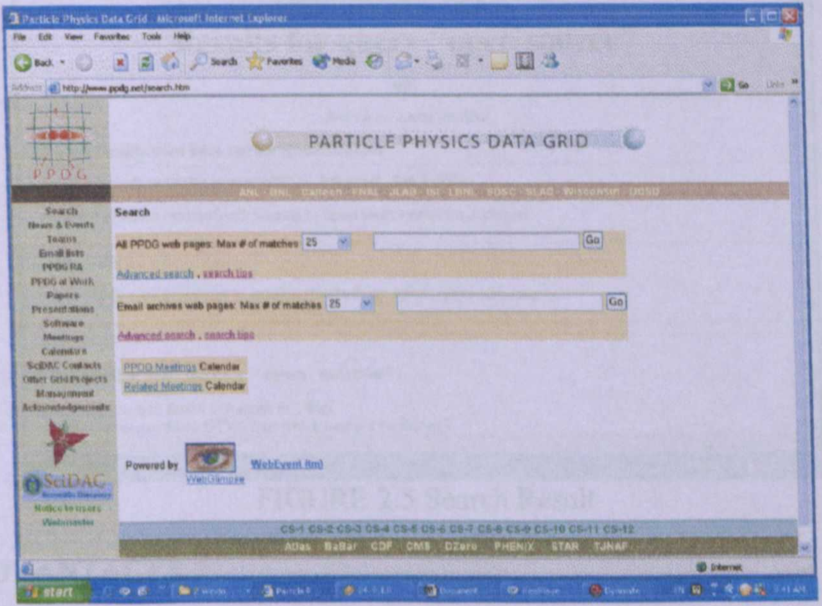
2.4 CASE STUDY

2.4.1 PARTICLE PHYSICS DATA GRID (PPDG)

2.4.1.1 OVERVIEW

The Particle Physics Data Grid (PPDG) is funded by U.S Department of Energy from the Office of Science, Office of High Energy and Nuclear Physics and Office of Advanced of Scientific Computing Research from the SciDAC program. It is a grid development project which develops and deploys Grid systems vertically integrating experiment-specific applications.

There are several modules in the PPDG websites. The modules are Search, News & Events, Teams, Papers, Presentations and others. The users can find the related data from the Search, Papers and Presentation modules. The user can search their interested data through the Search module. The search result will be display after users click the "Go" button. Figure 2.3 below shows the Search module of PPDG.

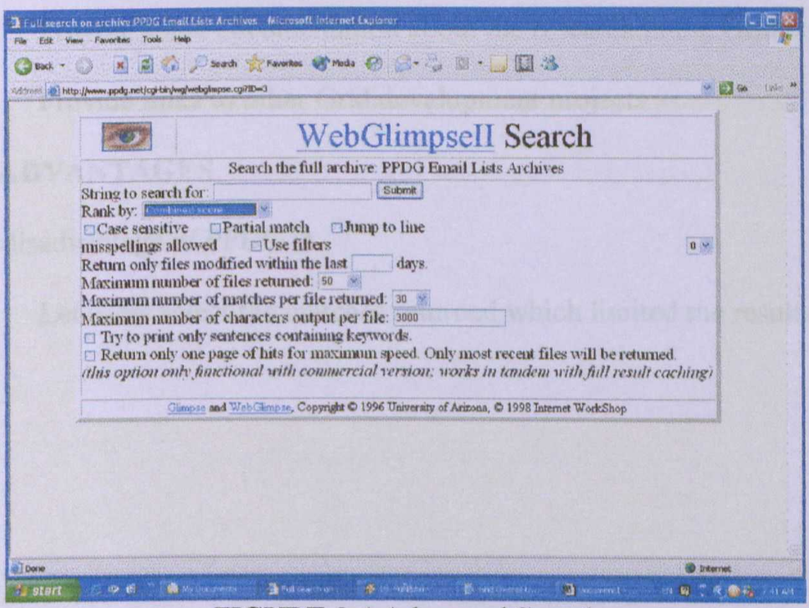

Particle Physics Data Grid - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
http://www.ppdg.net/search.htm
Go
PPDG
PARTICLE PHYSICS DATA GRID
ANL · BNL · Caltech · FNAL · JLAB · ISI · LBNL · SDSC · SLAC · Wisconsin · UCSD
Search
News & Events
Teams
Email lists
PPDG RA
PPDG at Work
Papers
Presentations
Software
Meetings
Calendars
SciDAC Contacts
Other Grid Projects
Management
Acknowledgements
SciDAC
Notice to users
Webmaster
Search
Advanced search
search tips
PPDG Meetings Calendar
Related Meetings Calendar
Powered by
WebGlimpse
WebEvent (tm)
CS-1 CS-2 CS-3 CS-4 CS-5 CS-6 CS-7 CS-8 CS-9 CS-10 CS-11 CS-12
ATlas BaBar CDF CMS DZero PHENIX STAR TJNAF
Internet
start

All PPDG web pages: Max # of matches 25 [Go]

Email archives web pages: Max # of matches 25 [Go]

FIGURE2.3 Search Module

Besides that, PPDG also provides advanced search which allow users consider more criteria. The advance search and search result are show in the figures below.



Full search on archive PPDG Email Lists Archives - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
http://www.ppdg.net/cgi-bin/wg/webglimpse.cgi?ID=3
Go Links
Done
Internet
start

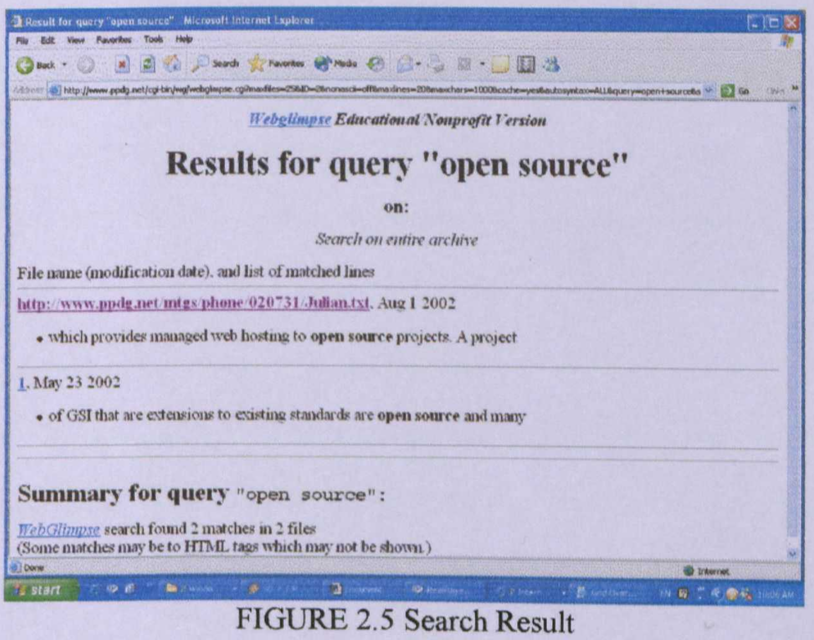
WebGlimpseII Search

Search the full archive: PPDG Email Lists Archives

String to search for: Submit
Rank by:
□Case sensitive □Partial match □Jump to line
misspellings allowed □Use filters 0
Return only files modified within the last days.
Maximum number of files returned: 50
Maximum number of matches per file returned: 30
Maximum number of characters output per file: 3000
□ Try to print only sentences containing keywords.
□ Return only one page of hits for maximum speed. Only most recent files will be returned.
(this option only functional with commercial version; works in tandem with full result caching)

Glimpse and WebGlimpse, Copyright © 1996 University of Arizona, © 1998 Internet WorkShop

FIGURE 2.4 Advanced Search



Result for query "open source" - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Go
Done
Internet
start

http://www.ppdg.net/cgi-bin/wg/webglimpse.cgi?maxfiles=25&ID=2&nonascii=off&maxlines=20&maxchars=1000&cache=yes&autosyntax=ALL&query=open+source&

Webglimpse ***Educational Nonprofit Version***

Results for query "open source"

on:

Search on entire archive

File name (modification date), and list of matched lines

http://www.ppdg.net/mtgs/phone/020731/Julian.txt, Aug 1 2002

- which provides managed web hosting to **open source** projects. A project

1, May 23 2002

- of GSI that are extensions to existing standards are **open source** and many

Summary for query "open source":

WebGlimpse search found 2 matches in 2 files
(Some matches may be to HTML tags which may not be shown.)

FIGURE 2.5 Search Result

2.4.1.2 ADVANTAGES

The advantages that can gain from the PPDG are:

- ➢ Support wildcards

- Provide some search tips to allow users more familiar with the search result
- Provide some documentation about the research in the Grid computing
- Provide links to other Grid development projects

2.4.1.3 DISADVANTAGES

The disadvantage of PPDG is:

- Let users select the matches returned which limited the results that return

2.4.2 ASTROGRID

2.4.2.1 INTRODUCTION TO ASTROGRID

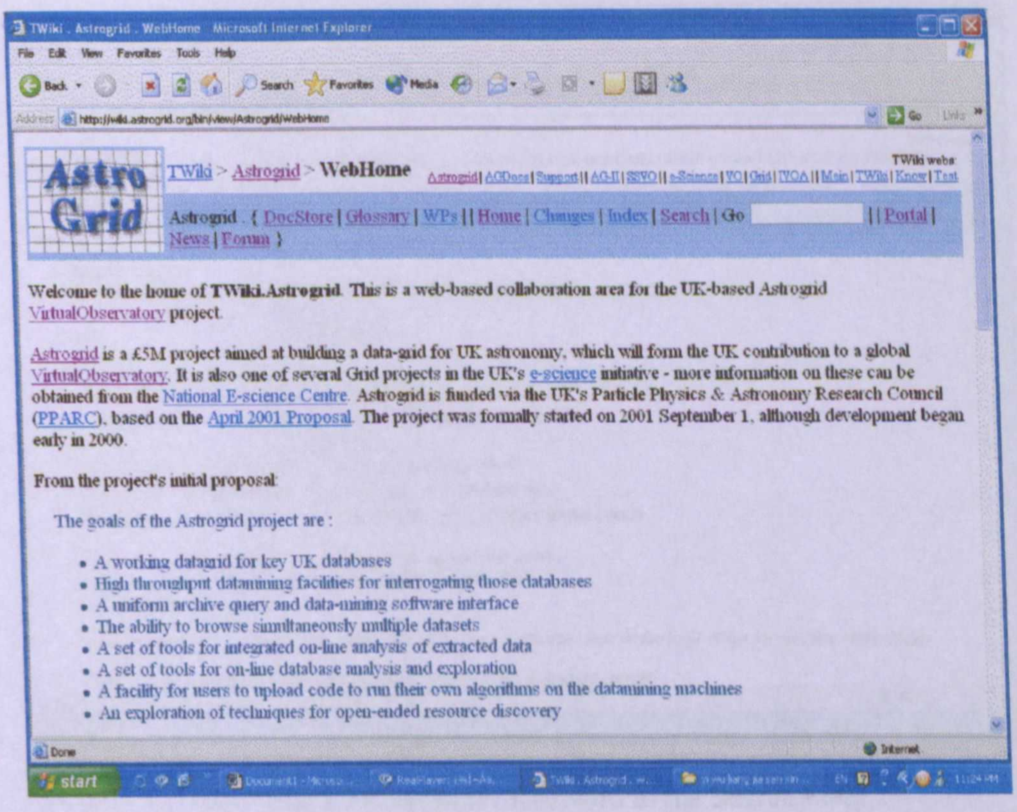
Astro Grid is an example of data grid that developed in the United Kingdom. The purpose of Astro Grid is built for explore astronomy field. There are two zones which found in the Astro Grid websites. Static pages and Collaboration pages are the two zones of Astro Grid websites.

Users can find the information about project from Astro Grid. Besides that, the project's personnel and activities also can found from this website. Users can follow the development of this first prototype of a Virtual Observatory.

The features or sites that will be added in the future are link together with Astro Grid through Collaboration pages. News, forum and wiki are the sites that linked to the Collaboration pages. The registered users can post articles, items of news, events and polls. One of admin staff will check the item that posted and release the item to the websites. Other users can comment on the items that posted in the news.

A number of grouped discussion areas are provided by the forum. Users can discuss the site-related topics, project-specific topics and general areas. The users can post new topic for discussion. The topics are categorized into several discussion areas.

Wiki is the sites that users least familiar with it. Registered users are allowed to change the pages on the sites and add their own pages. Figure 2.6 shows the main page of Wiki.


TWiki . Astrogrid . WebHome Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://wiki.astrogrid.org/bin/view/Astrogrid/WebHome
Go Links
Astro Grid
TWiki > Astrogrid > WebHome
TWiki webs:
Astrogrid | AGDocs | Support || AG-II | SSVO || e-Science | VO | Grid | IVOA || Main | TWiki | Know | Test
Astrogrid . { DocStore | Glossary | WPs || Home | Changes | Index | Search | Go || Portal | News | Forum }
Done
Internet
start

Welcome to the home of **TWiki.Astrogrid**. This is a web-based collaboration area for the UK-based Astrogrid VirtualObservatory project.

Astrogrid is a £5M project aimed at building a data-grid for UK astronomy, which will form the UK contribution to a global VirtualObservatory. It is also one of several Grid projects in the UK's e-science initiative - more information on these can be obtained from the National E-science Centre. Astrogrid is funded via the UK's Particle Physics & Astronomy Research Council (PPARC), based on the April 2001 Proposal. The project was formally started on 2001 September 1, although development began early in 2000.

From the project's initial proposal:

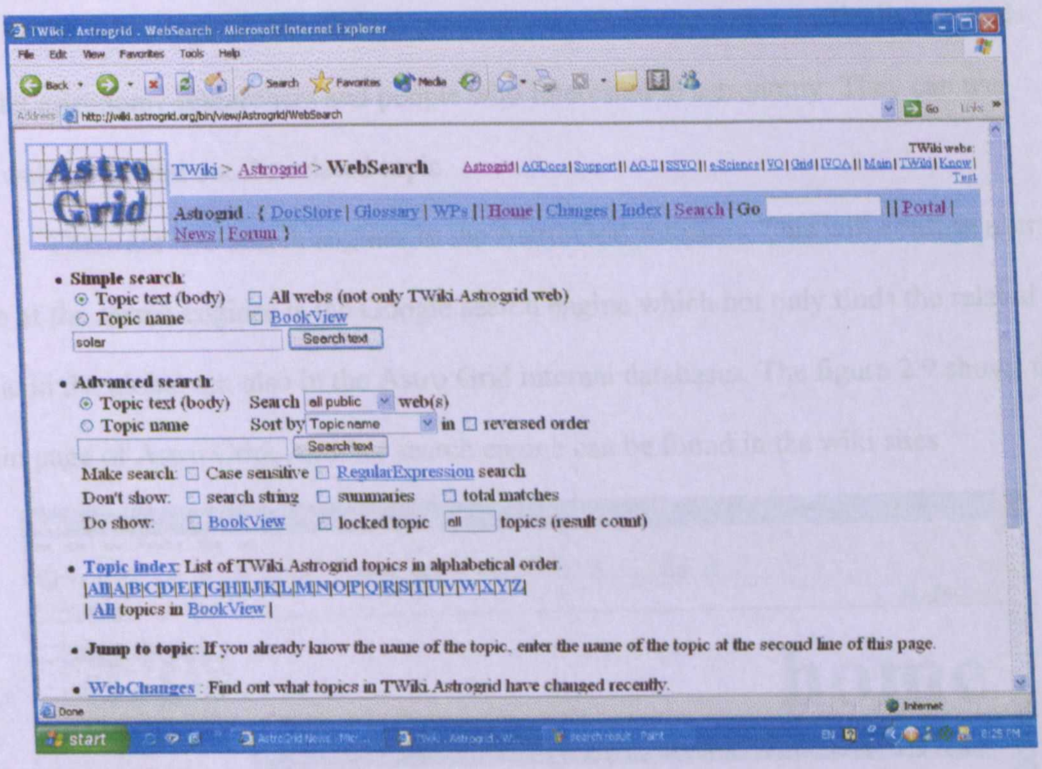
The goals of the Astrogrid project are :

- A working datagrid for key UK databases
- High throughput datamining facilities for interrogating those databases
- A uniform archive query and data-mining software interface
- The ability to browse simultaneously multiple datasets
- A set of tools for integrated on-line analysis of extracted data
- A set of tools for on-line database analysis and exploration
- A facility for users to upload code to run their own algorithms on the datamining machines
- An exploration of techniques for open-ended resource discovery

FIGURE2.6 Main Page of Wiki

2.4.2.2 THE FEATURES OF WIKI

The users can search some topics that related to astronomy by entering some keywords. The search results are show to the users. The search engine will search related topic based on the keywords. There are two types of search methods which are basic search and advanced search.


TWiki . Astrogrid . WebSearch - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://wiki.astrogrid.org/bin/view/Astrogrid/WebSearch
Go Links
Done
Internet
start

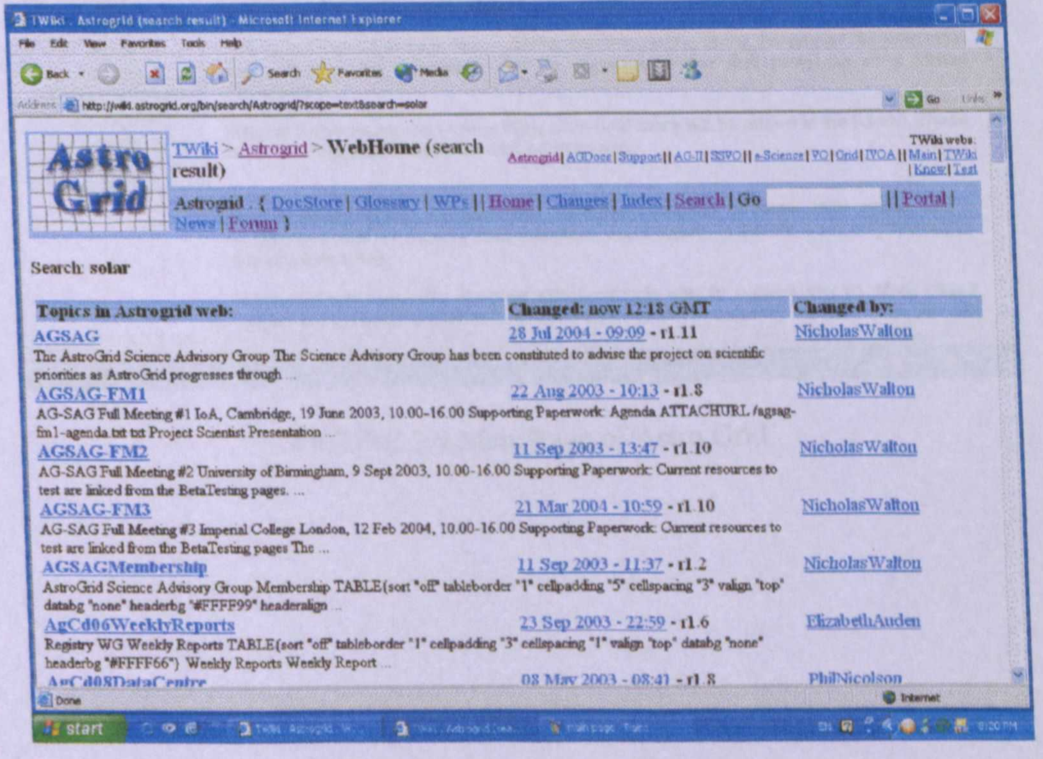
TWiki > Astrogrid > WebSearch

TWiki webs: Astrogrid | AGDocs | Support || AG-II | SSVO || e-Science | VO | Grid | IVOA || Main | TWiki | Know | Test

Astrogrid . { DocStore | Glossary | WPs || Home | Changes | Index | Search | Go || Portal | News | Forum }

- **Simple search:**
  ⦿ Topic text (body) □ All webs (not only TWiki.Astrogrid web)
  ○ Topic name □ BookView
  solar [Search text]
- **Advanced search:**
  ⦿ Topic text (body) Search all public web(s)
  ○ Topic name Sort by Topic name in □ reversed order
  [Search text]
  Make search: □ Case sensitive □ RegularExpression search
  Don't show: □ search string □ summaries □ total matches
  Do show: □ BookView □ locked topic all topics (result count)
- **Topic index:** List of TWiki.Astrogrid topics in alphabetical order.
  |All|A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|R|S|T|U|V|W|X|Y|Z|
  | All topics in BookView |
- **Jump to topic:** If you already know the name of the topic, enter the name of the topic at the second line of this page.
- WebChanges : Find out what topics in TWiki.Astrogrid have changed recently.

FIGURE 2.7 User Enter Keyword in the Search Field


TWiki . Astrogrid (search result) - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://wiki.astrogrid.org/bin/search/Astrogrid/?scope=text&search=solar
Go Links
Done
Internet
start

TWiki > Astrogrid > WebHome (search result)

TWiki webs: Astrogrid | AGDocs | Support || AG-II | SSVO || e-Science | VO | Grid | IVOA || Main | TWiki | Know | Test

Astrogrid . { DocStore | Glossary | WPs || Home | Changes | Index | Search | Go || Portal | News | Forum }

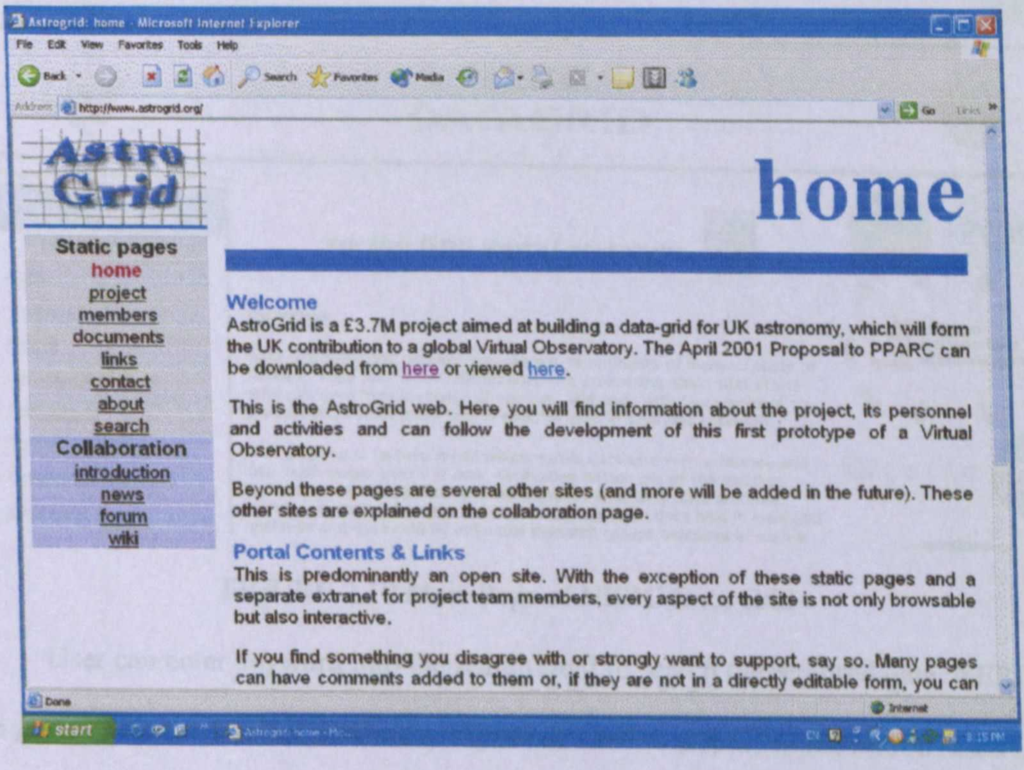
Search: solar

Topics in Astrogrid web:	Changed: now 12:18 GMT	Changed by:
AGSAG The AstroGrid Science Advisory Group The Science Advisory Group has been constituted to advise the project on scientific priorities as AstroGrid progresses through ...	28 Jul 2004 - 09:09 - r1.11	NicholasWalton
AGSAG-FM1 AG-SAG Full Meeting #1 IoA, Cambridge, 19 June 2003, 10.00-16.00 Supporting Paperwork: Agenda ATTACHURL /agsag-fm1-agenda.txt txt Project Scientist Presentation ...	22 Aug 2003 - 10:13 - r1.8	NicholasWalton
AGSAG-FM2 AG-SAG Full Meeting #2 University of Birmingham, 9 Sept 2003, 10.00-16.00 Supporting Paperwork: Current resources to test are linked from the BetaTesting pages. ...	11 Sep 2003 - 13:47 - r1.10	NicholasWalton
AGSAG-FM3 AG-SAG Full Meeting #3 Imperial College London, 12 Feb 2004, 10.00-16.00 Supporting Paperwork: Current resources to test are linked from the BetaTesting pages The ...	21 Mar 2004 - 10:59 - r1.10	NicholasWalton
AGSAGMembership AstroGrid Science Advisory Group Membership TABLE(sort "off" tableborder "1" cellpadding "5" cellspacing "3" valign "top" databg "none" headerbg "#FFFF99" headeralign ...	11 Sep 2003 - 11:37 - r1.2	NicholasWalton
AgCd06WeeklyReports Registry WG Weekly Reports TABLE(sort "off" tableborder "1" cellpadding "3" cellspacing "1" valign "top" databg "none" headerbg "#FFFF66") Weekly Reports Weekly Report ...	23 Sep 2003 - 22:59 - r1.6	ElizabethAuden
AgCd08DataCentre	08 May 2003 - 08:41 - r1.8	PhilNicolson

FIGURE 2.8 Search Result which Search by Keyword

The advantage of Atro Grid is providing a efficient and systematically methods that let astronomy researchers and people who interested in astronomy. They can use this website to find out the related topic.

There are two search engines in the Astro Grid websites. This will confuse users. One of the search engines is the Google search engine which not only finds the related topic in the global but also in the Astro Grid internal databases. The figure 2.9 shows the main page of Astro Grid. Another search engine can be found in the wiki sites.



Welcome

AstroGrid is a £3.7M project aimed at building a data-grid for UK astronomy, which will form the UK contribution to a global Virtual Observatory. The April 2001 Proposal to PPARC can be downloaded from here or viewed here.

This is the AstroGrid web. Here you will find information about the project, its personnel and activities and can follow the development of this first prototype of a Virtual Observatory.

Beyond these pages are several other sites (and more will be added in the future). These other sites are explained on the collaboration page.

Portal Contents & Links

This is predominantly an open site. With the exception of these static pages and a separate extranet for project team members, every aspect of the site is not only browsable but also interactive.

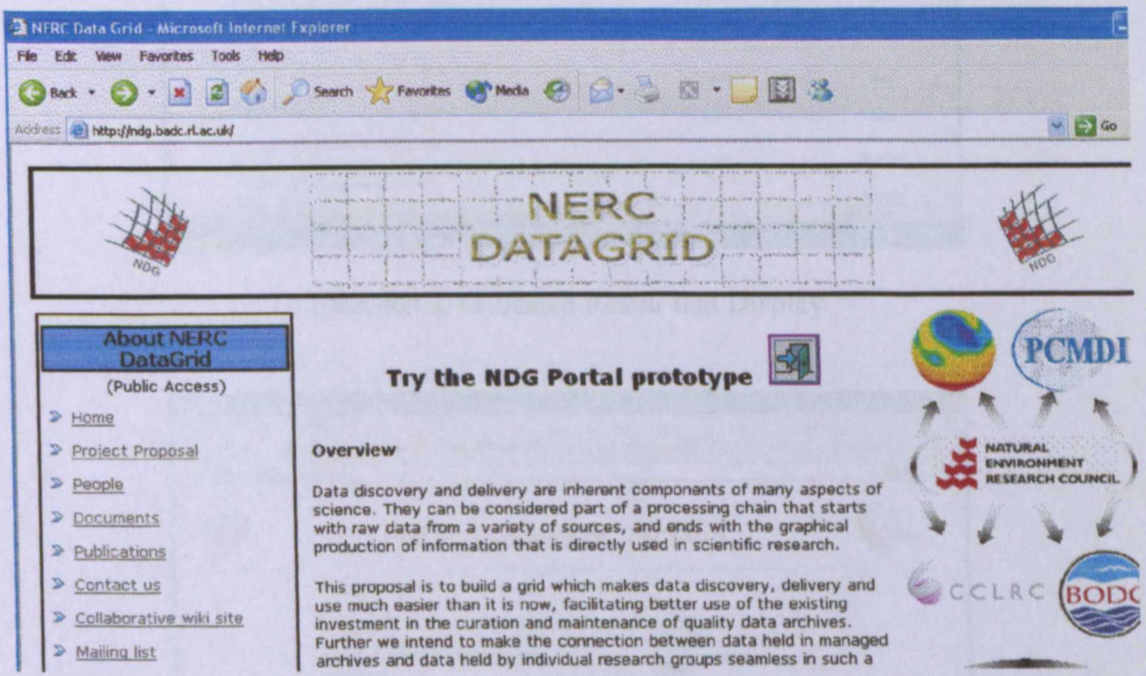
If you find something you disagree with or strongly want to support, say so. Many pages can have comments added to them or, if they are not in a directly editable form, you can

FIGURE 2.9 Main Page of Astro Grid

2.4.3 NERC DATA GRID

2.4.3.1 OVERVIEW

NERC data grid is to develop a portal to let users discover, use and deliver data. The portal for NERC is still under development. Hence, the prototype is uploading on the official websites. The main page of NERC data grid is show below.



NERC Data Grid - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites Media
Address http://ndg.badc.rl.ac.uk/
Go
NERC DATAGRID
NDG
NDG
PCMDI
NATURAL ENVIRONMENT RESEARCH COUNCIL
CCLRC
BODC

About NERC DataGrid

(Public Access)

- Home
- Project Proposal
- People
- Documents
- Publications
- Contact us
- Collaborative wiki site
- Mailing list

Try the NDG Portal prototype

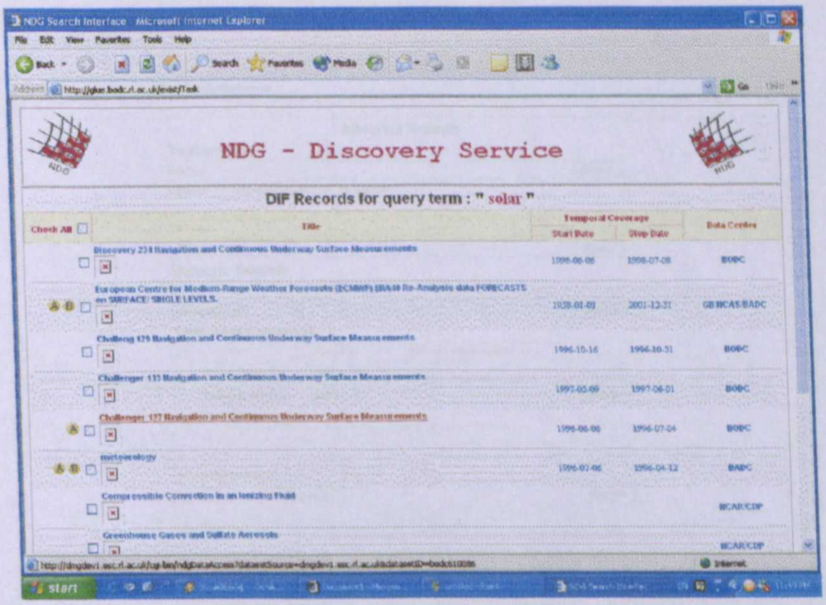
Overview

Data discovery and delivery are inherent components of many aspects of science. They can be considered part of a processing chain that starts with raw data from a variety of sources, and ends with the graphical production of information that is directly used in scientific research.

This proposal is to build a grid which makes data discovery, delivery and use much easier than it is now, facilitating better use of the existing investment in the curation and maintenance of quality data archives. Further we intend to make the connection between data held in managed archives and data held by individual research groups seamless in such a

FIGURE 2.10 Main Page of NERC Data Grid

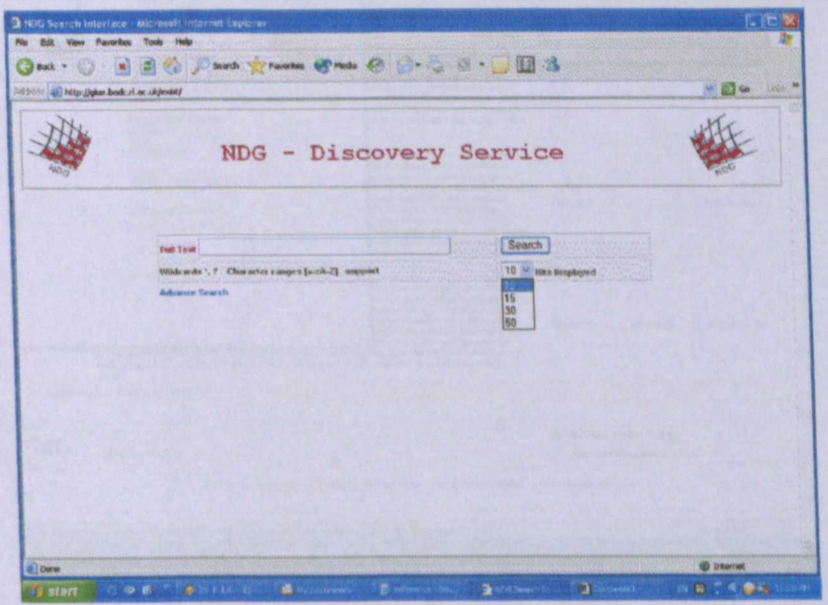
User can enter keyword into the search field to search the required data. NERC data grid allows users select how many hits those need to display in the search results. The maximum hits that allow users select is 50. The example of search result is shown in the figure 2.11.


NDG - Discovery Service

DIF Records for query term : " solar "

Check All □	Title	Temporal Coverage		Data Centre
		Start Date	Stop Date	
□	Discovery 234 Navigation and Continuous Underway Surface Measurements	1998-06-06	1998-07-08	BODC
Ⓐ Ⓑ □	European Centre for Medium-Range Weather Forecasts (ECMWF) ERA40 Re-Analysis data FORECASTS on SURFACE/ SINGLE LEVELS.	1958-01-01	2001-12-31	GB NCAS BADC
□	Challeng 125 Navigation and Continuous Underway Surface Measurements	1996-10-16	1996-10-31	BODC
□	Challenger 131 Navigation and Continuous Underway Surface Measurements	1997-05-09	1997-06-01	BODC
Ⓐ □	Challenger 127 Navigation and Continuous Underway Surface Measurements	1996-06-06	1996-07-04	BODC
Ⓐ Ⓑ □	meteorology	1996-07-06	1996-04-12	BADC
□	Compressible Convection in an Ionizing Fluid			NCAR/CDP
□	Greenhouse Gases and Sulfate Aerosols			NCAR/CDP

FIGURE 2.11 Search Result that Display


NDG - Discovery Service

Full Text [ ] Search

Wildcards *, ? Character ranges [a-zA-Z] support

10 hits displayed

Advance Search

FIGURE 2.12 Search Engine

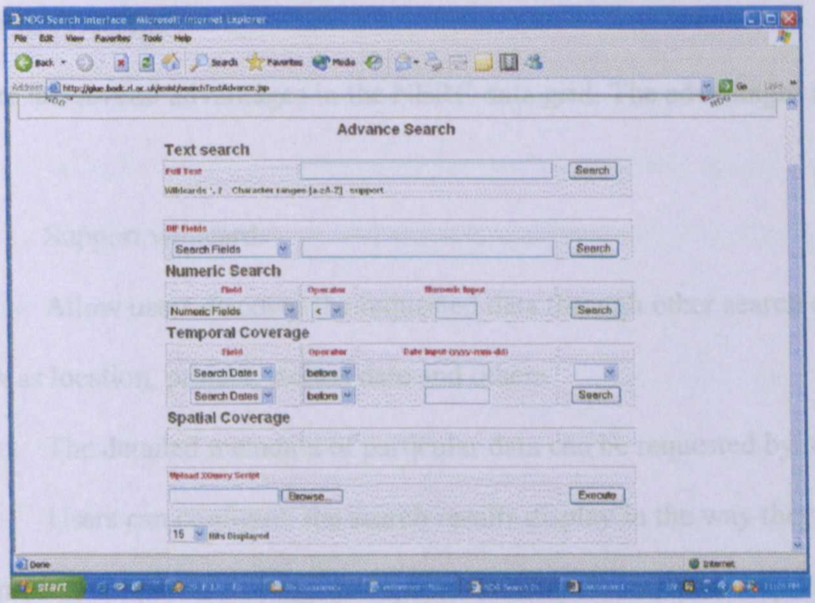


FIGURE 2.13 Advanced Search

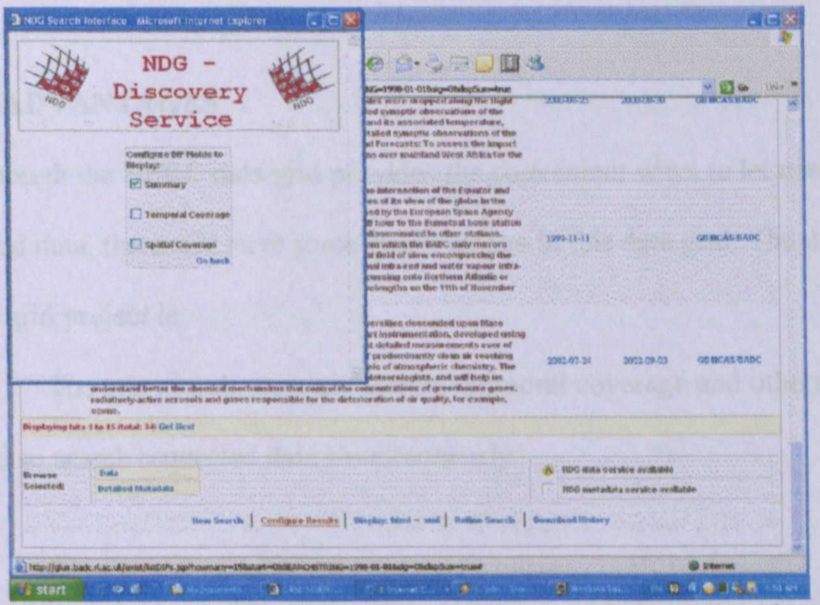

NDG Search Interface - Microsoft Internet Explorer
NDG - Discovery Service
Configure DIF Fields to Display:
☑ Summary
☐ Temporal Coverage
☐ Spatial Coverage
Go back
1999-11-11
GB NCAS BADC
2002-07-24
2002-09-03
GB NCAS BADC
Displaying hits 1 to 15 /total: 34 Get Next
Browse Selected:
Data
Detailed Metadata
NDG data service available
NDG metadata service available
New Search | Configure Results | Display: html - xml | Refine Search | Download History
Internet
start

FIGURE 2.14 Configure the Display Format of Search Results

2.4.3.2 ADVANTAGES

There are several advantages in the NERC data grid. The advantages are listed below:

- Support wildcards
- Allow users discover the requested data through other search condition such as location, project, started date and others.
- The detailed metadata of particular data can be requested by users
- Users can configure the search results display in the way they like. Summary, temporal coverage and spatial coverage is the three way that provided to display search result.

2.4.3.3 DISADVANTAGES

Although the NERC data grid provides the convenient ways to let users discover their required data, there still have some disadvantage in this data grid. The disadvantage of this data grid project is:

- The text search, numeric search, temporal coverage and others cannot be used to search requested data simultaneously.

2.5 COMPUTING TECHNOLOGY

There are several types of computing technologies in the real world. All these computing technologies are developed to improve performance and productivity. The computing technologies in the real world are grid computing, distributed computing, parallel computing, high-performance computing, clustering computing and peer-to-peer computing.

2.5.1 DISTRIBUTED COMPUTING

Distributed computing is an application can executed in a group of computers which connected in a network. A set of data is processing by allocating the data among the cooperating computers. The power of processing in distributed is same as supercomputer but the cost is cheaper. Hence, distributed computing is developed to replace supercomputer. Software which will defined what to process and how to divide the data among the computers need to be installed on each personal computer to handle the jobs processing in the distributed environment.

2.5.2 PARALLEL COMPUTING

Parallel computing is a technology which enables multiple computer resources used to solve a computational problem simultaneously. The resources that defined in parallel computing are a single computer with multiple processors; two or more computers are connected in a network or combination of both methods. The computational problems usually consists the following characteristics.

- The problem can split into discrete running parts. Then, all running parts can be solved simultaneously.
- Execute multiple program instructions at any moment in time.
- The problem can solved in less time with multiple compute resources.

There are two types of parallel computing. Single Instruction, Multiple Data (SIMD) and Multiple Instruction, Multiple Data (MIMD) are the two type of parallel computing.

2.5.2.1 SINGLE INSTRUCTION, MULTIPLE DATA (SIMD)

This type of parallel computing enable multiple processing units execute a single instruction at any given clock cycle. Besides that, each of the processing unit can operate on a different part data element. Instruction dispatcher which is a very high-bandwidth internal network and a large array of very small-capacity instruction unit normally can found in this type of machine. SIMD is suitable for processing specialized problem which need high degree of regularity. Image processing is one of the examples.

2.5.2.2 MULTIPLE INSTRUCTION, MULTIPLE DATA (MIMD)

Multiple instruction, multiple data (MIMD) currently is the most common types of parallel computing. This type of parallel computing enables every processing unit running a different instruction stream and operates with different data stream.

2.5.3 HIGH-PERFORMANCE COMPUTING (HPC)

High-performance computing is also known as supercomputer. High-performance computing is used to solve the computational problems that need significant processing power and need to quickly access. Besides that, HPC also can process a very large amount of data.

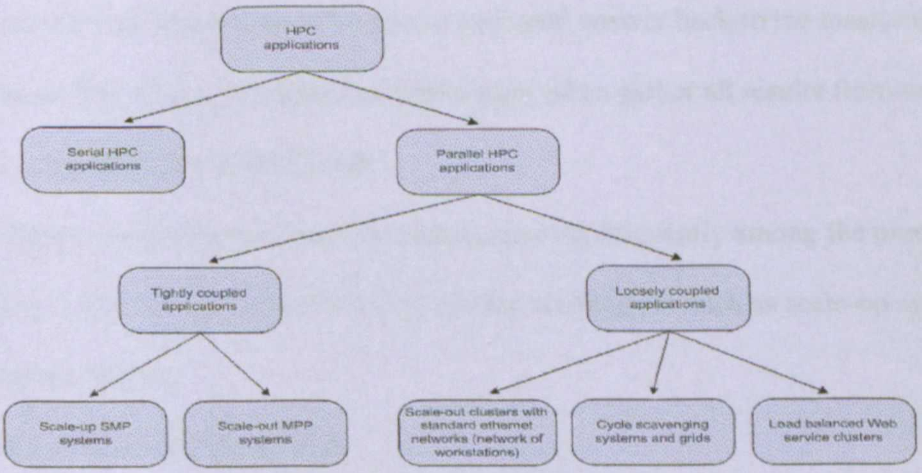

HPC applications
Serial HPC applications
Parallel HPC applications
Tightly coupled applications
Loosely coupled applications
Scale-up SMP systems
Scale-out MPP systems
Scale-out clusters with standard ethernet networks (network of workstations)
Cycle scavenging systems and grids
Load balanced Web service clusters

FIGURE2.15 High-Performance Computing Hierarchy

Figure2.15 shows the hierarchy of high-performance computer. Serial high-performance computing and parallel high-performance computing are the two types of high-performance computing.

2.5.3.1 SERIAL HIGH-PERFORMACE COMPUTING

Serial HPC use a single processor to execute the computational problems. The problem that faced in this technology is instruction cannot execute as fast as possible. This situation us caused by all instruction executed serially.

2.5.3.2 PARALLEL HIGH-PERFORMANCE COMPUTING

Parallel HPC use multiple processors to execute computational problems. This technology can be categorized into loosely coupled and tightly coupled.

2.5.3.2.1 LOOSELY COUPLED

This technology requires minimal inter-process communication. The communication model that used in loosely coupled involves a master and worker processors. The master keeps track of the task to be done. The worker processors are installed at every machine.

It performs the task which request by master and send answer back to the master when task finished. The master will generate final output when gather all results from worker.

2.5.3.2.2 TIGHTLY COOUPLED

Tightly coupled model requires communicating frequently among the processing units. This model can run in the following system architecture such as scale-up systems and scale-out systems.

2.5.3.2.2.1 SCALE-UP SYSTEM

When load increase, scale-up system allows more processors and more memory add to the single system. This will increase processing power and memory available for application

2.5.3.2.2.2 SCALE-OUT SYSTEM

An application can be scaled by partitioning the workload and spreading it across a set of cooperating system. Additional computer is added to increase processing power and memory available if load increase.

2.5.4 CLUSTER COMPUTING

Cluster computing uses multiple computers, multiple storage devices and redundant interconnections to form a large virtual single system. Load balancing is implementing in this cluster computing. The characteristics of cluster computing are listed below.

- The entire computing are same or similar type machines. Heterogeneous clusters still mostly under experimental.
- Tightly-coupled using dedicated network connections
- All machines share resources. Hence, the resources seem like a common home directory.

- The machine in clustering computing must trust each other.
- Must have software such as an MPI implementation installed to allow programs to be run across all nodes

Cluster computing is much smaller network which consists of fewer than 100 computers. The networking and software infrastructure for cluster computing is less mature. This problem makes the cluster computing difficult to develop as larger system.

2.5.5 PEER-TO-PEER COMPUTING

Peer-to-peer computing is a system which is decentralized, self-organizing distributed system. All or most communication in P2P computing is symmetric. Resources that shared are directly distributed among the systems. The goal of P2P computing is to satisfy the need of users. Figure 2.16 show the architecture of peer-to-peer architecture.

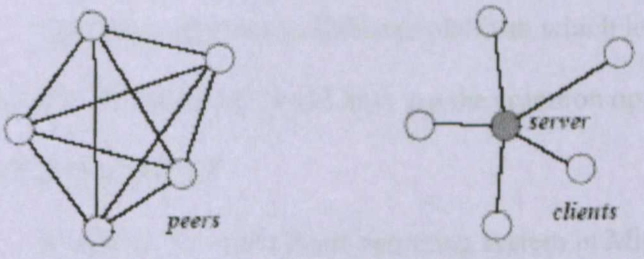


FIGURE2.16 Types of Peer-to-Peer Computing

There are two type of peer-to-peer computing architecture. Pure P2P architecture model and centralized architecture model are the P2P architecture. Pure P2P architecture is a technology that all peers can obtain resources or giving resources among the peers.

Example of this technology is Gnutella. The centralized architecture consists of central servers which serve all clients. Napster is the example for this type of architecture model.

2.6 OPERATING SYSTEM

Operating system is software which performs basic tasks, such as recognizing

input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers. Operating system can be classified to the following categories:

- Multi-user: Allows two or more users to run programs at the same time.
- Multiprocessing: Supports running a program on more than one CPU.
- Multitasking: Allows more than one program to run concurrently.
- Multithreading: Allows different parts of a single program to run concurrently.
- Real time: Responds to input instantly. General-purpose operating systems, such as DOS and UNIX, are not real-time

Operating system is a software platform which let application program running on top of it. Windows XP and Linux are the common operating system in the market.

2.6.1 WINDOWS XP

Windows XP is the latest operating system in Microsoft operating system family. It built on the core software code used in Windows 2000 and Windows NT workstation. NT kernel makes Windows XP more powerful, secure and stable then Windows Me, Windows 98 and Windows 95.

Windows XP support the NTFS file system which improve security. Users can encrypt files and folders and restrict access to files. This file system is more secure, more reliable and better performing file system than FAT32.

Windows XP has some enhancement to improve its user interface. The enhancements are:

- Redesign start menu

The most frequently used application is group together for easy access. Besides that, the five most-used applications will display to users first. Email and web browser are always available.

- Taskbar grouping

Keep user taskbar clean and organized. Users do not need to scroll through taskbars, clicking on buttons to find the document. In Windows XP, the system groups them according to application types.

2.6.2 WINDOWS 2000 PRO

Windows is an operating system which developed based on the NT technology. This operating system is design for business computing usages. Hence, the requirement for hardware is higher than the previous Microsoft operating system family. This is caused by the commercial purpose.

Window 2000 Pro is more stable and reliable than Window 98 and NT. When Window 2000 running for a long period, there are no any crash occur in this operating system. Active Directory which is a new feature in Window 2000 Pro allows a virtual private network is set up in a company. The encryption is performed in the Active Directory.

There is a weakness which occurs in Windows 2000 Pro. The weakness is lack of hardware and software compatibility. The hardware that compatible with Window 2000 Pro defined in Hardware Compatible Lists (HCL) while Application Compatible Lists define software that compatible with Window 2000 Pro.

2.6.3 LINUX

Linux is an operating system for internet/ intranet serving purposes. Besides that, it is a free UNIX type operating system. Linus Torvalds and developers around the world develop Linux. Linux source code is free and available for everyone who interested.

Linux is a stable and high performance operating system. It has made some progress, primarily in functionality important to internet infrastructure and web server capabilities. The capabilities are include a greater selection of drivers, easier installation, GUI based front ends for administrate and window management.

The advantages of Linux are listed below:

- Everyone who interest can freely to use Linux and the license will not expire. This is because Linux is licensed by General Public License (GNU).
- Linux is an ultra stable operating system. It do not occur crash although it work for a long period.
- Like UNIX, Linux support multitasking computing environment. Hence, a high performance computing for workstation or network provided by Linux.
- Linux strongly support for network connectivity. Besides that, it always compatible with the client and server architecture system.

2.6.4 UNIX

The first operating system written in C programming is UNIX. UNIX is not same as Windows which is a single operating system that owned by anyone. A class of similar system forms the UNIX operating system. Besides that, it is an open source and standard operating system. Developers can contribute ideas to enhance or improve the function of UNIX. Hence, many versions of UNIX are in the market now. Solaris, AIX, HP-UX, BSD/OS and MacOS are the different implementation of UNIX system.

Multitasking computing is implementing in the UNIX system. It can run simultaneously running different independent processes. Besides that, UNIX support multiple users using an individual UNIX system or running the same programs or files in the same time.

A powerful programming environment is provided by UNIX system for C, C++, FORTRAN and Java compilation. The developers who interested can use free development tools which available to design UNIX system. A features that provided by UNIX is portability. The codes are not change much when UNIX system move from one PC to another.

2.7 PROGRAMMING LANGUAGE

Several programming approaches can be taken as approach to the development of the grid application. Hence, it is need to consider many advantages and disadvantages of the programming approaches.

2.7.1 JAVA PROGRAMMING

Java is a small, simple, safe, object-oriented, interpreted or dynamically optimized, byte-coded, architecture-neutral, garbage-collected, multithread programming language with a strongly typed exception-handling mechanism for writing distributed, dynamically extensible programs. Java is developed by Sun Microsystems. It is a powerful language built to be secure, cross-platform and international.

Java is design for use in internet environment. Complete applications which can run on a single computer or distributed among servers and clients in a network can created by using Java programming approaches. Besides that, it can also be used to build applets which use as a part of web pages.

2.7.2 C++ PROGRAMMING

C++ which is an extension of C programming language was developed by Bjarne Stroustrup at Bell Laborites. C++ provides capabilities for object-oriented programming. The C++ language facilities structured and disciplined approach to computer design. Classes and function are the pieces that consisted in C++ programming language. Programmer can program each of pieces to form a C++ program. C++ programmers take advantage of the rich collections of existing classes and functions in the C++ standard library.

2.7.3 VISUAL C# .NET

C# is a simple, object-oriented, modern and type-safe programming language. It combined high productivity of Rapid Application development (RAD) and raw power of C++.

Visual C# includes an interactive development environment, visual designers for building Windows and web applications, a compilers and a debugger. It is also like other programming languages in the Microsoft Visual Studio .Net which provides access to the Microsoft .NET Framework. .NET Framework includes a common execution engine and a rich class library.

Visual C# can interoperate with other languages, across platforms with legacy data by virtue of the following features:

- ➢ Full interoperability support through COM+ 1.0 and .NET Framework services with tight library–based access
- ➢ XML support web-based component interaction
- ➢ Versioning to provide ease of administration and deployment

2.7.4 EXTENSIBLE MARKUP LANGUAGE (XML)

Extensible Markup language (XML) which is a simple, very flexible text format derived from Standard Generalized Markup Language (SGML) is designed to support the large-scale electronic publishing. XML Working Group which originally known as SGML Editorial Review Board developed the XML. This workgroup is formed under the auspices of the World Wide Web Consortium (W3C) in 1996.

The design goals for XML are:

- ➢ XML shall be straightforwardly usable over the Internet
- ➢ XML shall support a wide variety of applications

- XML should be compatible with SGML
- It shall be easy to write programs which process XML documents
- The number of optional features in XML is to be kept to the absolute minimum, ideally zero
- XML documents should be human-legible and reasonably clear
- XML documents should be human-legible and reasonably clear
- The design of XML shall be formal and concise
- XML documents shall be easy to create
- Terseness in XML markup is of minimal importance

XML documents which is a class of data objects is described by XML. Besides that, XML also partially describes the behavior of computer programs which process them. Each XML document has both a logical and physical structure. The physical structure of XML document is composed of units called entities. Entities are the storage units which contain either parsed or unparsed data. The declarations, elements, comments, character references and processing instructions composed the logical structure of the document. The logical structure is indicated by explicit markup. The logical and physical structures must nest properly.

2.7.5 WEB SERVICES DESCRIPTION LANGUAGE (WSDL)

In the web community, the communications protocols and message formats are standardized. WSDL which is XML-based document addressed this need. The network services are described as collections of communication endpoints. These endpoints have the capability to exchange messages.

Services that defined in the WSDL document is a collections of communication endpoints or ports. The concrete networks deployment or data format bindings are separated from the abstract definition of endpoints and messages. The abstract definitions can be reuse in this condition.

2.8 INTROCUTION TO DISTRIBUTED DATABASE

A collection of multiple, logically interrelated databases distributed over a computer network is called distributed database. A distributed database management (Distributed DBMS) is the software that permits the management of distributed databases and makes this distribution transparent to users. Actually, a distributed database is not a collection of files that can be individually stored at each node of the network.

There are two types of distributed databases such as homogeneous distributed database and heterogeneous distributed databases. The homogeneous distributed databases are the distributed databases which using the same distributed database management systems. There are at least one or more different types of databases in heterogeneous databases.

Data is physically stored across several sites in distributed database system. A DBMS manage each site to ensure that each site runs independently of other sites. A computer network is used to connect the processors at these sites. In distributed database system, the distributed data should be logically related where the relationship is defined according to some structural formalism. Besides that, a common interface is provided to access data.

A multiprocessor system is generally considered to be a system where two or more processors share some form of memory. There are two type of memory sharing. The primary memory sharing is called shared memory which is tightly coupled and the secondary memory is called shared disk which is loosely coupled.

The implementation of distributed database should not distinguish between client and server machines. Ideally, each site can perform the functionality of a client and a server. Hence, peer-to-peer architecture is the ideal protocol in the implementation of the distributed database.

Distributed database system provides some strength and advantages to the users. The strengths of distributed database system are:

➢ Transparent management of distributed and replicated data

The separation of higher-level semantics of a system from lower-level implementation issues is referred to the transparency. Transparent system hides the implementation of the system. To improve the performance and reliability of the system, some of the data are duplicated at other sites. Fully transparent access means that the users can still pose the queries without paying any attention to fragmentation, location or replication of data. All of these issues just let the system to solve them.

- Data independent

This is the fundamental of the transparency. It refers to the immunity of user applications to changes in the definition and organization of data and vice versa. There are two types of the data independence. First, logical data independence refers to the immunity of user applications to changes in the logical structure of the databases. Then, another type of data independence is physical data independence. All the details of the storage structure are hiding from the user applications. Hence, the user application which is written should not concern the physical data organization in the physical data independence.

- Network transparency

The user should be protected from the operational details of the network. The database applications either run on a centralized database or distributed database are not difference to the user applications. Location transparency and naming transparency are the kinds of network transparency. In location transparency, the command used to perform a task is independent of both the location of the data and the system on which an operation is carried out. A unique name is provided for each object in the databases. This mechanism is performed in the naming transparency.

- Replication transparency

It may not be a good idea to have too many copies of the data if the user applications are predominantly update-oriented. Replication transparency refers only to the existence of replicas not their actual location.

- Fragmentation transparency

Each database relation is divided into smaller fragments. Each of the fragments treats as a separate database objects. Fragmentation transparency can improve the performance, availability and reliability. Besides that, it reduces the negative effects which are carried out by replication. There are two types of fragmentation transparency. Horizontal fragmentation and vertical fragmentation is the two types of this transparency.

- Reliability through distributed transactions

It can eliminate single points of failure. It performs a consistent database state to another consistent database state even when a number of such transactions are executed concurrently and even failures occur. The concurrent execution of these two programs is needed to be synchronized by the system. To encapsulate a query within transaction boundaries, it is sufficient to declare begin of the transaction and its end.

- Improved performance

Performance of distributed database can be improved based on the two points.

- Enable data stored in close proximity to its point of use. This mechanism is called data localization.
- The inherent parallelism of distributed systems may be exploited for inter-query and intra-query parallelism.

The database is multiplexed by maintaining two copies. One of the copies is for ad hoc querying. This copy also called query database. Production database is the copy that updates by the application program. At regular intervals, the production database is copied to the query database.

- Easier system expansion

The new processing and storage power is easy adding into the network. An aspect of easier system expansion is economics. It normally costs much less to put together a system of smaller components with the equivalent power of a mainframe.

Although distributed database system has several of strength, it still consists of several problems which still not overcome. The problems of distributed database system are:

- Complexity

Distributed database system problems are inherently more complex than centralized database management ones as they include not only the problems found in a centralized environment, but also a new set of unresolved problems.

- Cost

Cost for distributed database system increase because the additional hardware is required. Besides that, more complex software and communication is added to solve some of the technical problems.

2.9 INTRODUCTION TO REPLICATION

Replication is the process that copies the data to the multiple databases. These copies of data are synchronized by replicating modifications in one copy to all the others. The motivation to implement replication in the databases is ensuring that single points of failure are eliminated. The overall system is still available even when failures occur in more than one site. The system availability and performance are improved by the replication. There are two types of replication protocol. They are strict replica control protocols and lazy replication protocols.

2.9.1 STRICT REPLICA CONTROL PROTOCOL

Strict replica control protocol is the protocol which enforces one-copy equivalence as the correctness criteria. The correctness criteria requires all the copies are synchronize and consistent after the end of each update transactions. To convert a logical read to read on any one of replicas and converts a logical write to all the replicas, the Read One Write All (ROWA) is introduced. This technique consists of a weakness. When one of the replicas is not available, the update transactions cannot terminate.

There is available solution to solve this problem. Read One Write All Available (ROWA-A) is another alternative to solve the problem. The idea of ROWA-A is the write commands are executed in the all the available copies and the transactions. The copies which are not available at the time can catch up after they are available.

2.9.2 LAZY REPLICATION PROTOCOLS

Lazy replication protocols do not attempt to perform the Write operations on all copies of the data item within the context of the transaction that updates that data item. The update on or more copies and later propagate the changes to all the other copies are performed in the lazy replication protocols.

The ownership parameter defines the permissions for updating replica copies. The replica copy which is can updatable is called primary copy. The other replica is called secondary copy. A master for the object is the node which stores the primary copy. Then, the node which stores secondary copy is called slave.

2.10 DATABASE MANAGEMENT SYSTEM

There are several databases which currently used in the industry. Databases are commonly used in the industry to keep structured collection of data. Oracle, SQL Server, IBM DB2 and MySql are the example of databases.

2.10.1 ORACLE

Oracle which is a multi-user database provides unprecedented ease-of-use and pre-tuned and pre-configured for today's dynamic workgroup and line-of-bus environment.

Fully integrated set of ease-to-use management tools, full distribution, replication and web features are included in the Oracle. To provide highest levels of availability, Oracle provides these functionalities such as fast fail over, easier management and zero data loss disaster. Data Guard which is the only complete data protection solution available on the market realizes the zero data loss disaster protection.

The platforms which are supported by Oracle are UNIX, Windows and Linux. However, it is expensive and separate licenses are required for each of its database engine.

2.10.2 MICROSOFT SQL SERVER 2000

Microsoft SQL Server 2000 is a client/server relational database management system. This database management system is designed for small and medium business application. The Microsoft SQL Server 2000 contains the following features.

- Provide analysis (OLAP) services and data mining
- Data Transformation Services (DTS) that automates extract, transform and load data routines from heterogeneous sources
- Using English query
- Rich XML support
- Enables web access to data
- Provides web and application hosting

2.10.3 MySQL

MySQL is the most popular open source SQL database. MySQL AB which is a commercial company that founded by the MySQL developers develop, distribute and support MySQL. Users can download the software from the official websites. The source code is possible modified by users according their needs.

MySQL is a relational database management system. Data are stored in separate tables rather than putting all the data in one big storeroom. The speed and flexibility will be improved.

MysQL is ideal for small and medium sized applications because it is a small, compact and easy to use database server. MySQL can operate on the UNIX, Linux and Windows platforms.

The features of MySQL are:

- Written in C and C++

- Fully multi-threaded using kernel threads. This means it easily use multiple CPUs if available.
- Easy to add another storage engine.
- All ODBC 2.45 functions are supported
- Clients may to the MySQL server using TCP/IP sockets, UNIX sockets or Named Pipes (NT).

2.10.4 IBM DB2

IBM DB2 is a relational database management system. This is one of the relational database management systems in the IBM products family. "Shared-nothing" architecture in DB2 enables a common code base to be used across these environments. Servers in a DB2 shared-nothing cluster work independently and in parallel on a subset of the overall data and on a subset of the SQL requests received by the cluster DB2 already integrate some new features in the database management system. The new features of DB2 are:

- Automatic computing which are the servers, operating systems and middleware can diagnose and correct problems without human intervention already be emphasized in the recent edition of DB2. database self-management and automation are the particular areas that focus by DB2.
- Support for standard-based web services
- Support Grid computing
- The "e-business on demand" business model requires an operating environment built on open standards to allow quick and cost-effective

innovation and reconfiguration. The infrastructure to support e-business on demand must be reliable, scalable and secure.

DB2 can operate on the UNIX, Linux and Windows platform.

2.10.5 MICROSOFT ACCESS

Microsoft Access is a relational database management system. It is also is a part of Microsoft Office family. The features of Microsoft Access are:

- Databases wizard is provided to help novice users to create their own database.
- Speech Recognition is provided that allows a user to dictate text and navigate menus using speech and voice commands.
- Compact and repair functionality that recovers files with broken forms and report.
- Provides encryption service that allows users to encrypt the databases which are created by them.
- Support XML
- Bind data access to embedded or linked XML files

2.11 DATA ACCESS TECHNOLOGIES

There are various kinds of data access technologies in the market. ODBC, OLE DB and ADO .NET are the examples of the data access technologies.

2.11.1 OPEN DATABASE CONNECTIVITY (ODBC)

ODBC is the most widely supported standard for data access. Microsoft Corporation is the developer of the ODBC. Its interface is a C programming interface that makes it possible for application to access data from a variety of database management systems. A middle layer which called database driver is inserted between an application and the DBMS. The database driver translates the application's data queries into the commands that the DBMS understands. To realize this work, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them. This is called ODBC-compliant.

The ODBC is not only available in Windows platforms but also UNIX, Macintosh and other platforms. It is also suitable to the large number of both relational and non-relational databases. ODBC is suitable to use when

- Client applications are Win32 only
- Client applications all support ODBC and users have an available driver for their database platform.

2.11.2 OBJECT LINKING AND EMBEDDING DATABASE (OLE DB)

OLE DB is not standards based. It is a standard that was intended for Win32 and ActiveX in particular. OLE DB is a set of COM-based interfaces that expose data from a variety of sources. Microsoft Connection Pooling and Distributed Transactions are supported by OLE DB. It interacts directly with the data store. There are a few application that support OLE DB drivers.

2.11.3 ACTIVE X DATA OBJECTS .NET (ADO.NET)

ADO .NET is the latest Microsoft technology that provides consistent access to data sources such as SQL server. It is designed to replace Data Access Objects (DAO) and Remote Data Objects (RDO). All sorts of different types of data which includes web pages, spreadsheets and other types of documents can be accessed by ADO .NET.

Data providers are used for connecting to a database, executing commands and retrieving results in a ADO.NET. The Data Set contains a collection of one or more DataTable objects made up of rows and columns of data. ADO .NET DataSet object can also be used independently of a .NET Framework data provider to manage data local to application or sourced from XML.

2.12 INTRODUCTION TO THE WEB SERVICES

A web services are self-contained and modular applications which support interoperate machine-to-machine interaction over a network. A interface is provided by web service to let other system to interact with web services. The interaction between web services and other system involved SOAP messages which typical conveyed using HTTP with an XML serialization in conjuction with other web-related standards. The major industry intiatives are Microsoft (.Net), IBM (Dynamic e-Business) and Sun Microsystem (Sun One).

Open Grid Services Architecture (OGSA) proposes the three main standards to implement web services in grid application. The three standards are Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL) and Web Service Inspection (WSI). The folowing is a briefly description of these standards.

- SOAP

It used to provide messaging between a Service Requestor and a Service Provider. It is a simple enveloping mechanism for XML payloads and defines a Remote Procedure Call (RPC) mechanism and conventions.

- WSDL

It is a XML document. A set of endpoints on messages is defined by WSDL that containing either document-oriented (messaging) or remote procedure calls (RPC) payload. WSDL is extensible to allow description of endpoints and the concrete representation of their messages fro a variety of different message formats and network protocol.

- WSI

It is a simple XML language. Related conventions for locating service descriptions which published by a service provider is did by WSI. A collection of service descriptions and links to other sources of services descriptions can be a content of a WS-Inspection Language (WSIL) document.

The mechanism which used to exchange messages is documented in a Web Service Description (WSD). WSD which written in WSDL is machine-processable specification. Messages format, datatypes, transport protocols and transport serialization formats are defined by WSD. All these information are used between the requester agent and provider agent. Besides that, some other related information can be also defined by WSD. The example is the network location at which a provider agent can be invoked.

The service description represents an agreement which govern the mechanics of interacting with that service. Hence, URL to a WSDL document is contained in a service description. Service description also consists of the URL to another WSIL document.

A concrete agent which is a computational resource must be implemented in the web services. Agent is a pieces of software or hardware in the web services. The purpose of implementing agent is to reliaze or request zero or more web services. The agent actually act as person or organization that perform the task.

2.12.1 OVERVIEW of ENGAGING WEB SERVICES

The two entities which operate in the web services are the provider entity and the requester entity. The provider entity is the person or organization which provides an appropriate agent to implement a particular service. To make use of a provider entity's web service, the requester entity will use requester agent which can interact with provider agent. The figure 2.17 below shows the relationship between provider entity and requester entity.

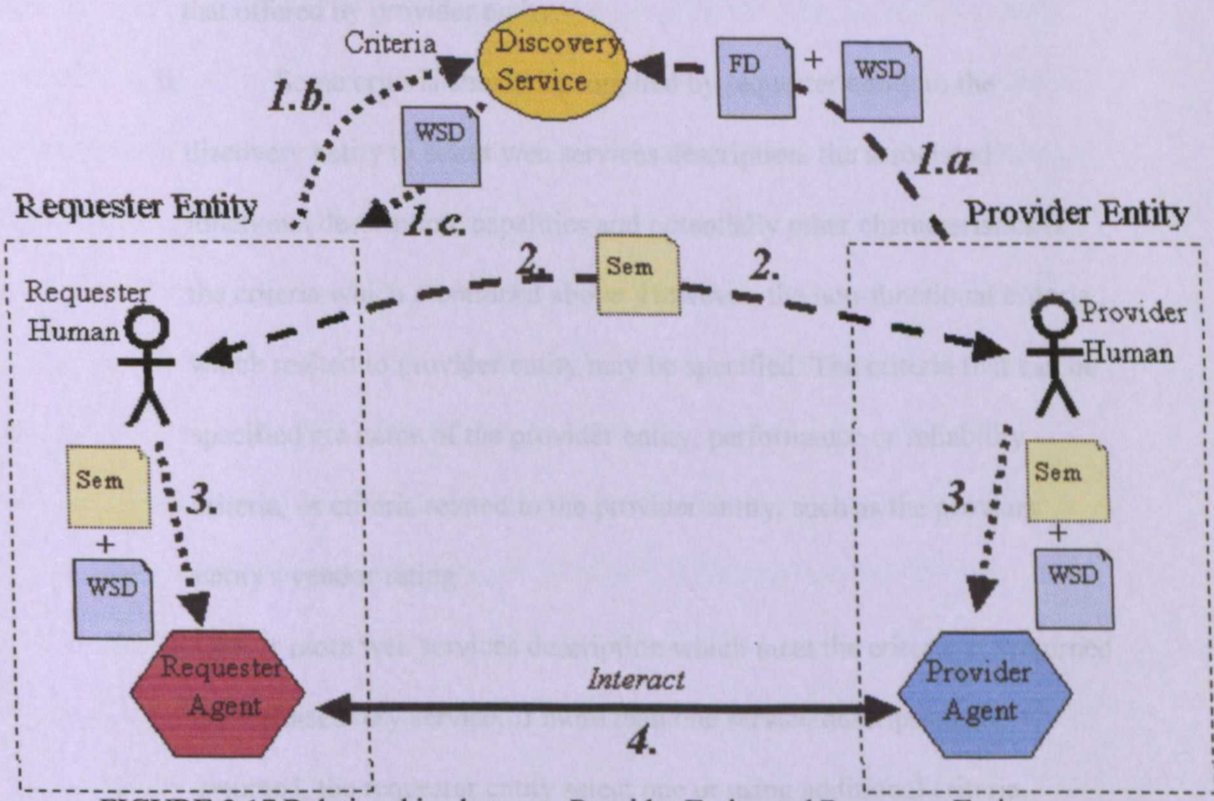


FIGURE 2.17 Relationships between Provider Entity and Requester Entity

A discovery service is needed when the requester entity which wishes to intiate an interaction with provider entity does not know what provider agent it wishes to engage. The discovery is a service that facilitates the process of performing discovery. Either the requester agnet, the provider agent or some other agent can perform this service.

There are several steps that should follow to engage a service after discovery services performed.

1. The requester and provider entities " become known to each other"

i. Both web service description and an associated functional description of the service are obtained by discovery service. The functional description is a description of the functionality of the service that offered by provider entity.

ii. Some criteria should be supplied by requester entity to the discovery entity to select web services description. the associated functional description, capalities and potentially other characteristics is the criteria which mentioned above. However, the non-functional criteria which realted to provider entity may be specified. The criteria that can be specified are name of the provider entity, performance or reliability criteria, or criteria related to the provider entity, such as the provider entity's vendor rating.

iii. One or more web services description which meet the criteria are returned by the discovery service. If more than one service description are returned, the requester entity select one or using additional criteria.

2. The semantics of desired interaction hsould be agreed by the requester and provider entities. Normally, the semantic is defined by provider entity. The requester just follow it. Besides that, the semantics can be defined in other ways. Example, the requester and provider entities can use the semantic which is defined by some industry standards. Step 2 requires the parties agree on the service description.

3. The service description and semantics are input to both the requester agent and the provider agent.

4. SOAP messages exchanged between the requester agent and provider agent on behalf of their owners.

There are two types of discovery process which state above. The two type of discovery process are manual discovery and autonomous discovery. Manual discovery defined that a discovery service which used by a requester *human* to locate and select a service description that meets the desired functional and other criteria. Manual discovery typically is perform at the design time. Under autonomous discovery, a requester agent uses discovery service to perform this task.

A number of condition must be establish to ensure computer programs can interact with each other successfully.

- A physical connection must exist between the computer program
- There must be agreement on the form of data. This agreement will define whether data is lines of text, XML structure and etc.
- The aggreement must be shared by two or more programs as to the intended meaning of data
- There must be agreement as to the implied processing of messages exchanged between the programs.

2.13 PROTOCOL

2.13.1 LIGHTWEIGHT DIRECTORY ACCESS PROTOCOL (LDAP)

Lightweight Directory Access Protocol (LDAP) is a protocol for accessing online directory services. LDAP is a simple protocol for updating and searching directories running over TCP/IP. This protocol makes use of X.500 which is too complex for simple internet clients to use.

There is several vendors support this protocol in the industries. The main vendors are:

- Apple
- AT&T
- Banyan
- HP
- IBM / Lotus
- Microsoft
- Novell
- Oracle
- Sun Microsystems

2.13.1.1 PROTOCOL MODEL

In LDAP model, protocol request which describe the operation that to be performed to a server is transmitted by a client. The server will perform the necessary operation in the directory after receive the request. After completion of the operations, server returns response to the requesting client. The response may contain the result of the operations or errors that occurs during the execution of operations.

No provision was made for protocol servers returning referrals to client in LDAP version 1 and 2. For improved performance and distribution LDAP, LDAP version 3 permits the servers to return to client referrals to other servers. The work which contacts with other server can be offloaded by the server to progress operation.

The core LDAP operations can be mapped to a strict subset of X.500 directory abstract. The relationships between LDAP protocol operations and DAP operations is not one-to-one mapping. Multiple DAP requests are needed to make in the server implementation which acts as a gateway to X.500 directories.

LDAP is run over connection-oriented and reliable transports. The LDAPMessage PDUs are mapped directly onto the TCP bytes stream. A protocol listener which provided by server implementation should be put on port 389.

2.13.2 X.509

X.509 is widely used standard for defining digital certificates. Actually, X.509 not yet officially defined or approved for standardization usage. Hence, the vendors implement X.509 in several of ways according to their needs. The goal of the Internet Public Key infrastructure (PKI) is to meet the needs of deterministic, automated identification, authentication, access control and authorization functions. X.509 can be implemented in the following applications such as WWW, electronic mail, user authentication and IPSec.

Users can gain the X.509 certificate from the trusted third-party which called certification authority (CA). Before they relying on the authentication or non-repudiation services associated with the public key in a particular certificate, they should review the certificate policy which generated by certification authority (CA).

2.13.3 SECURE SOCKET LAYER (SSL)

Secure sockets layer is one of the common security mechanisms that provide on the internet. Netscape is the developer for this protocol which used extensively by web browsers to provide secure connections. This protocol is important for credit card number and other sensitive data transmission. All SSL-protected HTTP transfer uses port 443.

Cryptography which is the encode data in a way that only can be decoded by its intended recipient is used in SSL. The third party who might be able to intercept the information cannot decode the information.

There are several fundamental capabilities to establish a communication over Internet.

- ➢ SSL server authentication

This can let user to confirm the server's identity. A certificate which issued by certificate authority (CA) is sent to the client. The client use standard techniques of public-key cryptography to check server's certificate and public ID are valid.

- ➢ SSL client authentication

User's identity is check by the server. Server check the certificate which sent by user. The public-key cryptography is used to check the certificate.

- ➢ An encrypted SSL connection

Information is sent between a client and server in a encrypted SSL connection. The high degree of confidentiality is provided.

2.14 MIDDLEWARE

Middleware is software that contains some services that can called by the Grid applications which is on top of it. Developers need to follow the some algorithm from the middleware to call the services from the middleware. The examples of the middleware are Globus, OGSI.Net, MS.Net Grid and others.

2.14.1 GLOBUS TOOLKIT

Globus Toolkit is an open standard software developed and blueprint by the Globus Project. Three operations that support by Globus built on top of a security infrastructure. They are resource management, data management and information services. Security functions such as single/mutual authentication, confidential communication, authorization and delegation are provided by the Grid Security Infrastructure (GSI).

Resource allocation, submitting jobs and managing job status and progress are the functions that supported by resource management. Actually, Globus toolkit does not provide its own job scheduler. It just provides tool and interfaces needed to implement schedulers for finding available resources and automatically send job to the suitable machines. Third-party schedulers usually are used.

Information services provide support for collecting information in the grid and for querying this information. The protocol that implements in the information services is Lightweight Directory Access Protocol (LDAP). File transferring among machines in the grid is supported by data management. Besides that, data management also manage theses transfers.

2.14.1.1 COMPONENTS OF GLOBUS TOOLKIT

Globus toolkit provides several components to support the operation of Grid applications. The components of Grid are:

- Grid Security Infrastructure (GSI)

GSI provides elements for secure authentication and communication in a grid. The protocols that used in the GSI are Secure Socket Layer (SSL), public key encryption and X.509 certificates. The main functions of GSI are single/mutual authentication, confidential communication, authorization and delegation.

- Grid Resource Allocation Manager (GRAM)

The remote execution and status management of the execution is supported by GRAM. Gatekeeper daemon which located on remote host handle the job that submitted by client. A job manger is created by gatekeeper. The job manager start and monitor the job. When the job finished, the job manger send the status information back to the clients and terminates. Figure 2.18 shows the overview of GRAM.

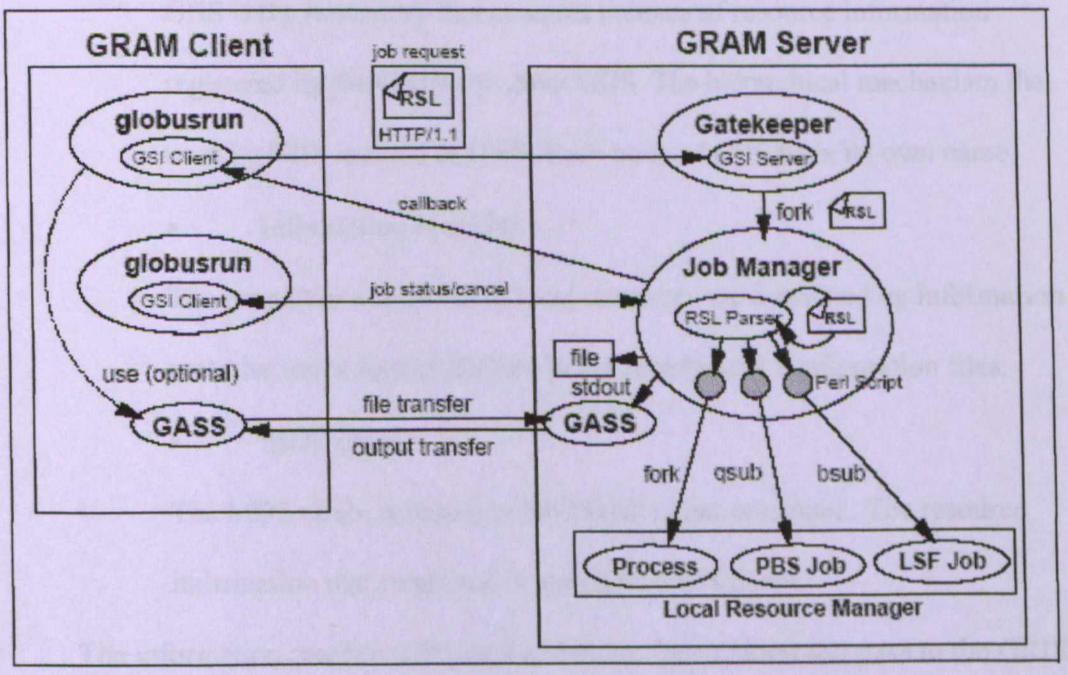


FIGURE 2.18 Overview of GRAM

➢ Monitoring and Discovery Service (MDS)

MDS provides access to static and dynamic information of resources. The following components are contained by MDS.

- Grid Resource Information Service (GRIS)

GRIS is the repository of local resource information which derived from information providers. Although GRIS is able to register its information with a GIIS, GRIS itself does not receive registration requests. The local information cached for a period of time which known as the time-to-live (TTL) when the information is requested to update. If GRIS does not receive request for its information, the information will be time out and deleted.

- Grid Index information Service (GIIS)

GIIS is the repository that contains indexes of resource information registered by the GRIS and other GIIS. The hierarchical mechanism that used by GIIS is same as DNS. Each node of GIIS have its own name.

- Information Provider

The properties and status of local resources are translated by information provider to the format defined in the schema and configuration files.

- MDS client

The MDS client is based on the LDAP client command. The resource information that requested is search by MDS client.

The information provider obtains the resource information and pass to the GRIS. The local information that managed by the GRIS need to register with the GIIS. MDS client get the resource information directly from GRIS and/or GIIS.

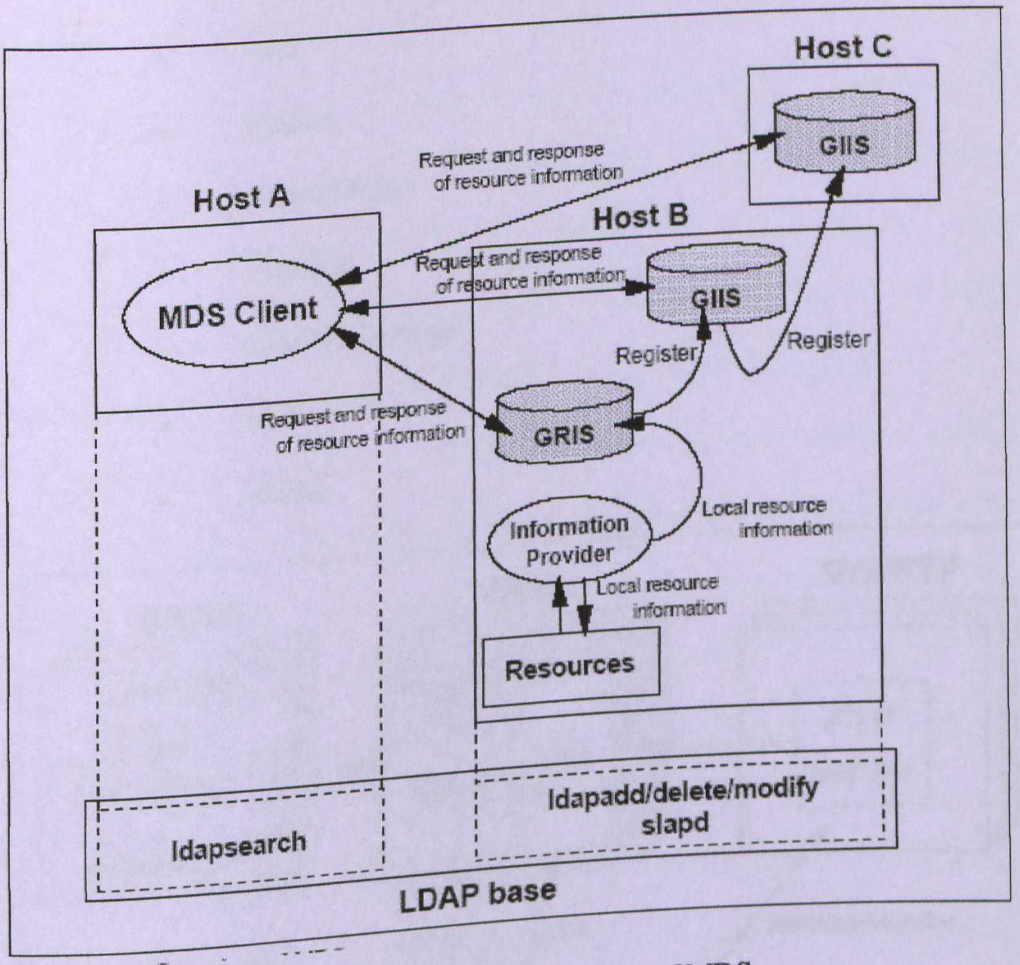

Host C
GIIS
Request and response of resource information
Host A
Host B
Request and response of resource information
MDS Client
GIIS
Register
Register
Request and response of resource information
GRIS
Local resource information
Information Provider
Local resource information
Resources
ldapadd/delete/modify
slapd
ldapsearch
LDAP base

FIGURE2.19 Overview of MDS

- GridFTP

A secure and reliable data transfer among the grid nodes is provided by GridFTP. Grid FTP is a protocol which based on FTP extends the standard protocol with facilitates with multistreamed transfer, auto-tuning and Globus based security. This protocol still not completely defined.

- API and software development's kit

These two components help developers to develop Globus related grid applications. APIs are implemented in the C language. Development's kit provides a rapid development kit which supports the following technologies:

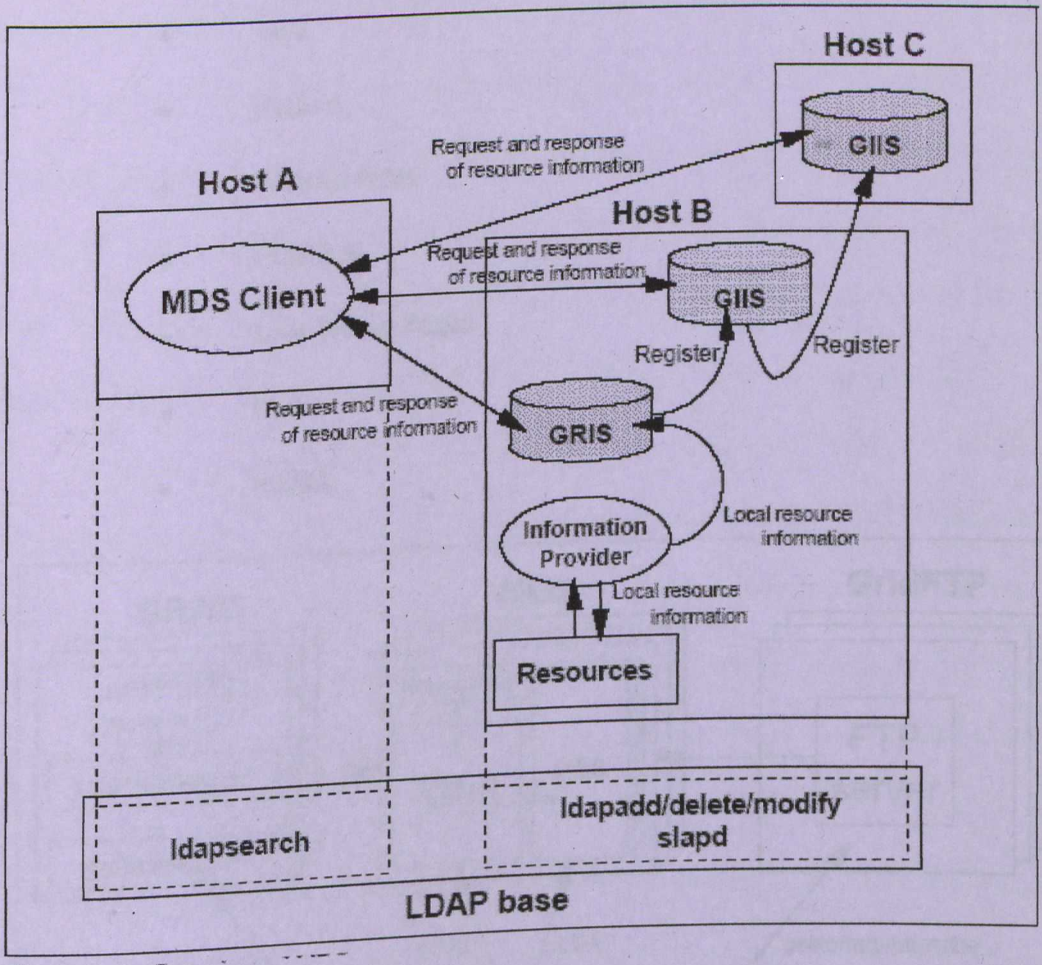

Host C
GIIS
Request and response of resource information
Host A
Host B
Request and response of resource information
MDS Client
GIIS
Register
Register
Request and response of resource information
GRIS
Information Provider
Local resource information
Local resource information
Resources
ldapadd/delete/modify
slapd
ldapsearch
LDAP base

FIGURE2.19 Overview of MDS

- GridFTP

A secure and reliable data transfer among the grid nodes is provided by GridFTP. Grid FTP is a protocol which based on FTP extends the standard protocol with facilitates with multistreamed transfer, auto-tuning and Globus based security. This protocol still not completely defined.

- API and software development's kit

These two components help developers to develop Globus related grid applications. APIs are implemented in the C language. Development's kit provides a rapid development kit which supports the following technologies:

- Java
- Python
- Web services
- COBRA
- Java Server Pages
- Perl
- matlab

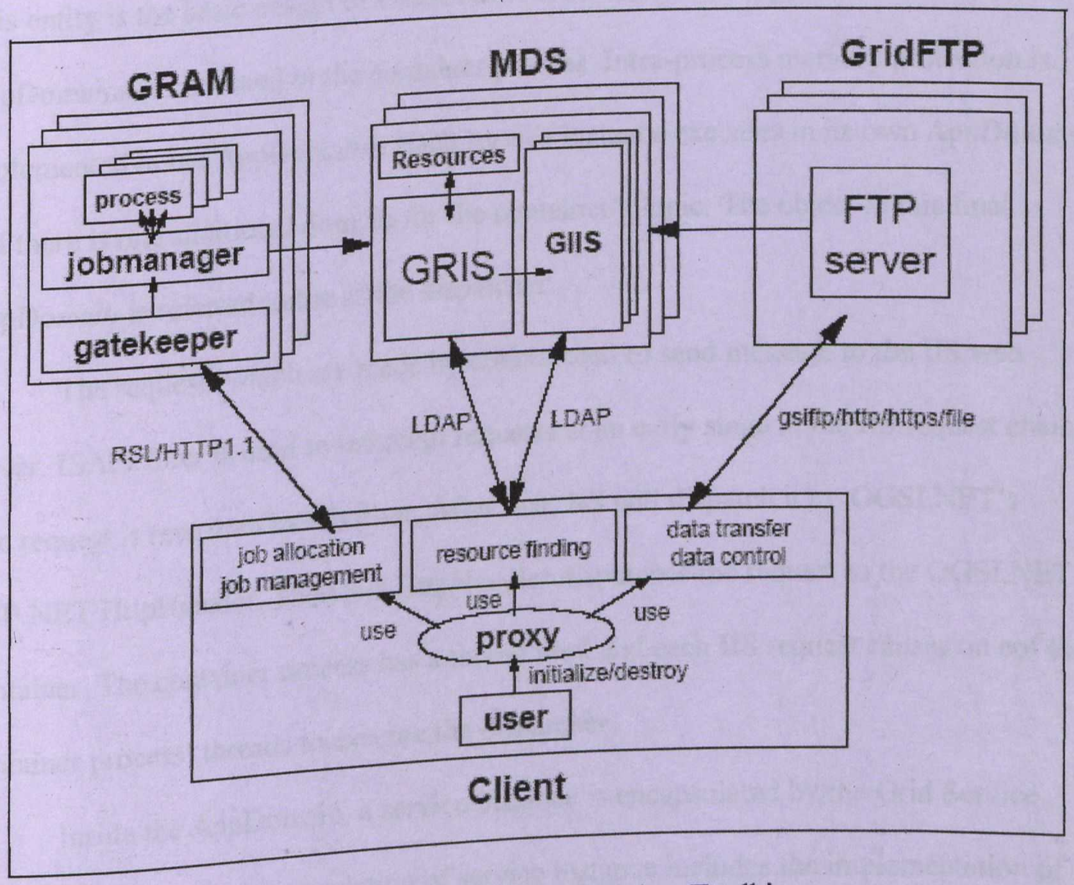


FIGURE 2.20 Overview of Globus Toolkit.

2.14.2 OGSI.NET

OGSI.NET is a middleware that used to support web services. It supports .NET framework. There are two goals for OGSI.NET. The first goal is to support the dynamic creation of stateful grid service instance that persist between client invocations. Then the clients' requests which use IIS to receive is the second goal of OGSI.NET. The IIS is used because to take advantage of the close integration of Microsoft software.

All the service instances which run on a host are hold by the containers entity. This entity is the basic design of OGSI.NET. A collection of ApplicationDomains or AppDomains is consisted in the container process. Intra-process memory protection is implemented in the AppDomains. Each service instance executes in its own AppDomain and there is one additional domain for the container's logic. The object in this final AppDomain is referred to the as the dispatcher.

The requests which are made by clients need to send message to the IIS web server. ISAPI filter is used to intercept requests at an early stage in the IIS request chain. The request is rewritten by the filter. After that, IIS will dispatch it to OGSI.NET's ASP.NET HttpHandler. Then this HttpHandler dispatches the request to the OGSI.NET container. The container process has a thread pool and each IIS request causes on eof the container process' threads to execute the dispatcher.

Inside the AppDomain, a service instance is encapsulated by the Grid Service Wrapper (GSW). The encapsulation of service instance includes the implementation of its methods and its service data.

2.14.2.1 SYSTEM ELEMENTS

To execute an grid application, the OGSI.ET provide some components. The components of OGSI.NET are:

➢ Dispatcher

The interface between the client request and the service instance which serves that request is called dispatcher. To route request messages to appropriate service instance and return results to the client is the main function of the dispatcher. The dispatcher contains a mapping from the request GSRs to AppDomains within the container the dispatcher does not process the incoming or outgoing messages.

➢ Grid Service Wrapper

The various functional units of a grid service instance are encapsulated by the Grid Service Wrapper (GSW). The GSW provides

- Pluggable, service-specific message serializers and deserializers
- Pluggable support for the port types supported by a service
- An SDE management API

➢ Light-Weight Service Wrapper

Some server-side services have limited functionality. These services are often only used in concert with some number of the other services and the protection of having these simple services in their own AppDomain is out-weighted by the overhead of constantly communicating across the AppDomain boundary. This service is called light-weight service. Light-weight is used by OGSI.NET for notification sinks for both server side and client side. Besides that, it also implement in the certain simple functionality or service.

- Factories

The instances of other services are created by factories. New AppDomain and a new Grid Service Wrapper are created in the service instance. A factory service stores a reference to the GSW in the new domain along with the published name of that instance. The dispatcher will receive this mapping.

- Message handlers

Message handlers perform message format specific processing on a service instance's incoming and outgoing messages. SOAP and remoting message formats are the two message handlers in the OGSI.NET. The request message is deserialized by message handlers from the dispatcher. When the request is completed or an exception is thrown, the handler will serialize the results into a byte stream which is passed to the dispatcher to be returned to the client.

- Factories

The instances of other services are created by factories. New AppDomain and a new Grid Service Wrapper are created in the service instance. A factory service stores a reference to the GSW in the new domain along with the published name of that instance. The dispatcher will receive this mapping.

- Message handlers

Message handlers perform message format specific processing on a service instance's incoming and outgoing messages. SOAP and remoting message formats are the two message handlers in the OGSI.NET. The request message is deserialized by message handlers from the dispatcher. When the request is completed or an exception is thrown, the handler will serialize the results into a byte stream which is passed to the dispatcher to be returned to the client.

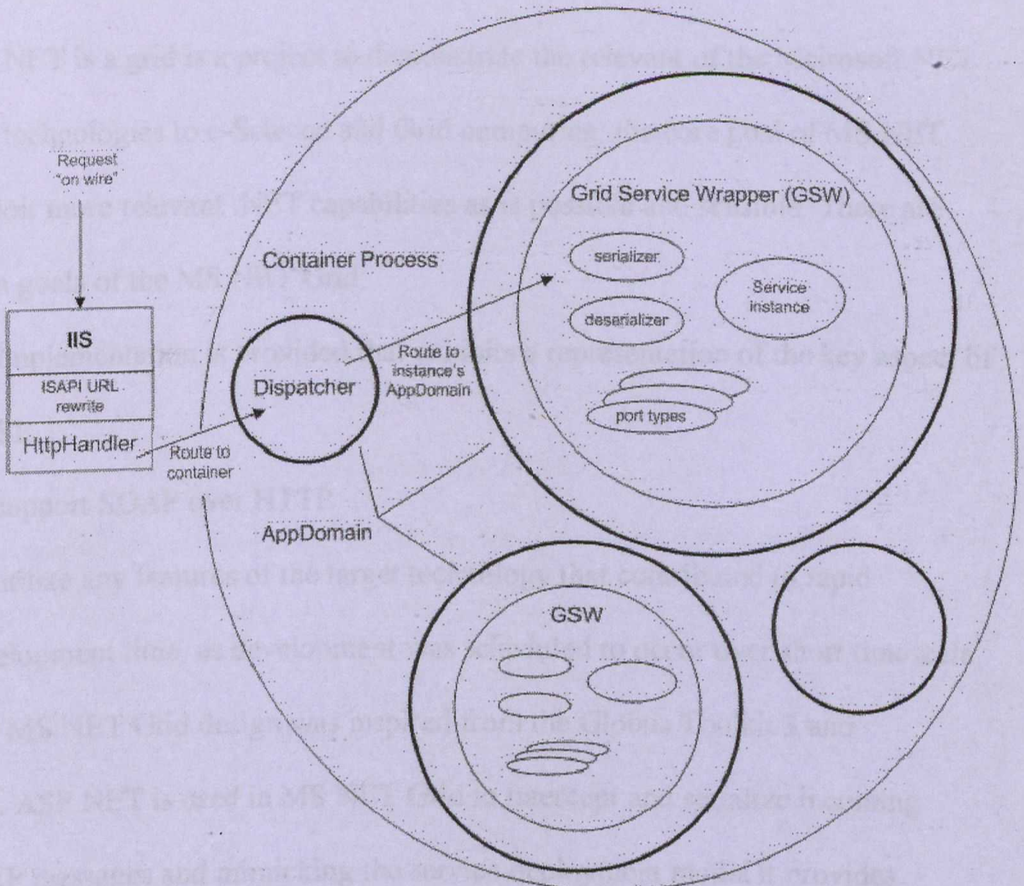


FIGURE2.21 Overview of OGSI.NET Components

2.14.2.2 SECURITY

A standards-based message layer security is provided in the OGSI.NET environment. The Web Service Extensions (WSE) pipeline is used to implement this message layer security. WSE run by the message handler in the Grid Service Wrapper. The memory protection of a process is provided by the AppDomains. There are no heavyweight activities of creating a new process for each service.

2.14.3 MS.NET GRID

MS.NET is a grid is a project to demonstrate the relevant of the Mcirosoft.NET suite of the technologies to e-Science and Grid computing. the core goal of MS.NET Grid is exploit more relevant .NET capabilities as is possible and sensible. There are other design goals of the MS.NET Grid:

- An implementation is provided that exhibits a representation of the key aspect of OGSI.
- To support SOAP over HTTP
- To utilize any features of the target technology that contributed to rapid development time, as development was scheduled to occur over short timescale.

The MS.NET Grid design was inspired from the Globus Toolkit 3 and OGSI.NET. ASP.NET is used in MS.NET Grid to intercept and serialize incoming HTTP/SOAP messages and mimicking the service deployment model it provides.

2.14.3.1 MS.NET GRID ARCHITECTURE

The OGSI container is implemented as an ASP.NET web application. Within this application are components for dealing with service instance management, managing communication with clients, providing OGSI portType-related functionality and allowing developers to deploy services. The following figure is show the architecture of MS.NET Grid.

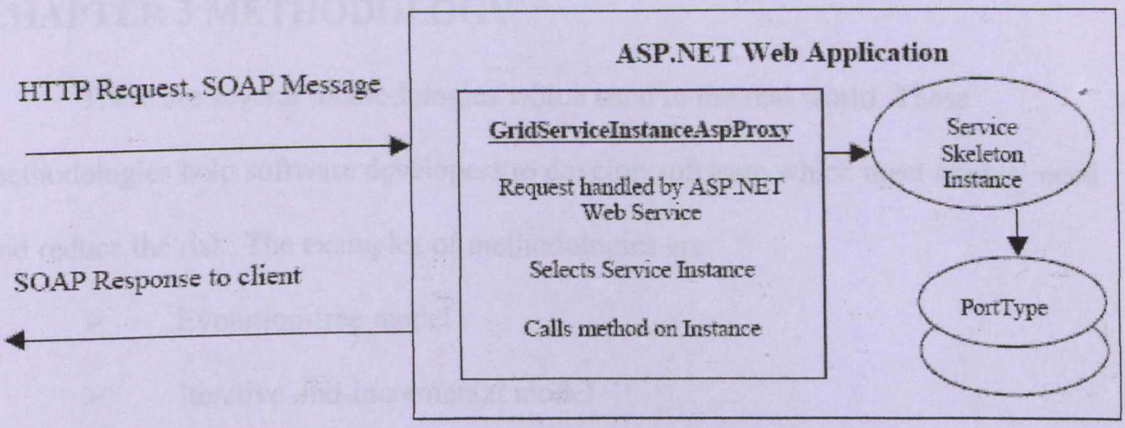

ASP.NET Web Application
HTTP Request, SOAP Message
GridServiceInstanceAspProxy
Request handled by ASP.NET Web Service
Selects Service Instance
Calls method on Instance
SOAP Response to client
Service Skeleton Instance
PortType

FIGURE 2.22 Architecture of MS.NET Grid

Two type of service lifetime are allowed to use in the OGSI container. First, the service lifetime applies to the service instances which are spawned by other services. This type of service lifetime is called transient service lifetime. The service instance lives until its termination time has passed. The second type of service lifetime is persistent lifetime the service instances is initiate when the container starts up and lives as long as the containing web application..

CHAPTER 3 METHODOLOGY

There are several methodologies which used in the real world. These methodologies help software developers to develop software which meet clients' need and reduce the risk. The examples of methodologies are:

- Evolution-tree model
- Iterative-and-incremental model
- Code-and-fix model
- Waterfall model
- Rapid-prototyping model
- Extreme programming
- Synchronize-and-stabilize
- Spiral model

All these methodologies have their strengths and weaknesses. Software developers should consider about the strengths and weaknesses of these methodologies before decide to use which methodologies in their software development project.

3.1 SOFTWARE DEVELOPMENT

The software development starts from scratch. The clients' requirements are determined from the requirement phase. The developers start to gather client's need. Analysis phase analyze client's requirement. When analysis phase is complete, the Design phases start. A design specification is produced. This is followed by implementation of complete software product.

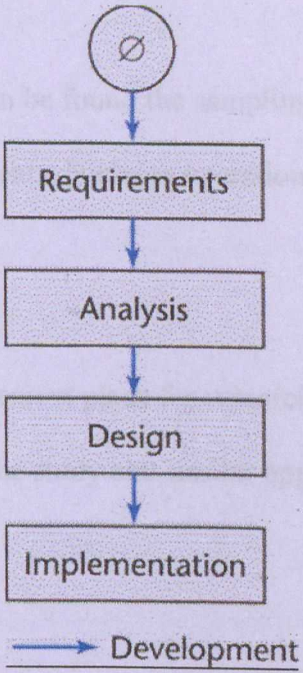


FIGURE 3.1 Software Development

3.2 FACT FINDING

Fact finding technique is used to gather more information for the purpose of develop software especially in the requirement phase. There are several technique are used. Interview, sampling, research, observation, questionnaires and discovery prototype are the example of fact finding.

3.2.1 INTERVIEW

The most common used fact finding technique is interview. First, the developers need to get the permission from the management to have an interview. Preparation that needs to do is create a list of topic that to be covered. Open-ended questions are preferred because this type of question can gather more clients' opinions.

3.2.2 SAMPLING

Many facts and details can be found the sampling of existing documents. The analysts collect historical documents, business operations manuals and forms, and information systems documents.

3.2.3 RESEARCH

Internet is the more convenient place for research. Research is an often-overlooked technique based on the study and similar applications. Site visit is a special form of research.

3.2.4 OBSERVATION

The analysts go to the client's working place to study people doing their jobs.

3.2.5 QUESTIONNAIRES

A larger number of questionnaires are distributed to people. The analysts collect back the questionnaires and analyzing them.

3.2.6 DISCOVERY PROTOTYPING

This method always used by development team when they have problem in defining system requirements. It is important that the prototype can be developed quickly. Hence, the prototype can be used during the development process.

3.3 ITERATION AND INCREMENTATION MODEL

According to Miller's Law, humans are enabling to concentrate on only approximately seven units of information which is also called chunks. However, the artifacts in the software development are more than seven chunks. Hence, the artifacts should be given priorities to measures the importance of them. Stepwise refinement is introduced. Software developers concentrate the important requirements and the less critical requirements will be postponed. After solving the important requirements, the less critical requirements will be consider.

There are five core workflows which are requirements workflow, analysis workflow, design workflow, implementation workflow and test workflow. All the five workflow are performed over the life cycle of a software product. However, a workflow will predominates over the other four workflows at a time.

There are two activities which are performed throughout the life cycle of the software product. Planning and documentation are the two activities. Besides that, testing is an activity that is performed during each of iteration.

Iteration and incrementation model can be used to solve the moving-target problem. This model correct the inevitable mistakes made while a software product is being developed.

The iterative-and-incremental model has much strength:

- This model offers multiple opportunities for developers to check software product. The test workflow is incorporates with every interation.
- The robustness of the underlying architecture can be determined relatively early in the life cycle. Various component artifacts are included in the architecture of a software product. When an iterative-and-incremental model is used, it soon

apparent whether the or not the architecture is robust. If it is clear that the architecture is not robust, the client needs to decide whether to abandon the project or start again from scratch.

- This model can mitigate risks early. Risks are invariably involved in software development and maintenance.

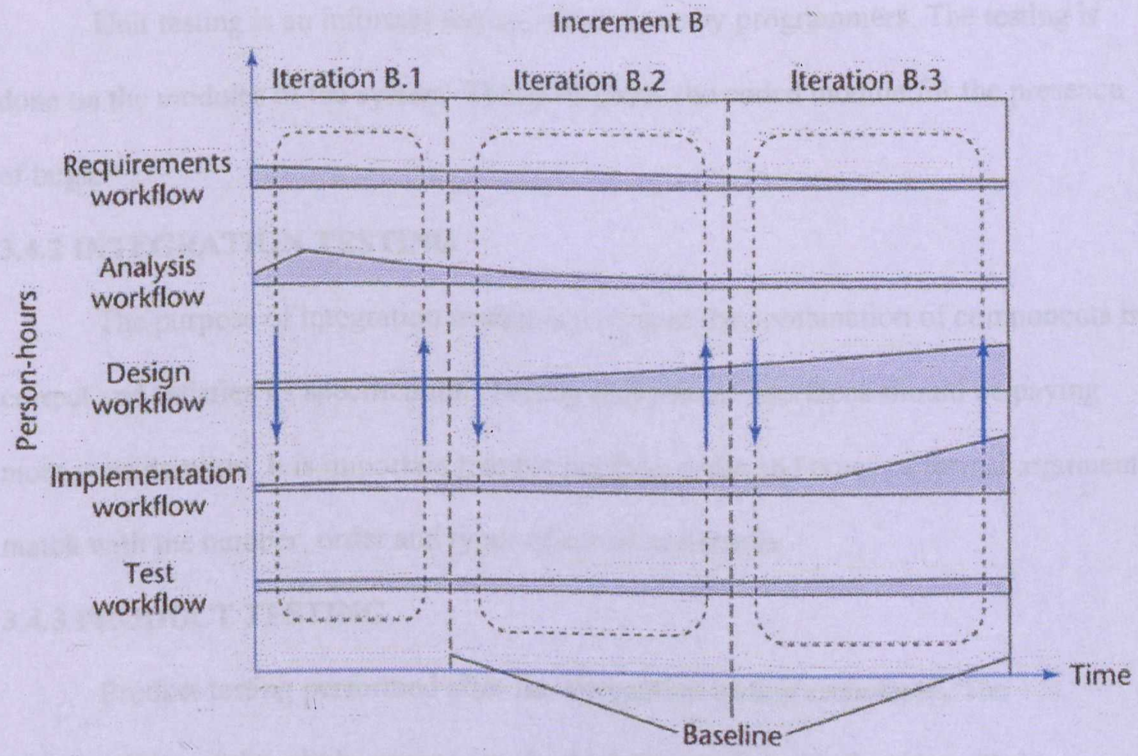


FIGURE3.2 Iteration and Incrementation

3.4 TESTING

Testing is an important phase in the software development process. This phase is carried out with other phases in the evolution-tree model from the beginning. Testing ensure the software that developed do not have problem. There are several types of testing.

3.4.1 UNIT TESTING

Unit testing is an informal testing which done by programmers. The testing is done on the modules of the system. This is to check the coded module for the presence of bugs.

3.4.2 INTEGRATION TESTING

The purpose of integration testing is to ensure the combination of components is correct and satisfies its specification. Testing component interfaces should be paying more consideration. It is important that the number, order and types of formal arguments match with the number, order and types of actual arguments.

3.4.3 PRODUCT TESTING

Product testing performed after the integration testing completed. The functionalities of the whole system are checked against the specifications. The constraints that listed in specification must be tested. The purpose of product testing is to ensure the specifications are correctly implemented.

3.4.4 ACCEPTANCE TESTING

Acceptance testing is performed when the software is delivered to clients. Clients use the actual data to check the functionalities of software. When clients satisfy the software which meet the specifications, the acceptance testing is complete.

CHAPTER 4 ANALYSIS

4.1 INTRODUCTION TO .NET FRAMEWORK

.NET framework is a development and execution environment which introduced by Microsoft Corporation. This framework allows different programming languages and libraries to work together. Window-based applications that created are easier to build, manage, deploy, and integrate with other networked systems. The .Net framework spans clients, servers and developer tools, and consists of:

- The .NET framework 1.1 is used for building and running all kinds of software. The software is including web-based applications, smart client applications, and XML web services.
- Developer tools provide an integrated development environment (IDE). The aim is to maximize developer productivity with .Net framework.
- A set of servers is integrates, runs, operates and manages web services and web-based applications. The example of servers is Microsoft Windows® Server 2003, Microsoft SQL Server™, and Microsoft BizTalk® Server.
- Client software helps developers to deliver a deep and compelling user experience across a family of devices and existing products. Windows XP, Windows CE and Microsoft Office XP are the examples of client software.

The .Net framework consists of the following features:

- The Common Language Runtime (CLR)

  Services which help manage application execution are provided by a language-neutral and execution environment

- The Framework Class Libraries (FCL)

A consistent, object-oriented library of prepackaged functionality

Basic infrastructures that develop Windows-based applications are provided by the .Net framework. This framework makes Microsoft's vision of connecting information, people, systems and devices. The following characteristics of .Net framework to realize Microsoft's .Net framework:

- Support for standard networking protocols and specifications

  Use standard Internet protocols and specifications like TCP/IP, SOAP, XML and HTTP. .Net framework allows a broad range of information, people, systems and devices to be connected.

- Support for different programming languages

  Support a variety of different programming languages. Developers can pick the suitable programming language to do their project.

- Support for libraries developed in different languages

  Provide a consistent programming model for using prepackaged units of functionalities. These functionalities make application development faster, easier and cheaper.

- Support for different platforms

  The .Net framework is available for a variety of Windows platforms.

4.2 PROGRAMMING LANGUAGES

There are several programming languages approaches which suitable to develop a grid application. Visual C#, Java and C++ are the suitable programming languages which are mentioned in chapter 2. At this chapter, Visual C# is chosen to develop a grid application.

4.2.1 VISUAL C# .NET

Visual C# .NET builds on a strong C++ heritage. Hence, C++ and Java developers can immediately familiar to C#. Some improvements are offered in C# programming language. The improvements include a unified type system, "unsafe" code for maximum developer control and powerful new language constructs easily understood by most developers.

Garbage collection is an important feature in C#. C# has a built-in garbage collection. Garbage collection provides the following advantage to developers:

- No memory leaks. All objects at some point are disposed during program execution. The garbage collection do not guarantee when will dispose objects.
- Data sharing is easier if garbage collections exist. Some applications which are developed need object sharing.
- Developers can more concentrate on program logic and correctness because programs should automatically become more "correct". Memory management is less important than the program logic at this level.

Source code documentation can be produced by C# developers with using XML comments. Developers reuse their code from within any programming language that supports .NET.

Besides that, the programming language that suggested in the middleware also is C#. The services which contained in the middleware need to be called out by the users need to written in the C# programming language.

Visual C# has several advantages compare with Java. The advantages of Visual C# are:

- Besides directly used in C#, libraries can be built in J#, Visual Basic .Net and managed C++ by using Visual Studio .Net.
- More keywords can be reserved in C#.. More additional built-in features are provided in C# which compared with Java.
- Operator overloading which allows developers to write code that seems natural to them is omitted in Java. This is an important features in
- The code can run faster in C# than Java. This is because methods in Java must always be treated as virtual. The compiler need to determine which actual function should be called at compile-time.
- There is an implied communication between a *base* class designer, and developers building subclasses. Hence, C# is good in control.

Besides that, the newest, faster-growing developer community participated by C# developers. Developers can join community to exchange code and resources, leverage skills across and multiple computing environments. When developers face some problems in software development, the developers can find some reference code and resources from community.

4.3 OPERATING SYSTEM PLATFORM

The Grid system needs to operate in the heterogeneous environments. Hence, Linux and Windows are chosen to realize these requirements. The Windows 2000 and windows XP are currently installed in the computers which in the faculty. Some computers in the research lab of faculty are already be installed the Linux.

There are some advantages of Windows platforms:

- More user friendly compare with UNIX

  The Graphical User Interface (GUI) helps to manage the system easily. Users can do a single process or instruction within a single click on the icon.

- Built-in web server

  Internet Information Server (IIS) is a built-in web server in Windows platform which are Windows 2000 Pro and Windows XP. Users do not need to download other web server from internet. Besides that, IIS can be updated with the latest version from internet.

Besides that, there are advantages which encourage the users commonly use Linux. The advantages of Linux are listed below:

- Open-source operating system

  Users can install or upgrade Linux software for free. They just need download it from Internet. It is more economical than Windows platforms which need to buy licenses.

- Reliability

The Linux can run for months without reboot the machine. The bugs which are found in the software can be solve quickly because Linux is open-source.

4.4 DATABASE MANAGEMENT SYSTEM

Several database management systems are commonly used in the markets. MySQL and Microsoft SQL Server are chosen to implement the Grid application. To implement database in the Linux environment, MySQL is used. This is caused by the MySQL is an open-source database management system which can operate on the Linux, UNIX and Windows platform.

While Microsoft SQL server is a Microsoft product, it provides a user-friendly user interface. Besides that, it is easy to integrate in the .Net Framework.

4.5 FUNCTIONAL REQUIREMENT

An action which is needed to perform by the target product is specified in the functional requirement. There are seven functional requirements. The functional requirements are:

- Search

  Search module allow user search their interested resources based on the keywords. The search queries will pass to the databases in parallel. After the databases perform the operation which request by the queries, the result will display to the users. Users can retrieve the resources by click the link.

- Log-in

  Users need to log-in before perform any operation that offer by the data grid portal. A username and password need to be keyed in by users. The authentication server will authenticate users. New users need to register first to gain the authority to access the data grid portal.

- Upload

  Users can upload some resources to the selected database. Besides that, users can select the option that ensures that the resource shared by public or just can view by the users only.

- User profile

  User's personal information is display in this module. Besides that, the 5 search hits which are search by users are display to users.

- Email

  An email module provides a way that let users send enquiries and feedbacks.

- Help

  Help module provide user manual to help users familiar with the functions of the data grid portal.

4.6 NON-FUNCTIONAL REQUIREMENT

Non-functional requirements are defined as constraints which describes the restrictions on the system. The system is expected to meet certain standards to improve it perform in the delivery. The non-functional requirements that need to achieve in this project are:

- User-friendly interface

  A user-friendly interface can reduce the learning curves of the users especially the users do not have any computer knowledge. A meaningful icon or other captions will be helpful to the users when they interact with the system.

- Accuracy

  The system is expected can perform a precise calculation when a user submits a job. The logic which used in the system should be correct and free from computation errors.

- Reasonable response time

  A fast response time means that the time to retrieve information within a reasonable interval time.

- Reliable performance

  The performance of a system is important to the clients. The system should work consistently. Besides that, any serious failures should not be happens in this system.

- Modularity

  A system is separated into different modules which haves distinct functionality. The developers easy to test, expand or maintain the system.

- Reusability, expandability and maintainability

  The coding of systems should be flexible. The system can be reused for the purpose of version upgrade.

- Security

  Security is an important part in the grid application. A moderate level of security protection and access control should be implemented in the system.

CHAPTER 5 SYSTEM DESIGN

There are three activities in the design workflow. Architectural design, detailed design and design testing are the three activities. A specification document which is description of the product's functionalities is the input for the design. The output is the design document which is a description of how to realize the functionalities of product.

Architectural design is also known as general design, logical design or high-level design. The whole product is broken into the smaller components. The components of the product are called modules. The specification is analyzed and the functionality is produced in the module structure.

The process that designs each of the modules in detailed is called detailed design. The detailed design is known as modular design, physical design or low-level design. The processes which interconnected the modules to form a complete product is not emphasized in the detailed design.

The third activity in the design workflow is the design testing. Testing is not the activity that performs after the architecture design or detailed design is completed.

5.1 SYSTEM FUNCTIONALITY DESIGN

5.1.1 CONTEXT DIAGRAM

Context diagram is an important part of the Software Engineering. The whole system is shown in the context diagram. The inputs and outputs of the system with the external system is shown in the context diagram. Figure 5.1 shows then context diagram.

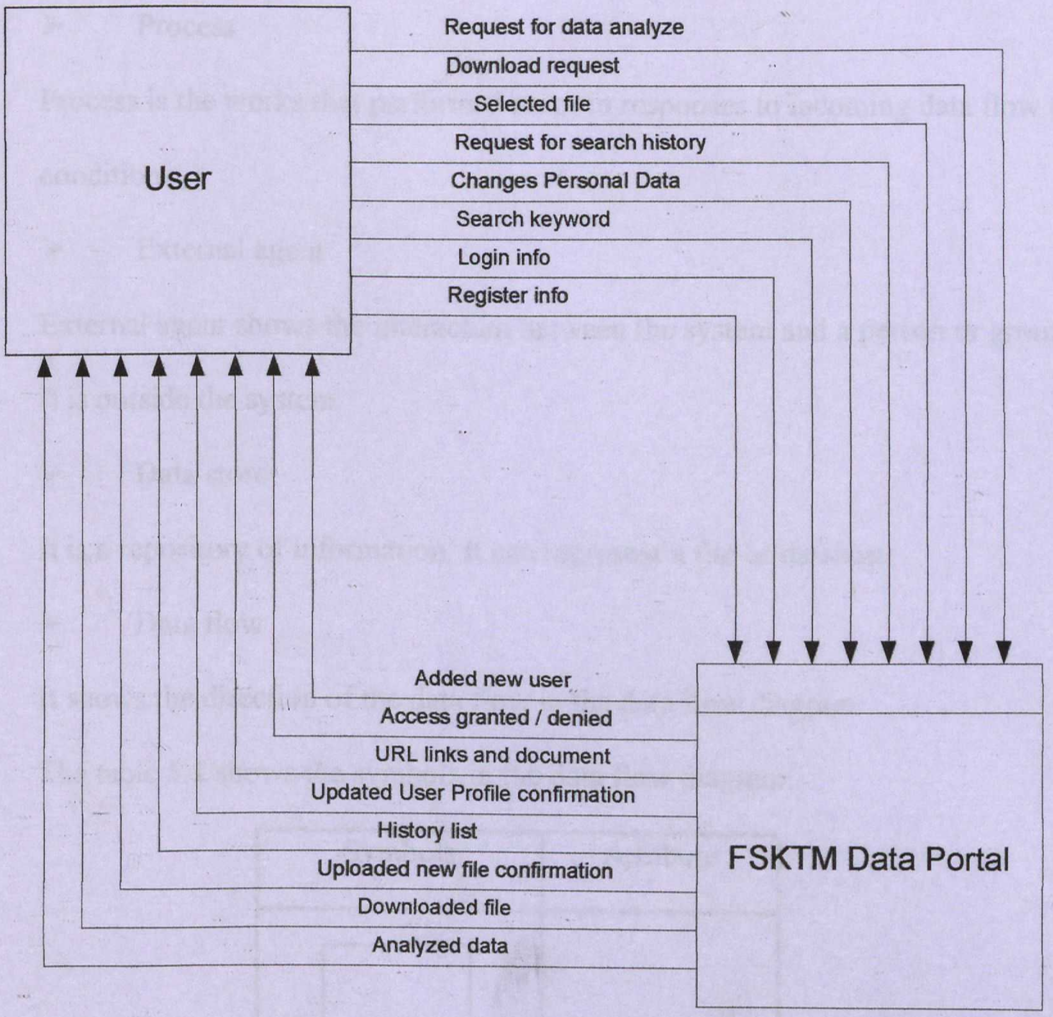


FIGURE 5.1 Context Diagram

5.1.2 DATA FLOW DIAGRAM (DFD)

The flow of data or information can be shown in the data flow diagram. Data flow diagrams can be grouped together or decomposed into multiple processes. Analysts use data flow diagrams to model processes and functional requirements. This is a primary tool that developed in the 1970s.

There are some descriptions for the symbols that used in the data flow diagrams.

➢ Process

Process is the works that performed on or in responses to incoming data flow or conditions.

➢ External agent

External agent shows the interaction between the system and a person or group. It is outside the system.

➢ Data store

It is a repository of information. It can represent a file or database.

➢ Data flow

It shows the direction of the data flow in the data flow diagram.

The table 5.1 shows the symbols in the data flow diagram.

Symbols	Attribute
□	Entity
→	Flow of Data

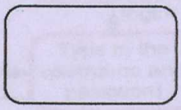
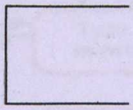
	Process
	Data Store

TABLE 5.1 Symbol of Data Flow Diagram

Figure 5.2 shows the data flow diagram of the system.

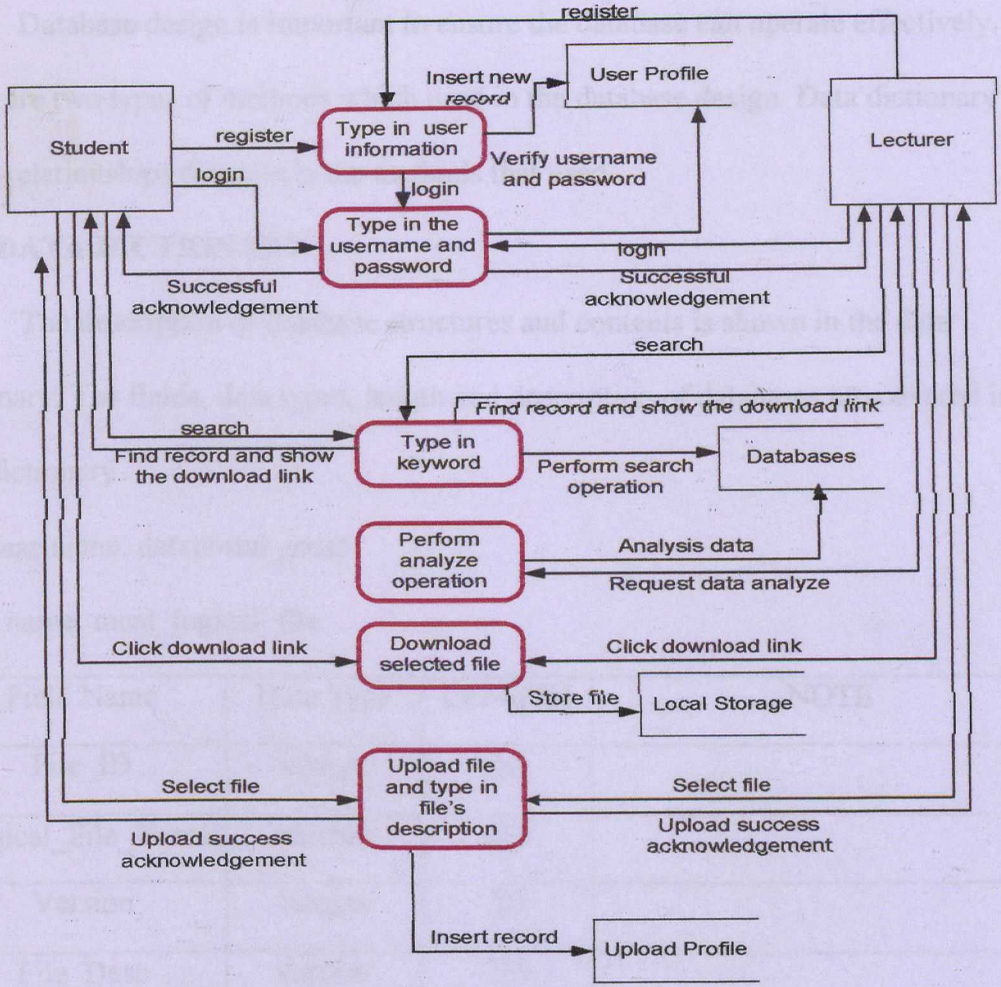


FIGURE 5.2 Data Flow Diagram

5.2 DATABASE DESIGN

Database design is important to ensure the database can operate effectively. There are two types of methods which used in the database design. Data dictionary and entity-relationships diagram is the methods that used.

5.2.1 DATA DICTIONARY

The description of database structures and contents is shown in the data dictionary. The fields, data types, length and description of databases are covered in the data dictionary.

Database name: dataportal_mcat

Table name: mcat_logical_file

Field Name	Data Type	LENGTH	NOTE
File_ID	integer	10	.
Logical_File_Name	varchar	255	
Version	integer	20	
File_Desc	varchar	255	
File_Size	double	10	
Physical_Path	varchar	255	
Collection_ID	integer	10	
Container_Service	varchar	50	
Is_Valid	integer	10	
Create_By	varchar	50	Name of owner
Create_Date	varchar	50	
Ip_Address	varchar	50	IP address of machine which the file located

TABLE 5.2 Table of mcat_logical_file

Table name: indexing

Field Name	Data Type	length	Note
Index_Word	varchar	50	
File_ID	varchar	255	
File_Name	varchar	255	
File_Desc	varchar	255	

TABLE 5.3 Table of indexing

5.2.2 ENTITY-RELATIONSHIP DIAGRAM (ER DIAGRAM)

There are three types of relationships in the ER diagram. One: one (1:1), one: many (1 :*) and many: many (* : *) are the relationships. The ER diagram is illustrated in the Figure 5.3.

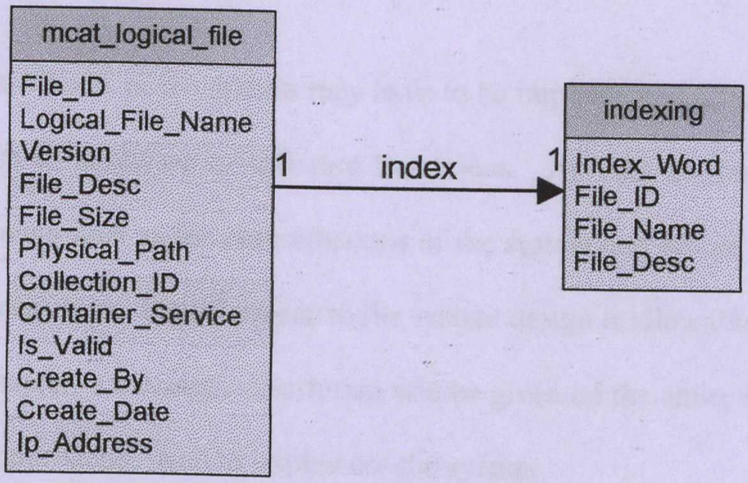


FIGURE5.3 ER Diagram

CHAPTER 6 IMPLEMENTATION

6.1 INTRODUCTION

In the system implementation phase, the major work includes coding and debugging which are needed to solved the problems and errors found during the coding stage. Programming language characteristics and coding style can profoundly affect the system quality maintainability.

Coding step translates a detail design representation of software into a programming language realization. In a software project, the requirements analysis, system design and implementation phases do not have a clear boundary. Each phase tend to overlap one another. This phase at time involves some modifications to the previous design.

Certain attributes of the system may have to be implemented differently from the original plan due to hardware or software limitations. As long as the system fulfills critical requirements such as the core functions of the system and do not violate the pre-determined business rules, modifications to the system design is allowable.

In this chapter, a thorough description will be given on the entire implementation process and the approaches used to implement the system.

6.2 DEVELOPMENT ENVIRONMENT

The development environment depicted here are the settings, conditions and surroundings where the actual system development takes place. It is important to take note of the environment in which the development was carried out because it may have effect on the speed and effectiveness of the system's implementation.

6.2.1 SOFTWARE REQUIREMENT

Table 6.1 shows the software specifications that were needed during the system's implementation. They are essential towards the development of the system, e.g. IIS 6.0 must be installed as the web server because the system utilized a web application as the interface for the user to retrieve and view information from the database. The Microsoft Internet Information Services 5.1 is one of the Microsoft products. Hence, it is compatible in the .NET environment. Furthermore, since the development is carried out in a .NET environment, thus the Microsoft .NET Framework must be installed before system implementation can be carried out.

In the implementation phase, the core development tool is the Microsoft Visual Studio 2003. All the code-behind coding is implemented by using this tool. Besides that, Dreamweaver MX is used to develop the HTML coding and design the GUI interface.

Table 6.1 Software Specification for the Implementation Process

Category	**Description**
Operating System	Windows XP
Web Server	Microsoft Internet Information Services 5.1
Special Requirements	Microsoft .NET Framework 1.1
Development Tools	Microsoft Visual Studio.NET 2003
HTML Coding	Dreamweaver MX

6.3 APPROACHES TO THE DEVELOPMENT OF THE SYSTEM

There are some methods or methodologies to implement the system design into the actual coding. A review of the coding methodologies, convention and the best practices for the developing the system is showed in the following sections.

6.3.1 CODING METHOLOGIES

A study to the different approaches which can be used to develop the system is needed before starting the development of the system's implementations. There are two types of methodologies which are used to develop the systems. The two different methodologies are top down approach and bottom up approach.

6.3.1.1 TOP DOWN CODING METHODOLY

Top-down programming refers to a style of programming where an application is constructed starting with a high-level description of what it is supposed to do, and breaking the specification down into simpler modules, until a level has been reached that corresponds to the primitives of the programming language to be used.

Top-down programming tends to generate modules that are based on functionality. Typically, the high-level specification of the system states functionality. This high-level description is then refined to be a sequence or a loop of simpler functions or procedures, that are then themselves refined, etc.

Among the advantages of top-down implementations are:

i. Avoiding chaos of attempting to design the system all at once. Planning and implementing can be incredibly complex. Attempting to get all subsystems in place and running at once is basically planning to fail.

ii. Enables separate development works to be done in parallel on different but necessary subsystems that can save a great deal of time.

iii. Prevents the developer from getting mired in details that might cause him/her to lose sight of why the system is developed in the first place.

Although top-down implementations have a lot fo advantages, this methodology still have its weaknesses or disadvantages. There are:

i. All decisions made from the start of the project depend directly or indirectly on the high-level specification of the application. However, specification tends to change over time. When that happens, there is a great risk that large parts of the application need to be rewritten.

ii. There is a chance that the system will be wrongly divided into unsuitable subsystems. Attention must be paid to overlapping needs and sharing of resources so that the partitioning of subsystems makes sense for the system.

iii. Modules generated are very specific to the application that is being written, thus not very reusable

6.3.1.2 BOTTOM UP CODING METHODOLOGY

An application is constructed starting with existing primitives of the programming language and constructing gradually more and more complicated features, until all of the application has been constructed. This types of coding methodology is bottom-up coding methodology..

In a language such as C or Java, bottom-up programming takes the form of constructing abstract data types from primitives of the language or from existing abstract data types. In Common Lisp, in addition to constructing abstract data types, it is common to build *functions* bottom-up from simpler functions, and to use macros to construct new *special forms* from simpler ones.

The advantages of bottom-up implementations are:

i. Testing is simplified since no stubs are needed. While it might be necessary to write test functions, these are simpler to write than stubs, and sometimes not necessary at all.

ii. Pieces of programs written bottom-up tend to be more general, and thus more reusable than pieces of programs written top-down.

iii. Critical functions can be coded first to test their efficiency.

At the other end, bottom-up implementation also has its weaknesses, as described below:

i. Development maybe somewhat directionless because overall program organization and higher level design is postponed.

ii. Developers may get too deep into the specific details of the different modules of the system and end up missing the deadline.

iii. The sub-modules of the system will still be eventually integrated.

6.3.1.3 THE CHOSEN METHODOLOGY

After reviewing the two approaches, the top-down coding methodology is used to develop and implement the system.

This decision is made partly due to the characteristics of the chosen programming language for development: C#.NET and ASP.NET which both advocates the usage of object oriented programming practice. Development in an object oriented environment practically envisions an abstract object at the top level before going down to the specifics.

Bottom up coding convention is more suitable for function driven programming languages like C and C++.

6.4 CODING IMPLEMENTATION

To implement the system design into actual coding, it is almost get started with the coding process. Before that, it is important to come up with a set of rules which should be abided by during the coding process, as described in the following sections.

6.4.1 CODING CONVENTION

The important attribute of the source codes is the coding convention. It determines the intelligibility of a program. Internal (source code level) documentation, methods of data declarations and approaches to statement constructions are the element of coding conventions.

Several coding conventions have been employed in the coding process to ensure system consistency, maintainability and readability. These coding conventions are as follow:

i. Use meaningful variable names, constant names, procedures names and parameter variable names to self document a program without excessive use of comments.

ii. Plan the layout of the program source code to improve its readability. Each sentence is started on a new line; statements following control structures are indented; white space is used to set off related blocks of codes, etc. These may seem insignificant but will help reduce the time needed to understand the program flow in the case where other programmers refer the code.

iii. All variables are declared at the beginning of procedures and declarations are separated from executables statements with a blank line to improve program readability.

iv. Insert comments to document the program and improve program understandability.

v. Group related types of codes together

6.4.2 CODING DOCUMENTATION

To begin the coding documentation, it should start with the selection of identifier names and then continue with the composition of the connecting all separate components. At the end, it should follow by the re-organization of the program.

Elements taken to consideration in coding for an easy to maintain and enhanced system are internal documentation, standard naming conventions and standard graphical user interfaces.

To provide a clear guide to programmer for future enhancements, comments are used in coding. The source code should embed with the statements of purpose indicating the functions of modules and descriptive comments. This is useful to describe the processing functions

Standard naming convention and also a standard usage of graphical user interface components is employed in developing the system. Standard naming conventions provides the programmers with easy identification of variables. A standard usage of graphical user interface components provides users an environment that will not generate much surprise to them.

6.4.3 CLASSIFICATION OF PROGRAM CODE

In this system, program coding is basically can be divided into a few sections. All these sections are divided based on the functions that performed. They are:

i. Database operations

This includes populating ASP.NET server controls with values, displaying data from database and basic data operations such as creating, updating and deleting records.

ii. Web Services

Web Services provide a group of functions that let the web applications to invoke.

iii. Windows Service

Windows Services run as agent in the client computer to let the web services to invoke for certain functions in this system.

6.4.3.1 DATABASE OPERATIONS

(i) Importing the required Class

Operations related to database for this system utilized the ADO.NET data access technology. Therefore, all database related operations must have the following statement:

```
using System.Data.SqlClient;
using System.Data.Odbc;
using System.Data.OleDb;.
```

(ii) Connecting to the Database

The database connection string is stored in the Web.Config file.

```
<add key="MySQLConnection" value="driver={MySQL ODBC 3.51
Driver};server=127.0.0.1;database=dataportal_mcat;uid=mca
t;pwd=mcat" />
<add key="SqlConnection"
value="server=(local);database=DataPortal;uid=portal;pwd=
portal" />
```

Therefore, any reference to the connection string is done like this:

```
String strConnection =
ConfigurationSettings.AppSettings["SqlConnection"];

SqlConnection objConnection = new
SqlConnection(strConnection);
```

Putting the connection string in the Web.Config file ensures that any future changes to the database (location, name, etc) will only need to be done to this file and not to the respective connection strings in all the different modules.

(iii) Database Operations

A standard database operation can utilize any of the ADO.NET components, including the Data Set, Data Adapter, Data Table, Data View, Data Column and Data Row. Below is a sample code that let user to change their password.

```
string strQuery = "update user_profile set
password="+"'"+password+"'" +"where
userName="+"'"+username+"'";

SqlCommand command = new SqlCommand(strQuery,connection);
command.ExecuteNonQuery();
connection.Close();
```

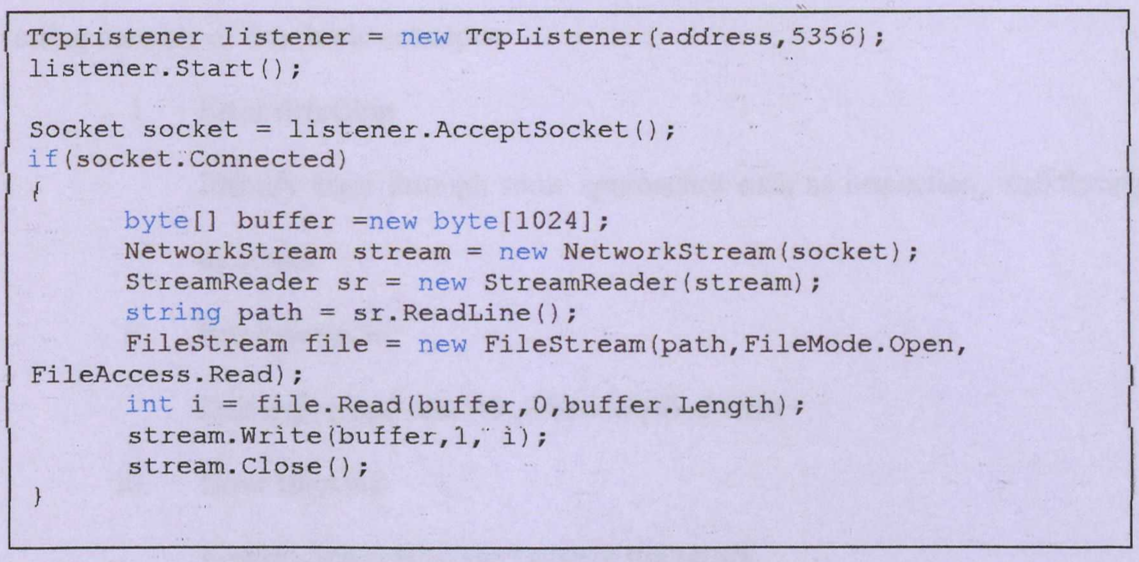
6.4.3.2 WEB SERVICES

The web services contain the core functions of this system. The web applications need to invoke the functions in the web services to executes functions in this system. Below is a sample for user to download a file.

```
byte[] buffer = new byte[1024];
IPAddress = ipAddress;
FileStorePath = path;
DownloadPath = "C:\\Inetpub\\wwwroot\\FSKTM Data Portal\\temp\\";
Size = size;
string message = "";
int total = 0;

TcpClient client = new TcpClient();
client.Connect(IPAddress,5356);

byte[] image = new byte[1024];
NetworkStream netStream = client.GetStream();
StreamWriter sw = new StreamWriter(netStream);
sw.WriteLine(FileStorePath);
sw.Flush();
sw.WriteLine(Size);
sw.Flush();

string filename = Path.GetFileName(FileStorePath);
DownloadPath = DownloadPath + filename;
FileStream fs = new
FileStream(DownloadPath,FileMode.OpenOrCreate,FileAccess.ReadWrite
);
while (total < Size && netStream.CanRead)
{
      int i = netStream.Read(image,0,image.Length);
      fs.Write(image,0,i);
      total = total + i;
}
netStream.Close();
fs.Close();
sw.Close();
```

6.4.3.3 WINDOWS SERVICES

Below is the sample code for user download a file by invoke the windows services.



```
TcpListener listener = new TcpListener(address,5356);
listener.Start();

Socket socket = listener.AcceptSocket();
if(socket.Connected)
{
      byte[] buffer =new byte[1024];
      NetworkStream stream = new NetworkStream(socket);
      StreamReader sr = new StreamReader(stream);
      string path = sr.ReadLine();
      FileStream file = new FileStream(path,FileMode.Open,
FileAccess.Read);
      int i = file.Read(buffer,0,buffer.Length);
      stream.Write(buffer,1, i);
      stream.Close();
}
```

CHAPTER 7 TESTING

7.1 INTRODUCTION

System testing is the testing of a complete system prior to delivery. The purpose of system testing is to identify defects that only surface when a complete system is assembled. Testing of performance, security, configuration sensitivity, startup and recovery from failure modes are included in the system testing. Besides that, system testing consists of four basic concepts:

i. Error detection

   Identify error through some approaches such as inspection, walkthrough or others.

ii. Error removal

   Debugging and removing the identified error

iii. Error tracking

   Finding and correct the cause of the errors

iv. Regression testing

   Testing is executing after the changed or modify on the code.

Testing enhances the integrity of a system by detecting deviations in design and errors in the system. Testing aims at detecting error-prone areas. This helps in the prevention of errors in a system. Testing also adds value to the product by conforming to the user requirements.

This can help in pin-pointing the goals that the testing process needs to achieve. The objectives of the testing process have been determined as below:

i. *Software Reliability*

Making sure that the system performs critical tasks or core functions as pre-determined in the system design phase. Furthermore, specific user requests and business rules must be fulfilled.

ii. *Software Quality*

Software quality is characterized by the correctness of program logic and implementation. It begins with testing the software during development. Each module must be tested to make sure that it functions correctly at the time it is written or modified. Test values and boundary conditions must both be verified. Next, the module should undergo interface testing to check for functional errors.

iii. *System Assurance*

The main purpose of system assurance is to deliver a quality product. Conformance to requirements increases the organization's confidence in the system.

iv. *Optimum Performance and Capacity Utilization*

Another purpose of testing is to ensure optimum performance and capacity utilization of e-commerce system components. The purpose of stress or capacity testing/planning is to make sure that the system is able to perform acceptably at peak usage.

v. *Price of non-conformance*

The main purpose of testing is to detect errors and error-prone areas in a system. Testing must be thorough and well planned. A partially tested system is as bad as an untested system. And the price of an untested and under-tested system is high.

7.2 TYPES OF TESTING

There are several testing to let developers top detect and correct the error in the system. The types of testing are including unit testing, integration testing and system testing.

7.2.1 UNIT TESTING

The process of test the procedure, function or the object class of the individual component is known as unit testing. Developers use this approaches to find out the errors in the components. The purpose of the unit testing is to ensure all the components within a system can operate correctly. There are several steps being carried out for this approach:

i. Develop test cases to show input is properly converted to desired output

ii. Boundary conditions are tested to make sure the functions run at boundaries established for limiting or restricting process

iii. Debugging the error to find out the cause of error and fix it

7.2.2 INTEGRATION TESTING

The integration testing is performed after all the object, components and individual sub modules have passed to local unit tests. To ensure the system that develop meet with the system design specification, a verification process is carried out to test the system. Integration testing ensures the valid linking and dynamic relationships which establishes between modules in the whole system.

7.2.3 SYSTEM TESTING

System testing is a series of different tests which primary purpose is to fully exercise the computer-based system. It is designed to reveal bugs not possible to individual components or to interaction between components and modules. System has been tested thoroughly to ensure that the simulation run smoothly. There were involved functional testing and performance testing for the system testing of the system.

CHAPTER 8 SYSTEM EVALUTION

8.1 INTRODUCTION

At the end of the System Development Life Cycle, after the product has been delivered and deployed, the system developer is left with one final task: evaluate the system, the processes involved to develop the system and most importantly lessons learnt from the entire lifetime of the system development.

This is important as it can help us identify best practices which have been applied and remember to use it again in the future. Besides that, we must also take note of the less than perfect events that happened and remember NOT to repeat the same mistake again.

Every system should go through the system evaluation phase, for the benefit of the system and also the system developer. It is only through experience that system developers can learn and enhance their skills. Therefore, a thorough evaluation should always be carried out at the very end of every system's life cycle.

8.2 PROBLEMS ENCOUNTER AND SOLUTION TO OVERCOME THEM

The development of the system was definitely not problem-free. With a time constraint impended on the system, it is important to overcome problems as fast as possible and with the best available solutions. The project schedule should be followed stringently amid problems to ensure the system will be delivered on time.

8.2.1 DIFFICULTY IN CHOOSING SUITABLE DEVELOPMENT TOOLS

With the introduction of .NET technology, it was a blessing in some ways but a huge limitation in others. .NET provides unimaginable possibilities with its cross-language feature, managed code concept and powerful language capabilities. However, once you decide to plan the system on top of a .NET platform, there is no turning back.

There is no choice of development tools as currently there are only one .NET compliant development tools, and that is the Visual Studio .NET 2003. Whether you like it or not, you will be stuck with it for the entire lifetime of the system. There is no freedom in choosing the development tools you like, e.g. there are a myriad of development tools for the Java language: JCreator, Sun Suite or even as simple as the notepad.

As the problem with .NET is inevitable, I tried to work around the dilemma by doing extensive research on the Visual Studio.NET suite to make sure I fully understand the features embedded inside and make full use of it. If I can't fight the problem, I might as well try to work with it.

8.2.2 DIFFICULTY IN THE CHOOSING THE SUITABLE PROGRAMING LANGUAGE

After deciding on the .NET technology, I was faced with the problem of choosing the programming language for my system. In order to satisfy everybody and cater for a large audience of programmers, Visual Studio .NET 2003 encompasses a variety of programming languages, among them: Visual Basic.NET, C#.NET, Visual C++, ASP.NET and J#.

With limited knowledge on the respective programming languages, it was understandable that I was worried I will end up choosing the wrong language for the system.

It was impossible to try out all the programming language and decide which is more suitable as there was not enough time. Therefore, I decided to seek advice from professional programmers who were proficient in all the languages. I did this by joining online forums on programming languages and asking the opinions from different people to avoid biased judgments.

As a result, I got more than enough help from the online forums. The users there have been extremely helpful and provided many valuable insights on the characteristics of each programming language.

8.2.3 LACK KNOWLEDGE IN THE CHOSEN PROGRAMMING LANGUAGE

I did not have any experience before in the chosen programming language: C# and ASP.NET. Again, there was no sufficient time to learn everything entirely from scratch. Thus, I had to pick up pieces from time to time as I moved on through the development stages of the system. Due to lack of full understanding on the concept of these two languages (plus the fact that they were very new languages themselves), I encountered some unavoidable problems during the coding of the system.

Again, I turned to the Internet for the possible solutions to my problems. I limited myself to not spending more than three days on the same problem. If the same problem persists, I will look for alternative solutions. It is just not feasible to hang on to the same problem, even if I might eventually manage to solve it three weeks later. There was a deadline to consider.

8.2.4 LACK OF KNOWLEDGE ON THE CHIOSEN TECHNOLOGY

.NET Technology is still in its infancy. We have never been exposed to it formally in lectures and there is no special courses dedicated specially to it yet. Therefore, a lot of research has to be done on our own.

.NET is very famous for its own "managed library" where a huge collection of classes and methods have been created specially for the .NET technology. This means that all the .NET programming languages can utilize it and eventually any cross usage between the programming languages is supported also.

From the beginning, I had identified a few classes from the .NET managed library which are essential for the development of my system. However, as time passes by, I realized that the classes that have been chosen were unable fulfill all the needs of my system. Due to lack of understanding of the .NET framework, I was faced with the problem of having an incomplete system on my hands.

Fortunately, there was still the traditional API library to turn to. It was found to be compatible with the .NET framework and the .NET languages. In the end, the classes from the API library were used instead of the .NET managed library.

8.2.5 DIFFICULTY TO CHOOSE A MIDLEWARE

After deciding chosen .NET, a suitable middleware become another problem in this system. The current Grid middleware is developing in two main programming languages which are Java and .NET. Hence, the research on the middleware starts to find out the suitable middleware to develop the system in the .NET environment. I find more middleware and samples which operate in the .NET environment through search engine such as google and yahoo.

With limited knowledge on the middleware, it is difficult to find out the most suitable middleware. Besides that, time constraints become another problem to choose suitable middleware.

Fortunately, many researchers put the relevant resources in the internet. These resources are helping me make more suitable decision in the choosing suitable middleware.

8.3 SYSTEM STRENGTH

The system has a few characteristics which denote the strength of the system. There are as described below:

i. Immediate response to User Query

By utilizing the “Auto Post Back” function of the ASP.NET controls, any selection done on the controls is posted back to the server for immediate action without having to press a submit button. This ensures minimized response time to the user request and higher satisfaction for the user.

ii. Reduced interactions with the Database

By using the newly introduced features of the ADO.NET data access technology, interactions with the database is reduced with the Data Set object where records are only retrieved once and loaded into the Data Set object for subsequent data operations. After the data operations are no longer needed, the records in the Data Set object can be updated back to the database. Therefore, records are only retrieved and copied back once.

iii. User Friendly Graphical Interface

The graphical user interface design is tailored to the average user capabilities and promises easiness of usage. Even a novice user will have no problem navigating through the web application.

iv. Reusability

Source codes are organized into reusable classes in an object oriented environment. Interface codes are separated from the logic processing clearly. This advocates for easy reuse of the different components of the system or while adding future enhancements to the source codes.

v. Automatic startup at System Boot Time

The agent which installed in the client computer will start up automatically after system boot up without any manual intervention by the administrator.

8.4 SYSTEM LIMITATIONS

The finished system is not without its limitations. There are described as below:

i. Invisibility of the Indexing Agent

The agent which is responsible for indexing the data in the database on the schedule time still visible from sight although can hide to the system tray. Users can stop the agent easily.

ii. Limitation to the Windows platform

Although Grid computing do not have any limitation to the platforms, this system just only execute in the Windows platform due to the chosen programming language. The system cannot operate in the Linux environment.

iii. No file replication

The entire files in the databases are standalone and no replication. Users can only retrieve file from single databases. Therefore, users fail to get the files if the files are corrupted or deleted by the owner.

8.5 FUTURE ENHANCEMENTS

There are a few suggestions for future enhancements to overcome the limitations of system. The suggestions are showed below:

i. Agent invisible from sight

The indexing agent should be visible from sight to ensure users do not stop agent easily.

ii. The system should operate in different platforms

To satisfy the Grid computing requirement, the system should operate in different platforms. The chosen middleware play important role to overcome this problem.

iii. File replication

The system should do the replication to improve the performance of the system

REFERENCES

Document

[1] Ian Foster, Carl Kesselman, and Steven Tuecke (2004). The Anatomy of the Grid Enabling Scalable Virtual Organizations. Available at http://www.globus.org/research/papers/anatomy.pdf

[2] Ann Chervenak, Ian Foster, Carl Kesselman, Charles Salisbury, and Steven Tuecke. (2004). The Data Grid:Towards an Architecture for the Distributed Management and Analysis of Large Scientic Datasets. Available at http://www.globus.org/documentation/incoming/JNPCApaper.pdf

[3] Vijayshankar Raman, Chris Crone, Laura Haas, Susan Malaika, and Chaitan Baru. (2002). Data Access and Management Services on Grid. Available at www.cs.man.ac.uk/grid-db/papers/dams.pdf

Sample Application

[4] Biology WorkBench. Available at http://workbench.sdsc.edu/

[5] DSG Grid Portal. Available at https://l59.dsg.port.ac.uk/dsgPortal/

[6] CCLRC e-Science DataPortal. Avaliable at http://www.e-science.clrc.ac.uk/web/projects/dataportal, http://dataportal.dl.ac.uk:8080/, and http://dataportal.dl.ac.uk:8080/dp/

[7] Alliance Science Portal. Available at http://www.extreme.indiana.edu/alliance

[8] International Lattice DataGrid. Available at http://www.lqcd.org/ildg/

Grid Middleware

[9] GridPort middleware. Available at https://gridport.npaci.edu/

[10] MS.Net Grid. Availbale at http://www.epcc.ed.ac.uk/~ogsanet/

[11] OGSI.Net Grid. Available at http://www.cs.virginia.edu/~humphrey/GCG/ogsi.net.html

[12] Globus Toolkits. Available at http://www.globus.org

[13] MyGrid. Available at http://www.mygrid.org.uk

[14] MSRF.Net. Available http://www.cs.virginia.edu/~gsw2c/wsrf.net.html

Web Sites

[15] http://www.mysql.com

[16] http://ip158.fsktm.um.edu.my/ilmiah2004/

[17] ftp://202.185.107.158/research/thesis

[18] http://www.winsupersite.com/reviews/

[19] http://www.codeproject.com

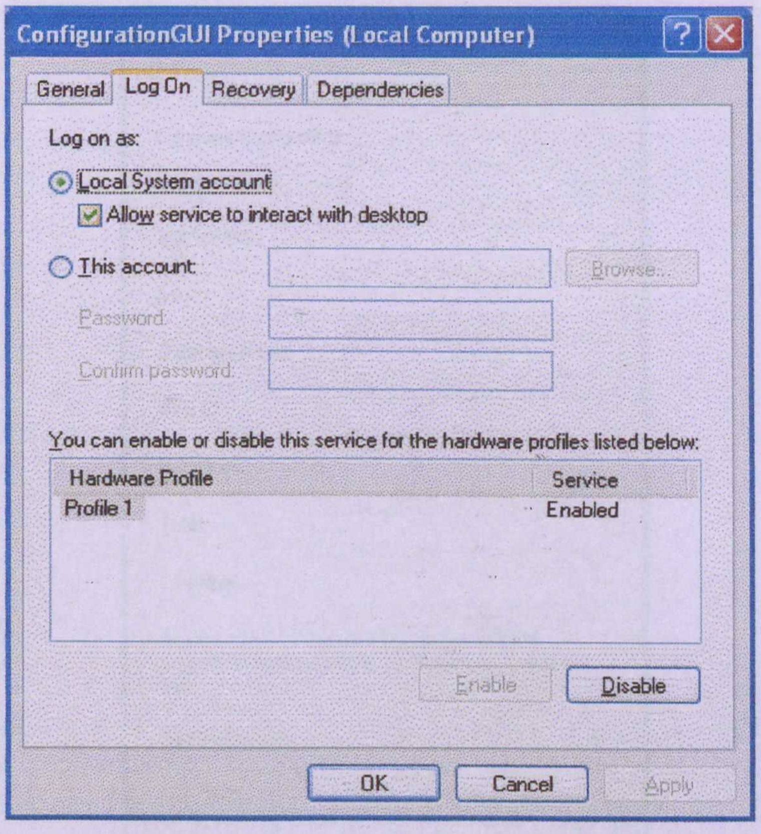
[20] http://www.csharpcorner.com

USER MANUAL

This user manual is for the FSKTM Data Portal. This user manual divides into two parts which are prerequisite requirements and description of the each module's functions.

PREREQUISITE REQUIREMENTS

1. Install WSRF.Net and Web Services Enhancement Service Pack 1 in the server machine.
2. Install the GridServer into the server machine. Then go to Control Panel -> Administrative Tools-> Services, start the ServerSocket.
3. Install the GridAgent into the client machines. Then go to Control Panel -> Administrative Tools-> Services, select the properties of ConfigurationGUI then select the allow service to interact with desktop. After that, start the ConfigurationGUI and download



ConfigurationGUI Properties (Local Computer)

General | Log On | Recovery | Dependencies

Log on as:

● Local System account

☑ Allow service to interact with desktop

○ This account:

Password:

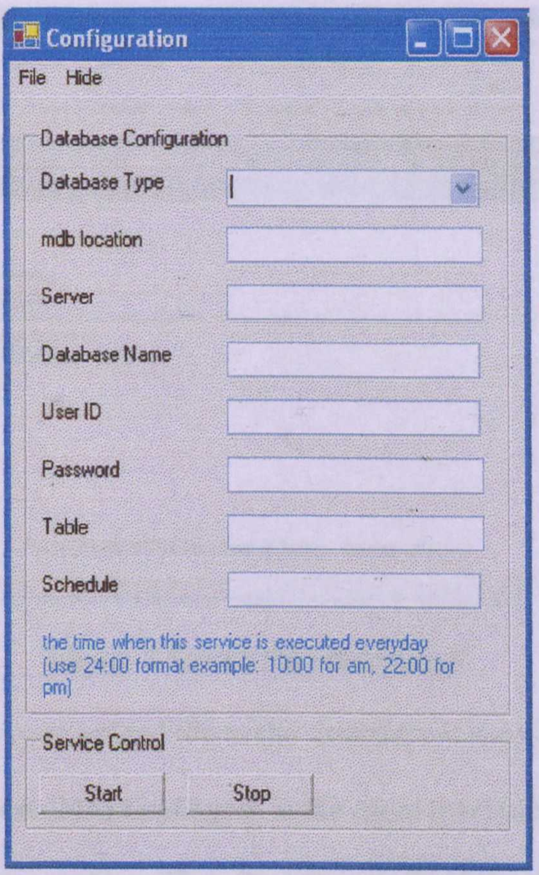
Confirm password:

You can enable or disable this service for the hardware profiles listed below:

Hardware Profile	Service
Profile 1	Enabled

Figure 1The properties dialog box of ConfigurationGUI

Another GUI is prompt out. User need to fill in the all the information that requested and click the start button.

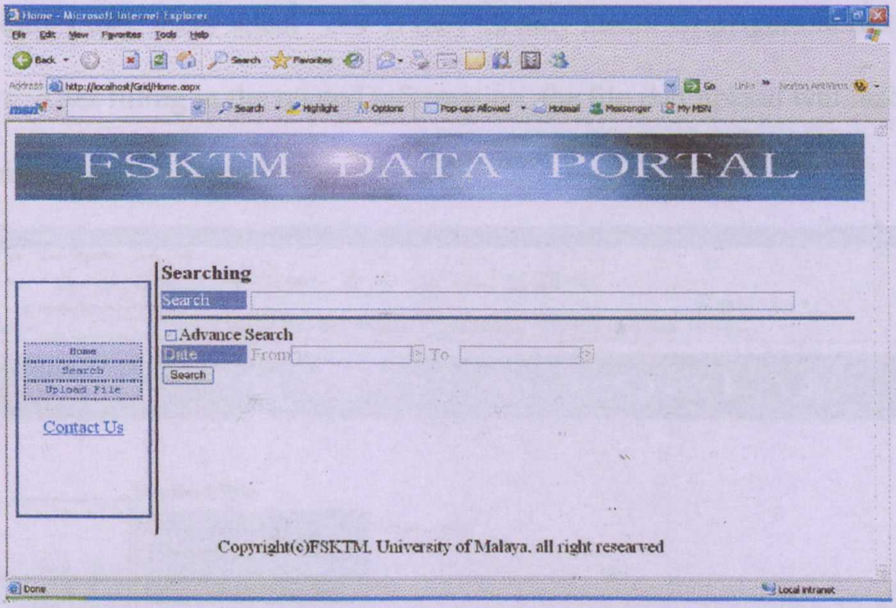


the time when this service is executed everyday (use 24:00 format example: 10:00 for am, 22:00 for pm)

Figure 2 GUI of ConfigurationGUI to let users fill in information

FUNCTIONALITY OF THE SYSTEM

1. Users can search the information through this search module. Users need to key in keyword in the search field. Then click search button. Besides that, users can use advance search to limit the result that come back based on the date. After the result returned to the server, the web page will display all result to users.


Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Grid/Home.aspx
Go Links Norton AntiVirus
Search Highlight Options Pop-ups Allowed Hotmail Messenger My MSN
Home
Search
Upload File
Contact Us
Search
Done
Local intranet

FSKTM DATA PORTAL

Searching

Search

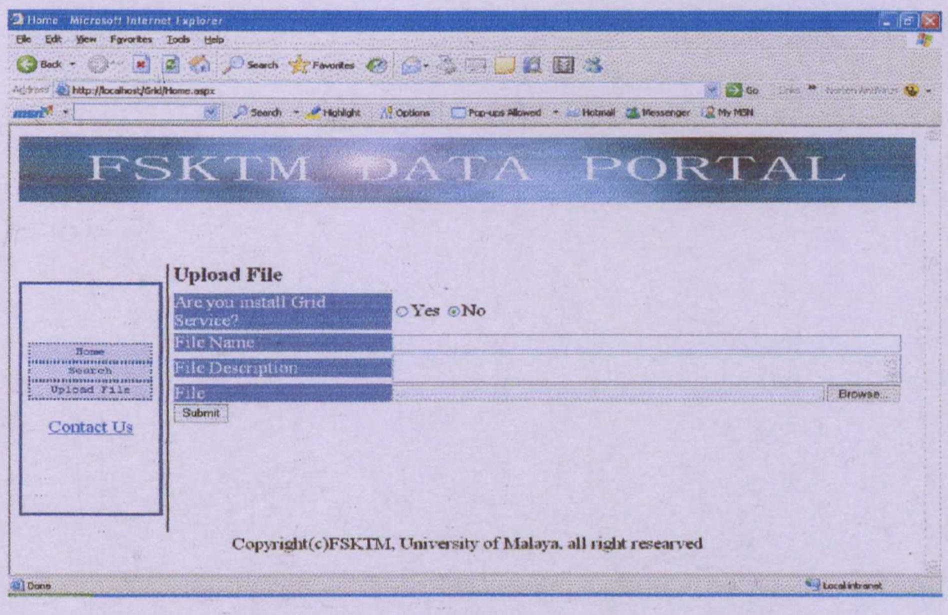
□ Advance Search

Date From To

Copyright(c)FSKTM, University of Malaya, all right researved

Figure 3 Search the information

2. Besides that, users can upload file to the databases in the Grid environment. if the users do not install the GridAgent in the client machines, they need to choose the 'No' in the first question. Then the users require filling in the related data and browsing for the files which need to upload. Then click submit button. The files will copy to the default database in the server.


Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Grid/Home.aspx
Go Links Norton AntiVirus
Search Highlight Options Pop-ups Allowed Hotmail Messenger My MSN
Home
Search
Upload File
Contact Us
Browse...
Submit
Done
Local intranet

FSKTM DATA PORTAL

Upload File

Are you install Grid Service? ○ Yes ● No

File Name

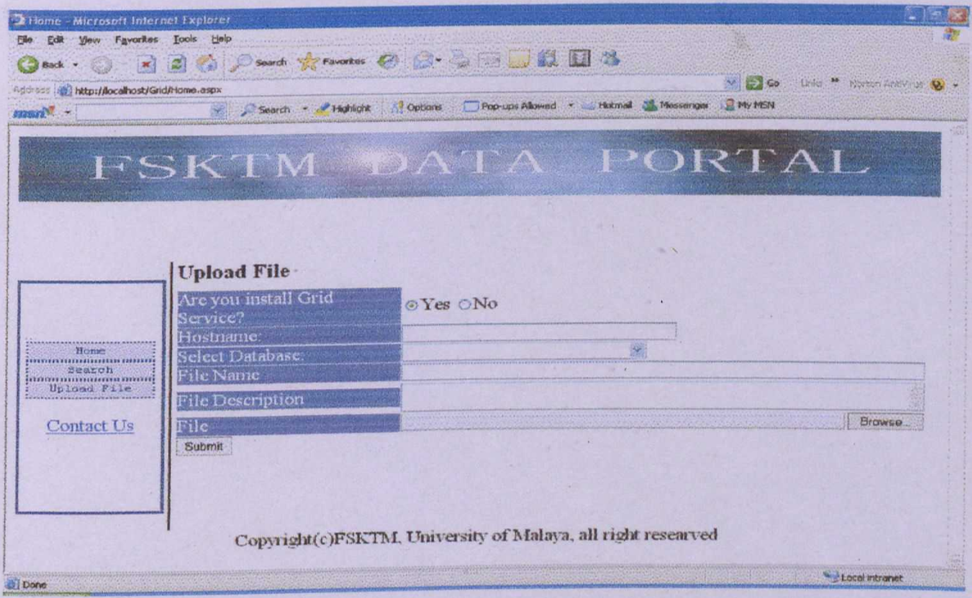
File Description

File

Copyright(c)FSKTM, University of Malaya, all right researved

Figure 4 Upload file in the environment without GridAgent

Besides that, users select 'Yes' if they already install GridAgent in the client computers. After filling in the related information, the file that upload will insert into the local database.


Home - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost/Grid/Home.aspx
Go Links Norton AntiVirus
msn Search Highlight Options Pop-ups Allowed Hotmail Messenger My MSN
FSKTM DATA PORTAL
Home
Search
Upload File
Contact Us
Browse...
Submit
Done
Local intranet

Upload File

Are you install Grid Service? ● Yes ○ No

Hostname:

Select Database:

File Name

File Description

File

Copyright(c)FSKTM, University of Malaya, all right researved

Figure 5 Upload file in the GridAgent environment