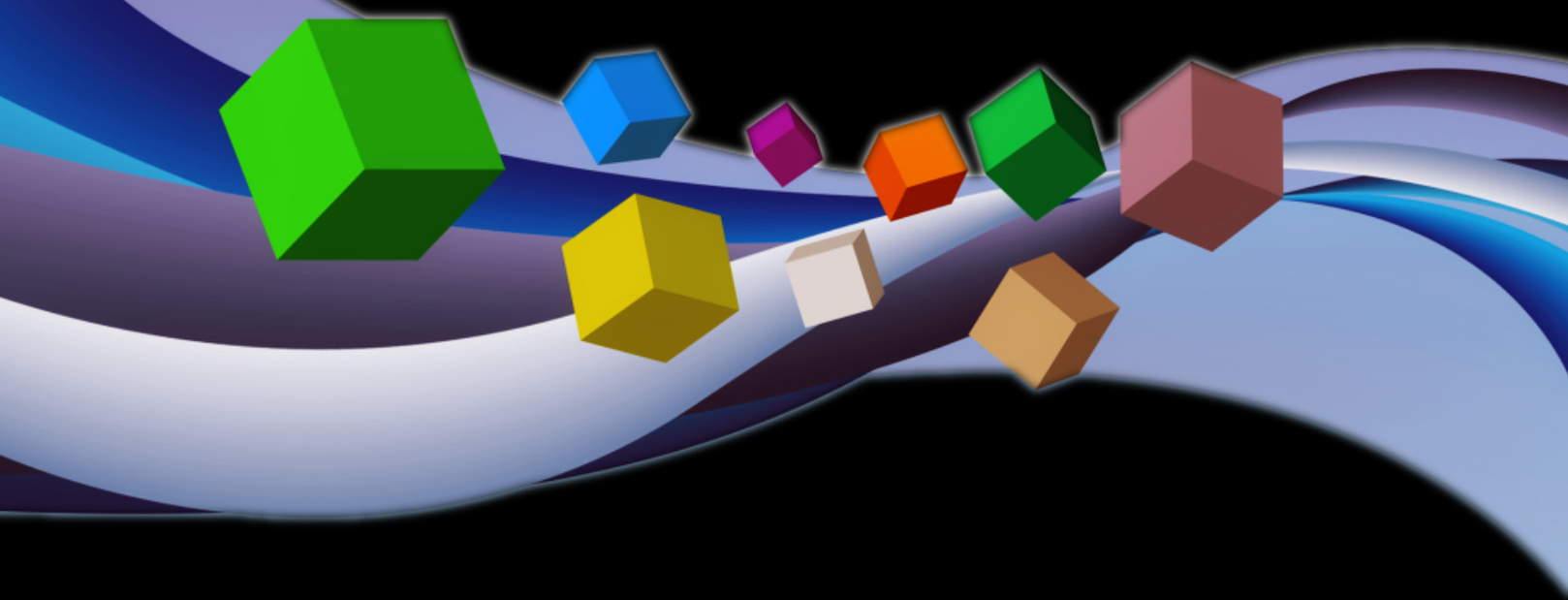
Grid Computing Cluster

The Development and Integration of Grid Services and Applications

Grid@USM
2008/09 Report



Edited by Bahari Belaton and Lim Lian Tze
Platform for Information & Communication Technology Research
Universiti Sains Malaysia

Grid Computing Cluster

The Development and Integration of Grid Services and Applications

Grid@USM
2008/09 Report

Edited by Bahari Belaton and Lim Lian Tze

Platform for Information & Communication Technology Research
Universiti Sains Malaysia

Platform for Information & Communication Technology Research
Universiti Sains Malaysia
11800 USM, Penang, Malaysia

Perpustakaan Negara Malaysia Cataloguing-in-Publication Data

Grid computing cluster : the development and integration of grid services and applications / edited by Bahari Belaton, Lim Lian Tze.
ISBN 978-983-3986-58-3
1. Computational grids (Computer systems). 2. Electronic data processing–Distributed Processing. I. Lim, Lian Tze.
004.36

Proof-read by Norliza Hani Md. Ghazali.

Designed and typeset by Lim Lian Tze with LaTeX.

First printing, July 2009.

Cover image by ©B S K (`http://www.sxc.hu/profile/spekulator`).

CONTENTS

Project Overview

Sub-Projects

Project Activities

Note: The abbreviations Prof. and AP are used for *Professor* and *Associate Professor* respectively throughout this book.

Image: The Petronas Towers, Kuala Lumpur. Photograph by ©Mahjabeen Mankani (http://www.sxc.hu/profile/mejmankani).

PROJECT OVERVIEW

Image: It's a wide, yet connected world. Illustration by ©B S K (http://www.sxc.hu/profile/spekulator).

INTRODUCTION

Grid computing can provide easy and transparent access to high performance computing by aggregating geographically distributed resources to act as a single powerful resource. With grid computing technologies, users can access to any virtual resources such as computers, special instruments, software applications and databases in a seamless manner. It is analogous to the electric power grid, where users do not need to know the details of the technology. They simply connect to common interfaces and subscribe only what they need. Grid computing has evolved from cluster computing, meta-computing, utility computing and today, a new term has been coined as cloud computing.

In order to achieve the objective of providing seamless access to high performance resource and virtual services, the concepts of aggregation and management are vital. Two types of resources need to be aggregated, hardware and software. The management aspect involves management of technical applications (resource management) and jobs (resource allocation). Resource monitoring monitors the grid resources and mitigate problem while resource allocation accepts job submission and ensures efficient and accurate delivery. Aggregation of grid services is classified into data and applications. Distributed data management (data placement, data fragmentation or replication and heterogeneous data transformation) provides data distribution transparency for various data sources.

The grid application can be any kind of services that can reside anywhere and accessible from any place. Application distribution transparency is a main objective in gluing different services together in grid environment. In order to manage the unstructured, distributed services, a good resource discovery framework has to be in place.

Universiti Sains Malaysia (USM) is one of the first universities to embark on grid computing research in Malaysia. In 2002, the first grid testbed was established through an e-Science project involving USM, Univeristi Teknologi Malaysia (UTM), and University of Malaya (UM) in drug discovery and liquid crystal simulation. The Malaysian Biogrid testbed has been established in collaboration with the Malaysian Biotechnology and Bioinformatics Network, Universiti Kebangsaan Malaysia (UKM), and Sun Microsystems. Through The Pacific Rim Applications and Grid Middleware Assembly (PRAGMA) and as part of the newly formed Malaysian Research Network (MyREN), USM is committed to taking Grid Computing Applications and Research in Malaysia to the next level.

USM researchers have communicated with other PRAGMA participants about experiences in understanding practice and policy issues concerning sharing resources and the establishment of collaborations with other PRAGMA institutions.

In 2002, a research grant was awarded for the project *"The e-Science Grid: Development of Back-end Grid Engine and Grid Infrastructure"*. Two high performance clusters were assembled, each with 34 processors, 18 GB RAM and 160 GB shared storage, running on a high-speed Gigabit Ethernet backbone switch. One cluster is located at USM and the other at Universiti Teknologi Malaysia, Kuala Lumpur (UTM-KL). The *e-Science Grid Portal* integrates both *e-Science Grid* components (such as Grid Resource Monitoring, Grid Resource Allocation, Grid Resource Metering, Grid Resource Prediction, etc) and *e-Science Grid* applications (such as Iterative Solver Agent, Molecular Simulation and Molecular Docking).

In order to deliver a powerful grid that enables easy and secure access to aggregated resources, a portal has been developed to enable access from anywhere at any time to distributed technical applications and data. The *e-Science Grid Portal* also allows an organisation to employ a single URL through which users can gain browser-based access to customised or personalised information, as well as applications. With a customisable interface that provides content which meets each individual user's specific needs, portals spare users the hassle of hunting for information by making it accessible from any networked device. By aggregating information, standardising tools, and providing global access to applications and information, portals give organisations the agility they need to provide distributed shared services to users through a single point of access.

In the year 2004, USM joined effort with Asia Pacific Science and Technology Center to set up a biogrid for local access within USM. This effort is further extended to include the Genome Center in UKM. The aim of the biogrid is to provide a testbed for running biological applications between USM and UKM. Another notable achievement is the Sun-USM collaboration, which is tied with Nanyang Technological University, Singapore, via the Asia-Pacific Science & Technology Center (APSTC). In the MoU signed for the setting up of the Bio-grid test bed between USM and UKM, sponsored by Sun Microsystems, one proposal was to set up an Asia-Pacific Science & Technology Forum (APSTF) at USM. Training and workshops in grid computing and bioinformatics will be conducted under APSTF, and there will be collaboration on research projects with the parent center (APSTC) and other APSTF sites in India and Thailand. This will lead to the formation of an international grid in the Asia Pacific region, connecting Singapore, India and Thailand. APSTF Malaysia plans to propagate the convenience and relevance of grid computing to researchers in various areas, and not just in the sciences but also in the social sciences and education, so that the grid will be fully utilised and its benefits reaped by all.

Limitation faced at the beginning stage is the issue of limited IP addresses. At the moment, the Aurora Cluster cluster, which has previously been the most deployed by others, is now used as a gateway to others. Under the Aurora Cluster, users of PRAGMA are connected through MyREN. This cluster is meant for regional connectivity including UM, USM and UTM.

In the year 2007, under the Research University grant scheme, we have obtained a new grant for the project "Grid Computing Cluster – the Development and Integration of Grid Services & Applications" to further strengthen the research and development of campus grid projects which consist of various applications namely:

- Setting up USM Campus Grid,
- Grid Application to Wave Front Propagation and Containment of Vector-Borne Diseases,
- Dynamic Replica Management in Data Grid Environment,
- Using Grid Technology to Create a Render-Farm for Blender 3D Animation,
- Grid-enabled Blexisma2,
- B2B Standards Component Modeling,
- An Automated Java Testing Tool on the Grid, and
- iNet-Grid.

Each of these sub-projects will be described in later chapters.

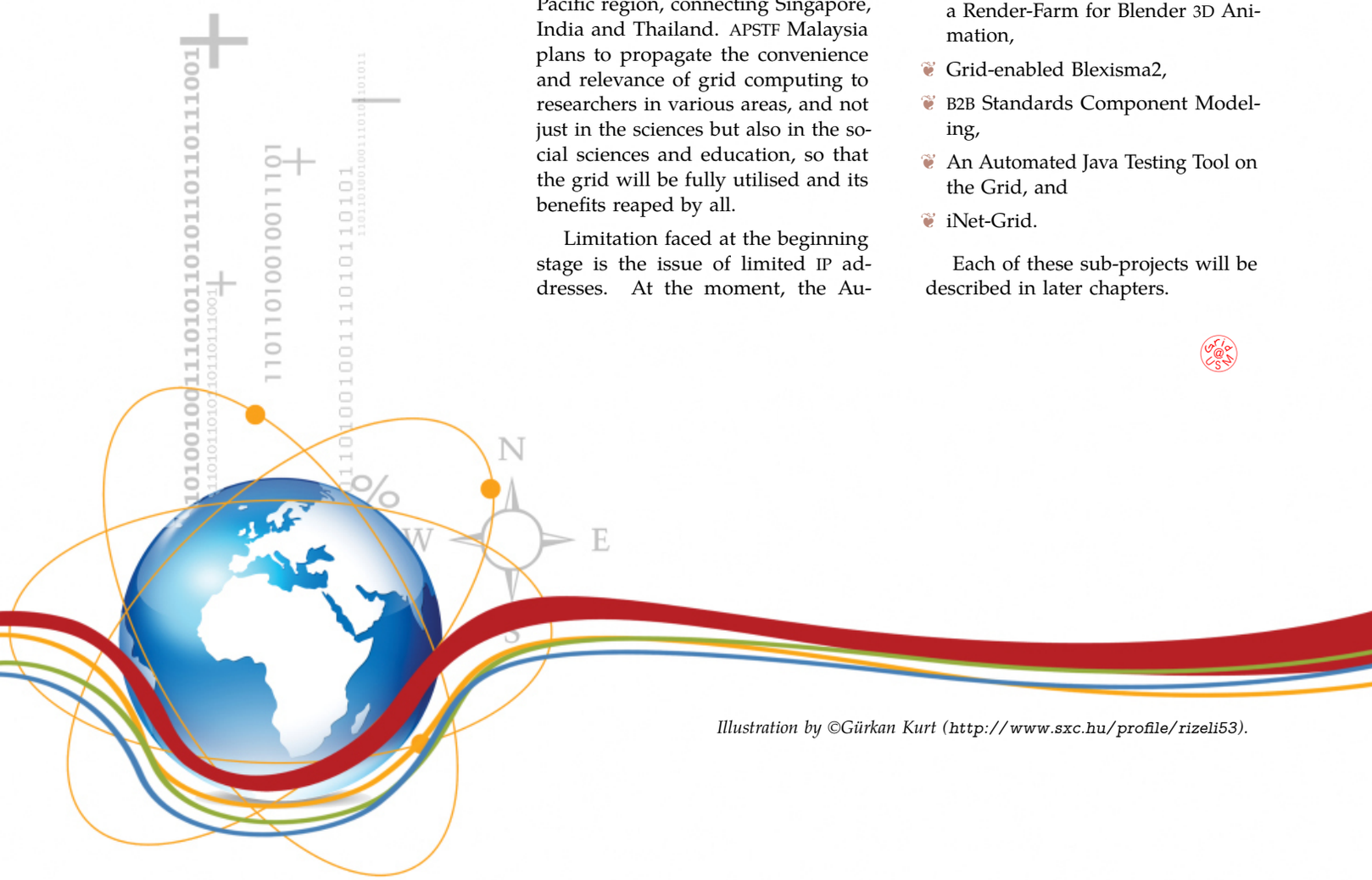
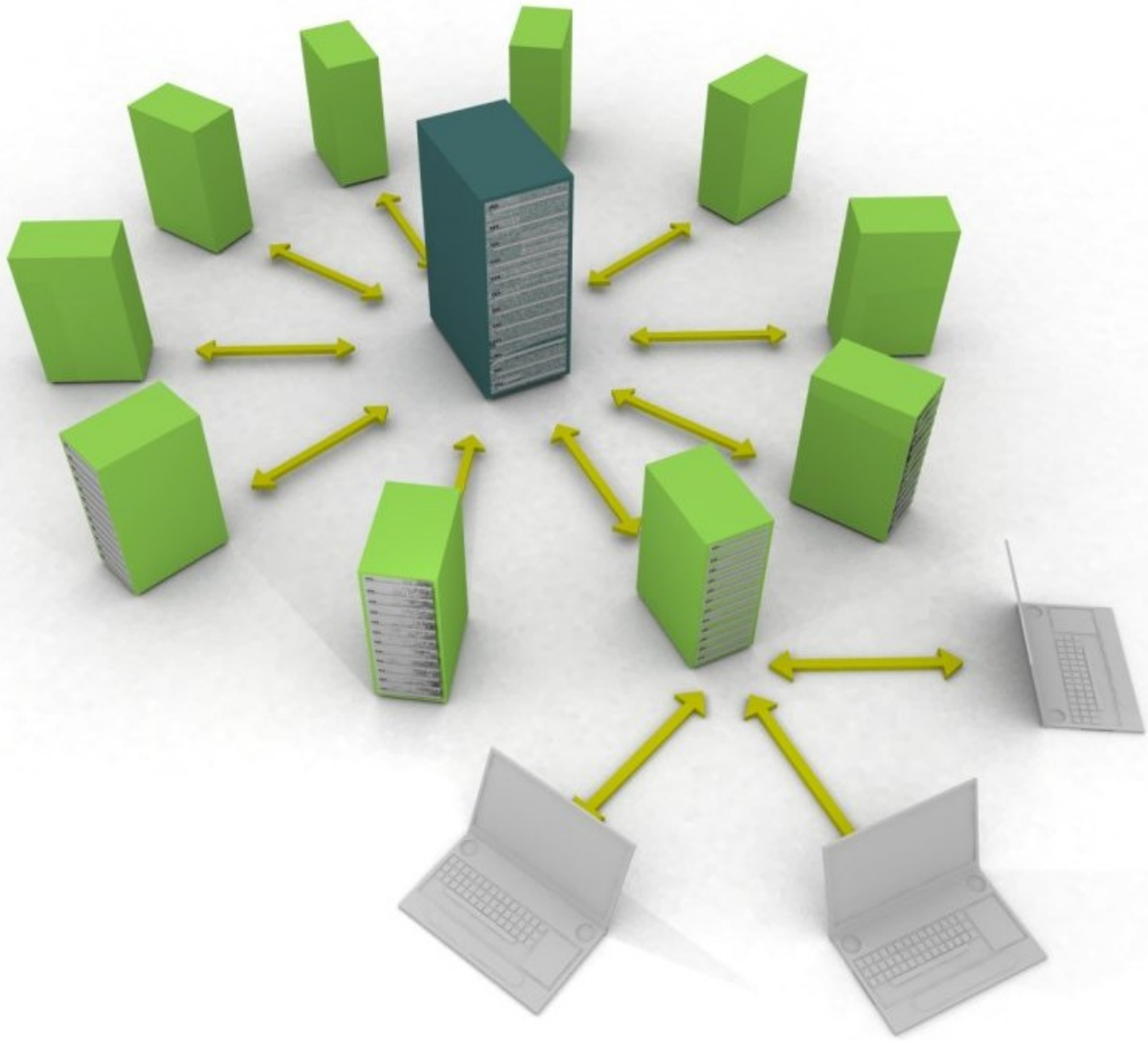


Illustration by ©Gürkan Kurt (http://www.sxc.hu/profile/rizeli53).

Image: A simple network. Illustration by ©Svilen Mushkatov (http://www.sxc.hu/profile/svilen001).



Grid@USM:
DEVELOPING & INTEGRATING GRID APPLICATIONS & SERVICES

USM'S R&D SUSTAINABILITY

To attain the level of excellence as a research university (RU), USM in recent years has seriously engaged herself in various efforts to formulate and chart her future research and development (R&D) directions. The pinnacle of these efforts was translated into a 2 year strategic plan "USM Research-Intensive University 2007–2009".

One of the crucial input elements for attaining and subsequently sustaining RU status is the creation of highly conducive research environment and infrastructure. Grid computing is an example of such infrastructure, aiming at providing excellent research facilities by ensuring good computational horsepower to support high impact domains in the bio-sciences, physical sciences, information and communication technology (ICT), Environment sciences, Education, Arts and others.

Grid computing is by its nature highly distributed geographically, consists of highly specialised equipment (storage, grid engines and management tools), as well as expensive. It is as such an ideal example of common, core computational resources to be created and pooled among aspiring researchers who require high performance resources.

Equally important, USM needs a focused, holistic and dedicated effort

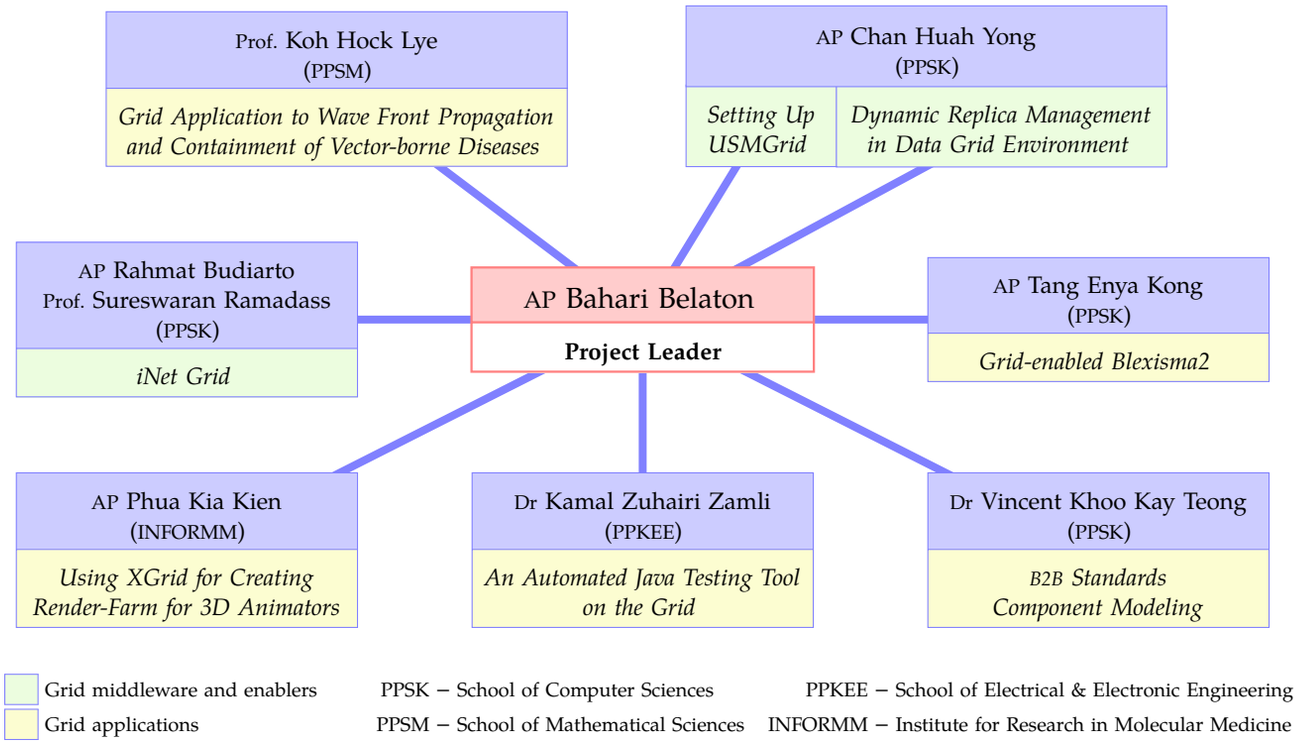


Figure 1: Grid@USM Project Team and Sub-projects

to build a sustainable human capital in grid computing. We need *research-based human capital* in order to achieve and maintain technical human resource support with technical skills, knowledge and experience to ensure uptime of project deliverability, releasing academics to concentrate on activities to further the frontier of knowledge.

GRID COMPUTING AND USM

USM is no stranger to grid computing. We pioneered various grid computing research initiatives at both national and international levels, as exemplified by our past achievements:

- USM led the first e-Science grid test-bed in 2002, connecting USM–UTM–UKM in a project to simulate liquid crystal and drug discovery.
- USM also led the bio-grid test-bed engaging industry partners Sun Microsystems, Biotechnology and Bioinformatics Network (BBN) and UKM.
- USM launched its *University Grid and International Collaboration* initiative in May 2005 to provide a well-designed grid computing infrastructure among its three campuses, pooling grid resources to provide computational support to R&D activities.
- USM participated in Malaysian Research Network (MyREN) which provided the bandwidth to connect grid resources within and outside USM.
- USM is the first Malaysian member in The Pacific Rim Applications and Grid Middleware Assembly (PRAGMA). The PRAGMA community regularly uses USM's computational resources via the AURORA cluster.
- An MoU was signed between USM, Sun-APSTC (Asia-Pacific Science & Technology Center) and National University of Singapore (NUS), where one of the proposals was to set up Asia-Pacific Science & Technology Forum (APSTF) at USM.
- USM was granted the honour to host PRAGMA 15 in October 2008, a milestone in charting and leading R&D in grid computing.

Physically, pockets of grid infrastructures in the form of clustered computers, high-end grid engines and other specific resources (e.g. domain-specific software) have been acquired and deployed within USM. The grid infrastructures are mainly concentrated at the School of Computer Sciences (PPSK), the Centre for Knowledge, Communication & Technology (PPKT) and the Laboratory of Biocrystallography & Structural Bioinformatics (MBBS). The latest addition is the grid engine deployed at the Collaborative μElectronic Design Excellence Center (CEDEC) at the Engineering campus.

GRID ACCESSIBILITY AND HUMAN EXPERTISE

Despite the various progresses made, particularly with respect to infrastructure, the real benefits of these computational powers to the scientists were not yet fully realised. The penetration rate among pure end-users is perhaps at the lower end – 10 % at most. Clearly, with such costly investment in the infrastructure and heavy participation from technologists (particularly computer scientists), a well-designed deployment plan to increase accessibility to the end-users is desperately needed. In fact, PRAGMA was established solely to handle this last-mile issue – making grid resources accessible to researchers.

Zooming into the technical level, there are two main barriers hampering the quick and large-scale acceptance of grid computing among researchers.

Firstly, the technology itself still needs further R&D efforts to make it more user-friendly. The ultimate

aim here is to allow researchers with computation-intensive requirements to simply *plug* their application into grid technology and run it seamlessly without major user intervention. An analogy would be the seamless manner with which we plug electrical devices into power sockets and they will simply function – without us knowing what actually goes on behind the power socket.

The second problem is related to the readiness of grid technology. There are still many open research questions to address its four main domain areas: policy and governance of grid resources; grid middleware and tool enablers; grid applications; infrastructure and security.

It is with USM's R&D needs and sustainability in mind that we set up Grid@USM, a grid computing research cluster to address the problems mentioned above. Our two main driving principles and objectives are:

- Training of human capital in grid technology, and
- Increasing the accessibility and utilisation of USM grid infrastructures.

Grid@USM's pilot project, led by AP Bahari Belaton and entitled *Grid Computing Cluster – the Development and Integration of Grid Services and Applications*, was initiated under an RU research grant. This RU project involves integrated research coupling expertise from different fields. The technologists (mainly computer scientists) will continue to work and improve on the previously mentioned grid computing domain areas, while grid infrastructure accessibility and utilisation is increased by grid-enabling existing research applications by experts from other domains. This is directly related to our third objective:

- To showcase these grid-enabled applications during PRAGMA 15.

The human capital training will focus on research-based skills, knowledge and hands-on experience in grid computing for both technologists and domain experts.

Project Objectives

- *To increase accessability and utilization of* USM Campus Grid *infrastructure*
- *To train research-based human capital in grid computing*
- *To showcase potential grid applications in PRAGMA 15*

THE IMPLEMENTATION PLAN

The project can be viewed as having two main domains:

Grid middleware and tool enablers, concerning the administration and management of grid resources and users. We allocated 1/3 of the overall project resources to this fundamental stage.

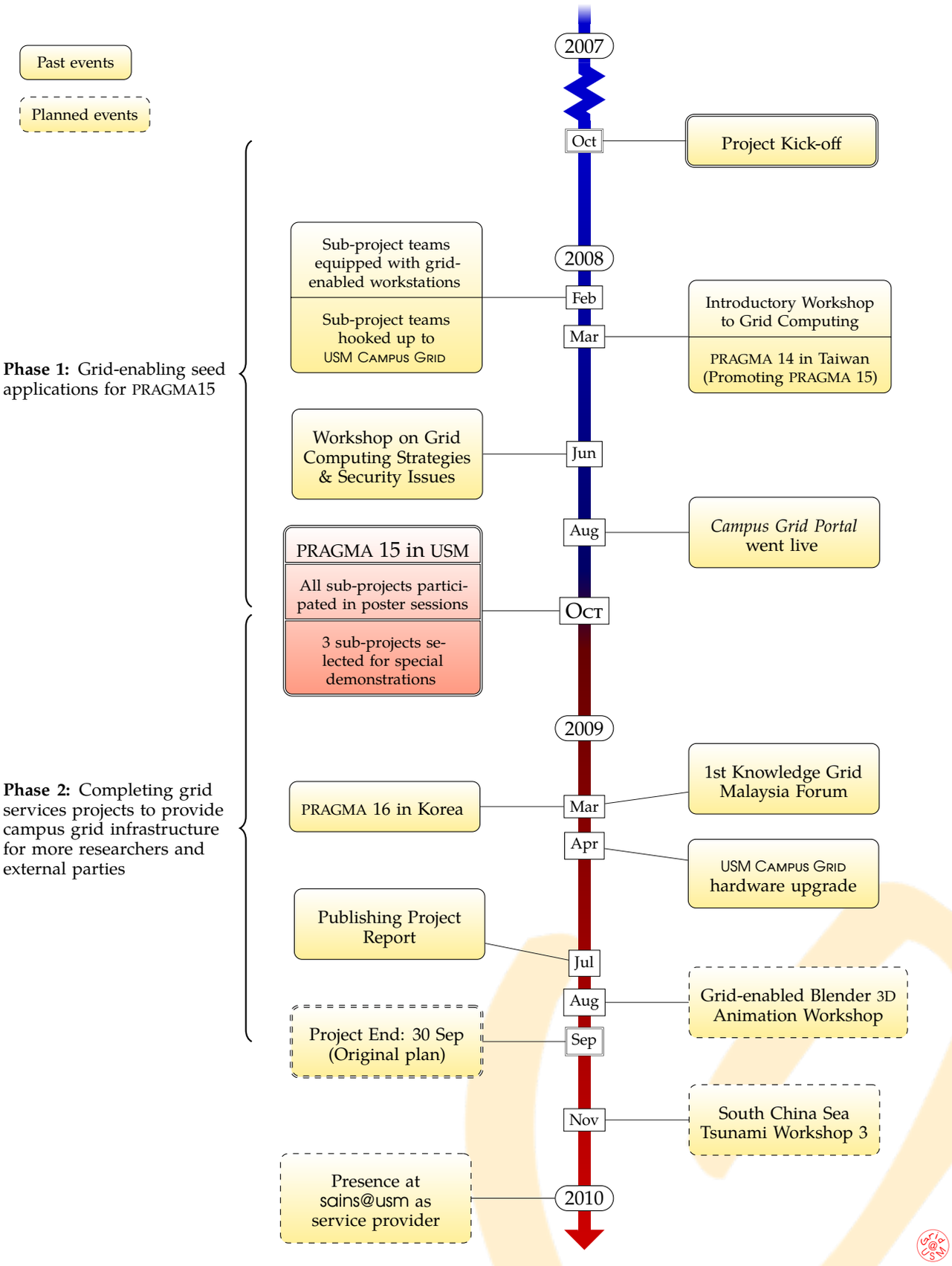
Grid applications, where readily available applications are grid-enabled to run on existing grid infrastructure available at USM. This accounts for 2/3 of the project.

The former task is entrusted to the Grid Computing Lab (GCL) of PPSK. As for the latter aspect, a call for research proposals issued by the ICT Research Platform elicited applications from researchers of various disciplines, ranging from medicine, computer sciences to engineering. Seven seed applications and their principle investigators were identified. Together with the task of setting up USM Campus Grid i.e. the grid infrastructure for researchers, we have a total of 8 sub-projects, the leaders of which are also members of the Grid@USM team.

To achieve the project aims, domain experts in the respective fields were employed as Graduate Research Assistants (GRAs) and/or Research Assistants (RAs) for each grid application sub-project. These GRAs and RAs were given training to get acquainted with the grid engines and infrastructure at USM. They then adapted the domain-specific applications to run on the grid environment under guidance of the respective sub-project leaders. Finally, these grid applications were deployed on USM Campus Grid, where they are accessible via various grid tool enablers. Details about the research methodology of each sub-project are reported in later chapters, as are the various activities carried out under this project.

Grid@USM

PROJECT MILESTONES

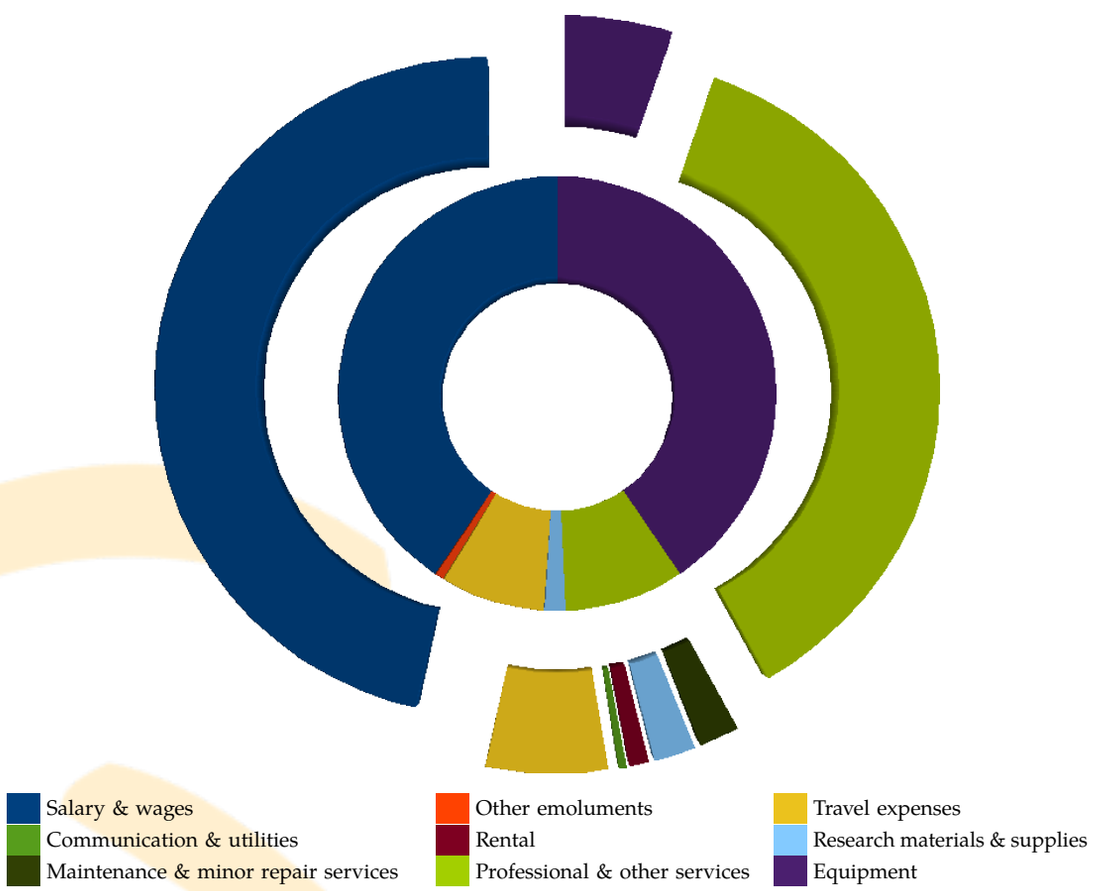


PROJECT EXPENDITURES

January '08 – May '09

Project allocation and expenditures until May 2009. All amounts are in RM.

	USM Expenditure Code & item	Project Allocation	Amount Spent	Project Balance
111	Salary & wages	465,600	274,885	190,715
115	Other emoluments	0	4,272	−4,272
221	Travel expenses & subsistence	60,000	52,456	7,544
223	Communication & utilities	4,000	5	3,995
224	Rental	10,000	0	10,000
227	Research materials & supplies	20,400	11,226	9,174
228	Maintenance & minor repair services	20,000	0	20,000
229	Professional & other services	368,000	59,579	308,421
335	Equipment	52,000	274,436	−222,436
	Total	**1,000,000**	**676,859**	**323,141**



Project allocation (outer ring) and actual expenditure up till May '09 (inner pie)

Image: Eye on Malaysia by Titiwangsa Lake, Malaysia. Photography by ©Mee Lin Woon (http://www.sxc.hu/profile/meelin).

SUB-PROJECTS

Image: Server room at DESY, Hamburg. Photograph by ©Christian Hoffmann (http://www.sxc.hu/profile/brcwcs).

SETTING UP USM CAMPUS GRID

Grid Computing Lab

School of Computer Sciences

TEAM MEMBERS

Project Leader : AP Chan Huah Yong

Researchers : Ang Sin Keat, Tan Chin Min, Kheoh Hooi Leng, Cheng Wai Khuen, Muhammad Muzzammil bin Mohd Salahudin

INTRODUCTION

In order to increase the accessibility of USM's grid infrastructure, we at the Grid Computing Lab (GCL) embarked on this effort to set up USM Campus Grid to link research groups in all 3 Universiti Sains Malaysia (USM) campuses. USM Campus Grid also acts as a testbed for the sub-project teams' grid-enabled applications. We welcome researchers, both in and outside USM, to utilize USM Campus Grid resources.

In a nutshell, our objective is to integrate clusters at the 7 sub-project team locations with GCL filtered available resources and PPKT resources.

USM CAMPUS GRID RESOURCES

Centrally, several machines were set up at GCL to serve all sub-project team members, to handle directory and service queries (Lightweight Directory Access Protocol or LDAP), grid credential management, user authorisation, as well as for computational processing needs.

On the other hand, this project involves 7 sub-project teams from different research groups and centres of USM, spanning all 3 campuses in different geographical locations:

Main campus (Minden, Penang)

- **Grid Computing Lab (GCL), School of Computer Sciences:** Dynamic Replica Management in Data Grid Environment.
- **Computer-aided Translation Unit (UTMK), School of Computer Sciences:** Grid-enabled Blexisma2.
- **School of Mathematical Sciences (PPSM):** Grid Application to Wave Front Propagation and Containment of Vector Borne Diseases.
- **Information Systems Engineering Research Group (ISE), School of Computer Sciences**: B2B Standards Component Modeling.
- **Network Research Group (NRG), School of Computer Sciences:** iNet Grid.

Transkrian Engineering Campus (Nibong Tebal, Penang)

- **School of Electrical & Electronic Engineering (PPKEE):** An Automated Java Testing Tool on the Grid.

Health Campus (Kubang Kerian, Kelantan)

- **Institute for Research in Molecular Medicine (INFORMM):** Using XGrid for Creating Render-Farm for 3D Animations.

We supplied each sub-project team with a Dell Vostro 400 Mini Tower Desktop workstation, on which we installed and configured the *Rocks Clusters 4.2.1 (Cydonia)* operating system, the Sun Grid Engine (SGE), the *Globus* toolkit, the *MyProxy* credential management service, the *Ganglia* moni-

Table 1: Recent USM Campus Grid resources

Cluster	No of CPU	Physical Hosts	Memory (each host)	Storage (each host)
Aurora	32	16	2 GB (master node) 1 GB (compute node)	80 GB
GCLDatarm	20	5	2 GB	500 GB (master node) 250 GB (4 compute)
GCLMicro	40	5	8 GB	500 GB
GCLHome	8	1	8 GB	5 TB
LDAP server	4	1	4 GB	500 GB
gclportal	4	1	4 GB	500 GB
gclproxy	2	1	1 GB	160 GB
gcldns	1	1	786 MB	160 GB
mathvbd	4	1	2 GB	500 GB
isitb2b	4	1	2 GB	500 GB
utmkblex	4	1	2 GB	500 GB
engtest	4	1	2 GB	500 GB
navómon	4	1	2 GB	500 GB
Comsics	–	–	–	–

Table 2: Physical machines purchased under RU grant

Model	Quantity	Particulars
Dell Vostro 400 Mini Tower Desktop	7	A single desktop was distributed to each member in the campus grid project
Apple X-Grid Server CTO	1	Used for INFORMM's project
Sun SPARC Enterprise T1000 Server	3	Machines just arrived and still in installation progress.
Dell Power Edge™ 1950 Rack Mount Server	1	Machine just arrived and still in installation progress.
Dell Power Edge™ 1950 Rack Mount Server	3	Machines just arrived and still in installation progress.
Dell Power Edge™ SC1435 Server	5	Set up as GCLMicro cluster.

toring system, and tools for Message Passing Interface (MPI) programming.

Our research officers (ROs) set up the machines on-site at each sub-project team's lab, connecting them to USM Campus Grid such that each cluster can access resources on all other clusters in the grid. We also connected the USM Campus Grid to the Malaysian Research Network (MyREN), therefore allowing external parties to utilize USM grid resources as well – including PRAGMA members. We also created user accounts and grid credentials for the sub-project team members during this set up exercise.

The current machine resources are listed in Tables 1 and 2, while Figure 1 shows a conceptual view of USM Campus Grid's architecture.

USM CAMPUS GRID USERS

In line with Grid@USM's objective of increasing the accessibility and utilisation of USM grid infrastructures, we welcome researchers – both in and outside the School of Computer Sciences and even USM – to use USM Campus Grid for their computational needs. Table 3 lists current registered USM Campus Grid users and their research projects.

SHARED HOME DIRECTORY

Each sub-project team originally maintained their data and application files on their own machines, and therefore had to transfer files between machines before submitting processing jobs on other clusters. We therefore set up a shared user home directory, by using the network file system (NFS) protocol and the Lightweight Directory Access Protocol (LDAP) server to get the user's details in setting up the user profiles.

With this shared home directory, USM Campus Grid users can log on to any node and have access to all their files. They are therefore able to initiate and submit jobs without the hassle and overhead of transferring required files between nodes.

CAMPUS GRID PORTAL

We have built a web portal for USM Campus Grid at `http://gclportal.usmgrid.myren.net.my` using the *GridSphere* platform. The *Campus Grid Portal* acts as an information centre for USM Campus Grid and Grid@USM's projects. Upon logging in, members may customize the portal by specifying groups that they wish to be in.

We describe other important features of the *Campus Grid Portal* in the following sections.

PROJECT PROGRESS UPDATES WITH TRAC WIKI

We have set up a *Trac* Wiki platform for the respective sub-project teams to post information and progress updates about their respective applications. This wiki serves as a project monitoring mechanism, as well as an invaluable archive for recording their experience using Grid technologies.

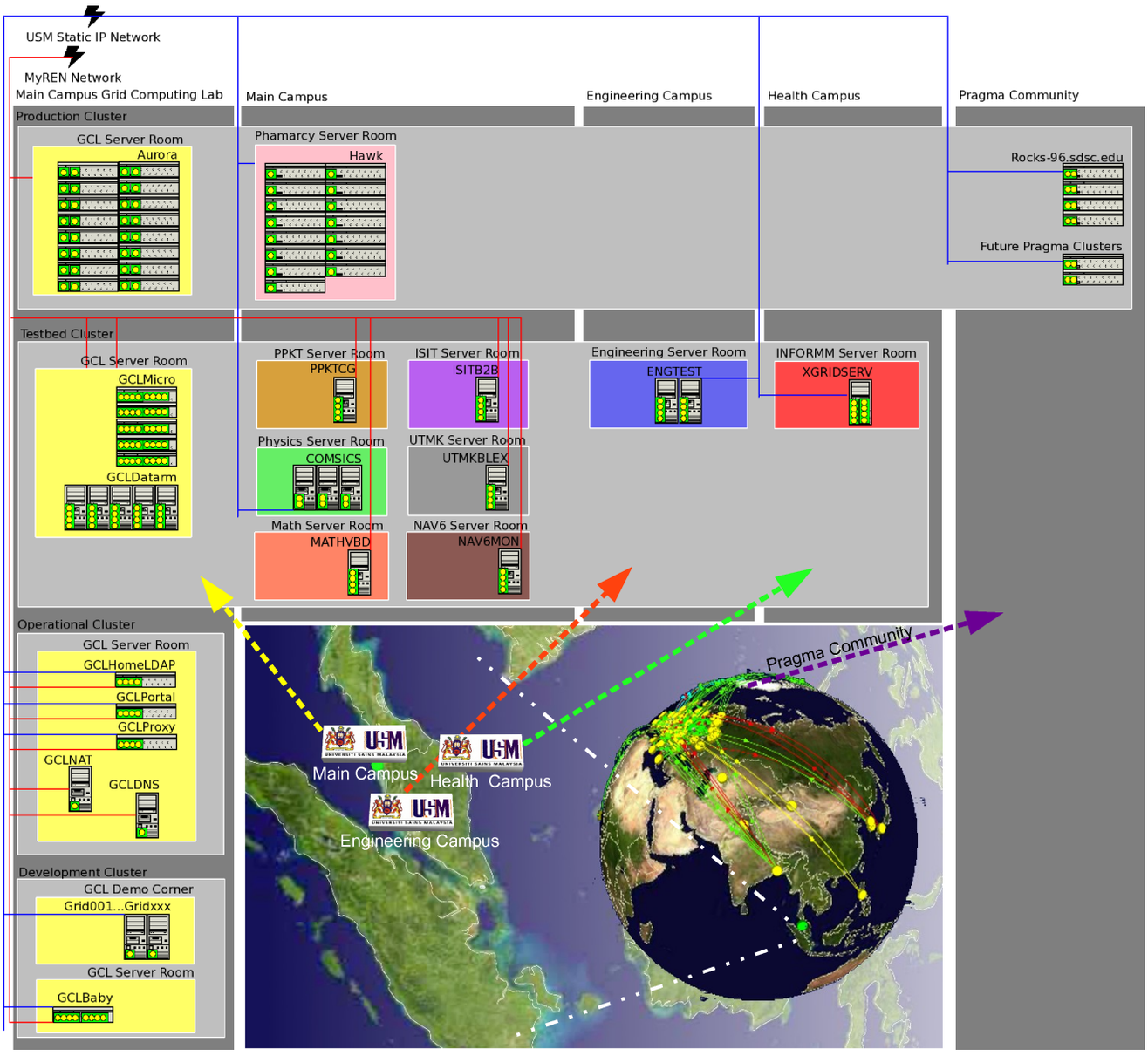


Figure 1: Simplified view of USM Campus Grid architecture

APPLICATION SHOWCASES WITH I-FRAME

Sub-project teams may showcase their grid-enabled applications in the *Campus Grid Portal* using a number of mechanisms. In cases where a Web graphical user interface (GUI) is available and presumably hosted on the sub-project team's node, this is just a matter of informing GCL researchers of the relevant URL. We would then embed the web interface as an *i-Frame portlet* in the *Campus Grid Portal*. In other words, the portal acts as a one-stop showcase of the sub-project grid-enabled applications.

Each member can specify the member group they wish to be in, thus controlling the i-Frame applications that will appear on their personalised portal pages upon logging in. This also provides an easier route for members to make use of the results from another team's application, i.e. via the portlets and without having to know the application's original URL. Currently, the UTMK and ISE project teams have incorporated Web GUI front-ends of their applications into the *Campus Grid Portal* as i-Frame portlets (Figure 2).

JOB SUBMISSION PORTAL

Job submission portlets were deployed to provide a user-friendly GUI for submitting jobs to the grid, as opposed to using the command line. Authorised users are only required to provide their passwords and the location of a executable file. At the click of a button, the job is submitted to a designated cluster. The portal will notify the user when the job is completed, and users may then download the results. In particular, we have created a dedicated job submission portlet to allow thePPSM team to submit their computation jobs to the Aurora cluster.

SINGLE LOGON

We implemented a single log-on system to provide transparent access to other grid resources in USM. The system currently allows *Campus Grid Portal* users to access the Centre for Knowledge, Communication &

Table 3: USM Campus Grid Users

User	Project
AP Phua Kia Kien	Using XGrid for Creating Render-Farm for 3D Animators
Lim Lian Tze	Grid-enabled Blexisma2
Prof. Koh Hock Lye	Grid Application to Wave Front Propagation and Containment of Vector Borne Diseases
Dr Kamal Zuhairi Zamli	An Automated Java Testing Tool on the Grid
Prof. Sureswaran Ramadass, AP Rahmat Budiarto	iNet Grid
Dr Vicent Khoo Kay Teong	B2B Standards Component Modeling
Atheer Supervisor: Dr Nur'Aini Abdul Rashid	MSc (Mixed Mode)
Adnan Supervisor: Prof. Sureswaran Ramadass	PhD
Salah Salem Supervisor: AP Chan Huah Yong	MSc (Mixed Mode)
Ahmad Abusnaina Supervisor: AP Chan Huah Yong	MSc (Mixed Mode)
Dr Yoon Tiem Leong	–
Ng Kok Fu Supervisor: AP Norhashidah Hj Mohd Ali	PhD, *Parallel Preconditioned Explicit Group Methods for Distributed Memory Multicomputers*
Koay Kok Teong Supervisor: AP Norhashidah Hj Mohd Ali	MSc (Research), *Parallel Numerical Algorithms*

Figure 2: An i-Frame portlet in *Campus Grid Portal*

Technology (PPKT)'s portal with a one-time sign-in, and vice versa.

SUPPORT & MAINTENANCE

To summarise, the GCL team acts as a support and maintenance team of USM Campus Grid and Grid@USM 's RU project. Among the tasks we have been (and are) carrying out include:

- managing USM Campus Grid machines located centrally at GCL, including hardware upgrades;
- managing and monitoring USM Campus Grid services, including management of users, credentials and computational resources;
- managing the *Campus Grid Portal* as an information centre and job submission portal;
- handling project management and administrative issues;
- supporting sub-project teams in grid-enabling applications, e.g.:
  - installing *Blender 3D* on GCLDatarm and creating a sample PHP page for SGE job submission for the INFORMM team;
  - preparing sample Fortran code, with and without enabling MPI, for the PPSM team;
  - enabling database connectivity between Microsoft SQL2005 and Linux clients using FreeTDS, C/C++ library scripts and binding with ASP.NET (C#) for the ISE team.

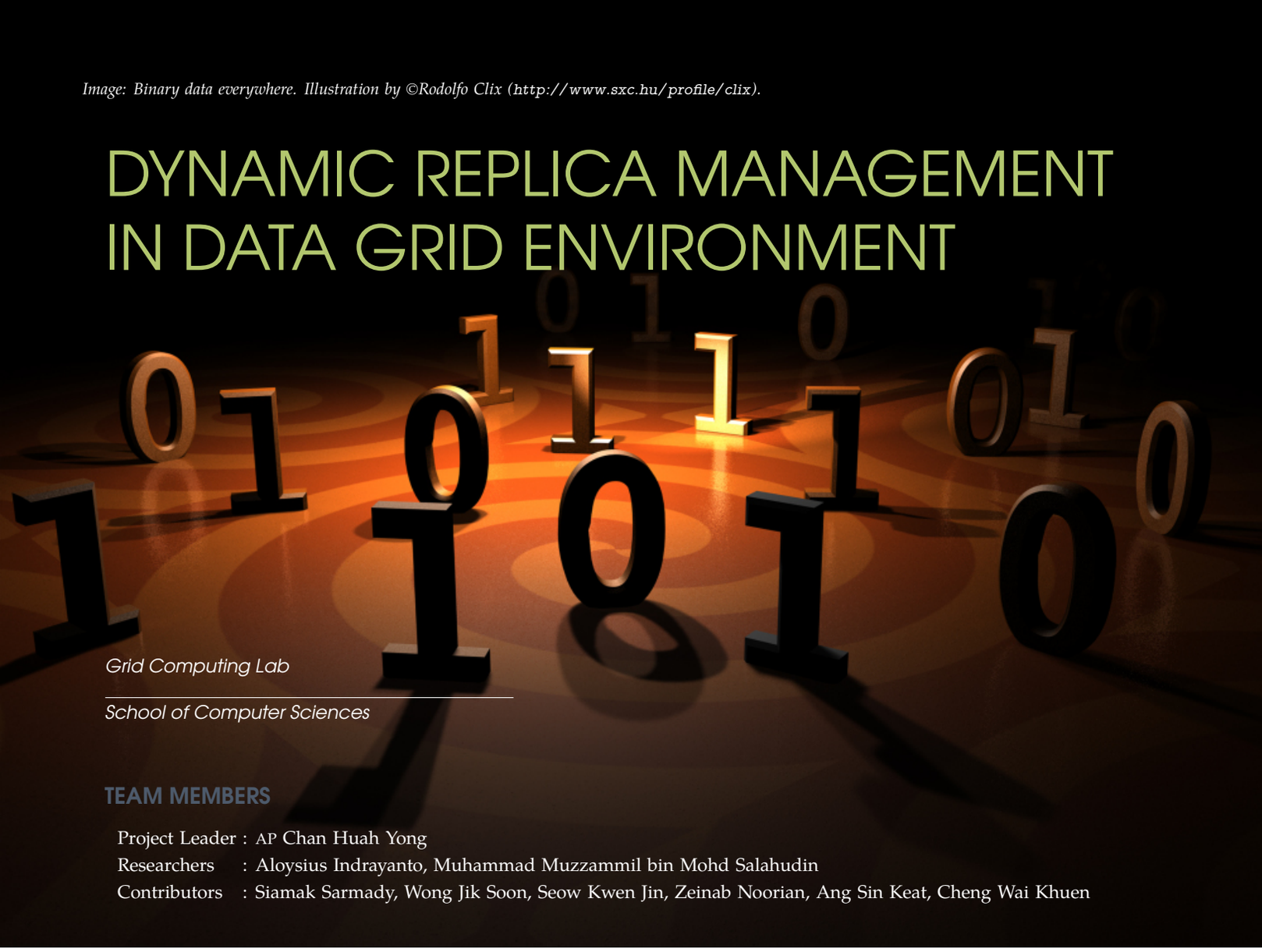


Image: Binary data everywhere. Illustration by ©Rodolfo Clix (http://www.sxc.hu/profile/clix).

DYNAMIC REPLICA MANAGEMENT IN DATA GRID ENVIRONMENT

Grid Computing Lab

School of Computer Sciences

TEAM MEMBERS

Project Leader : AP Chan Huah Yong
Researchers : Aloysius Indrayanto, Muhammad Muzzammil bin Mohd Salahudin
Contributors : Siamak Sarmady, Wong Jik Soon, Seow Kwen Jin, Zeinab Noorian, Ang Sin Keat, Cheng Wai Khuen

INTRODUCTION

Accessing, managing, querying, and programming a database system are normally dependent on the type of the database deployment, standards, conventions defined by the database vendor, and the language that will be used to define the query. Currently, the Relational Database Management System (RDBMS) is one of the most widely used database system. Query on these kind of databases are usually done by using a language named Structured Query Language (SQL).

RDBMS is currently implemented and supported by many different vendors. Every vendor has its own method or convention on how users should access its database system. This convention is defined as a set of programming procedure called Application Programming Interface (API). These differences would cause a developer (programmer) the need to learn many different API if he/she wants to use and integrate RDBMS deployments from multiple vendors.

SQL is a standardized language. However, the standard set of SQL commands normally are not enough for a vendor. Hence, database vendors would normally add non-standard extensions to the SQL so that users can use their RDBMS more efficiently. This extension would cause an administrator or a developer the need to learn many non-standard SQL commands. In addition, some vendors also provide a custom API which will need to be used if a developer wants an even more efficient access to the RDBMS. Some vendors only provide custom APIs to access the RDBMS' specific features (no extension to the SQL language).

Learning many different APIs and non-standard SQL extensions would not be efficient for a developer. Time could be wasted to factorize what can be done and what cannot be done with a particular RDBMS product/deployment. Hence, interoperating RDBMS from multiple vendors would be a maintenance problem.

Performing a custom distributed query between two different RDBMS that are installed in two different hosts (computers) would be impossible due to the fact that the vendors use their own proprietary protocols for communication between their RDBMS servers and clients. Also, not all RDBMS vendors support distributed query between two (or more) hosts. In addition, the cost of the software license and deployment for these kind of databases are expensive.

A large database vendors, such as Oracle and IBM DB2, provides supports for distributed query. Hence, their system is called Distributed Database System (DDS). However, the two systems may not cooperate well with each other. Hence, performing a distributed query between Oracle and DB2 would be difficult.

The VDDBMS project aims to provide a compatibility layer between different RDBMS implementations and the end users. The main objective is to allow distributed queries between two or more RDBMS from different vendors with the same SQL and API.

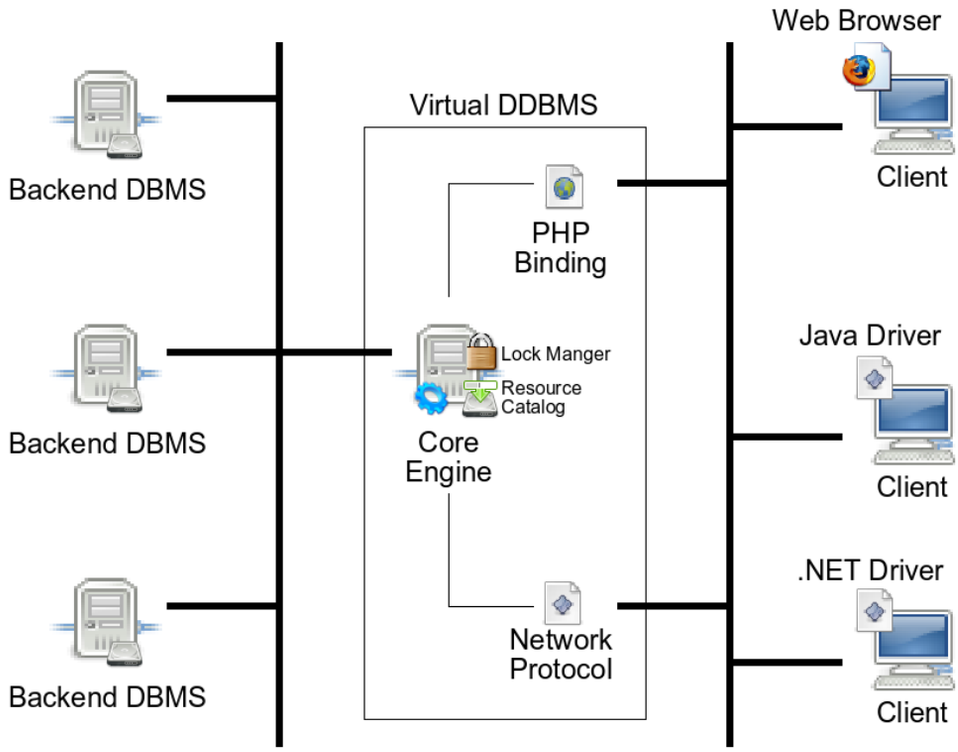


Figure 1: Current architecture of the VDDBMS system

METHODOLOGY

The methodology applied by the project to achieve its goal can be briefly described as follow:

1. An SQL parser has been implemented to parse the user-given SQL statements into tokens (symbols and data).
2. A resource locator has been implemented to add references to the tokens to record the hosts (data sources) on which the SQL would need to be executed.
3. A token splitter has been implemented to split the tokens into sub-tokens. These steps are necessary because a data source does not have to execute the complete tokens.
4. An instruction sequencer has been implemented to execute to sub-tokens in the appropriate data sources in the correct sequence.
5. A SQL generator has been implemented to translate back the sub-tokens into SQL statements understandable by the target data sources.
6. A post-processor has been implemented to aggregate back the sub-results from the data sources into the final result.

DESIGN & IMPLEMENTATION

The system can be separated into four main modules:

1. Core engine;
2. Backend drivers for each RDBMS vendor to be supported;
3. Language binding;
4. Network server and client drivers.

The system is mainly developed using C++ in Linux. Figure 1 shows the current architecture of the system.

The core engine interacts with the backend RDBMS using the specific drivers. The core engine is working together with resource catalogue and lock manager to process user requests. User can connect to the VDDBMS system either locally or remotely. Currently local connection is supported by native API in C++ and PHP. Remote connection is supported by network API in C++, PHP, Java, and .NET.

Figure 2 shows the modules in the VDDBMS system with their simplified dataflow. In the figure the solid lines defines communication path which are done using API. The dashed line defines communication path which are done using network protocol.

Most of the API communication paths are private. Only the C++ and PHP public interfaces are usable by the user. Hence, changes in the internal API would not raise incompatibility issue with user applications.

The network (VDDBMS) server and the client drivers (C++, Java, .NET) are completely independent. As long as the same protocol is used, changes in the server would not affect the client. The protocol has been designed to be extensible. Hence, a newer server would be able to manage older client driver implementations.

Figure 3 shows the high level flow of the VDDBMS engine while Figure 4 shows the detailed flow of the core execution engine.

The main features of the current system are:

1. Data replication and horizontal fragmentation.
2. Support for simple SQL `INSERT`, `DELETE`, `UPDATE`, and `SELECT` queries.
3. Prepared statement for SQL `INSERT`.
4. Native `DATE/TIME` field support.
5. Stored procedure with support for looping, conditional statements, simple calculations, and atomic integer operations.
6. MySQL, PostgreSQL, Microsoft-SQL, and Oracle are supported as the backend RDBMS (data sources).

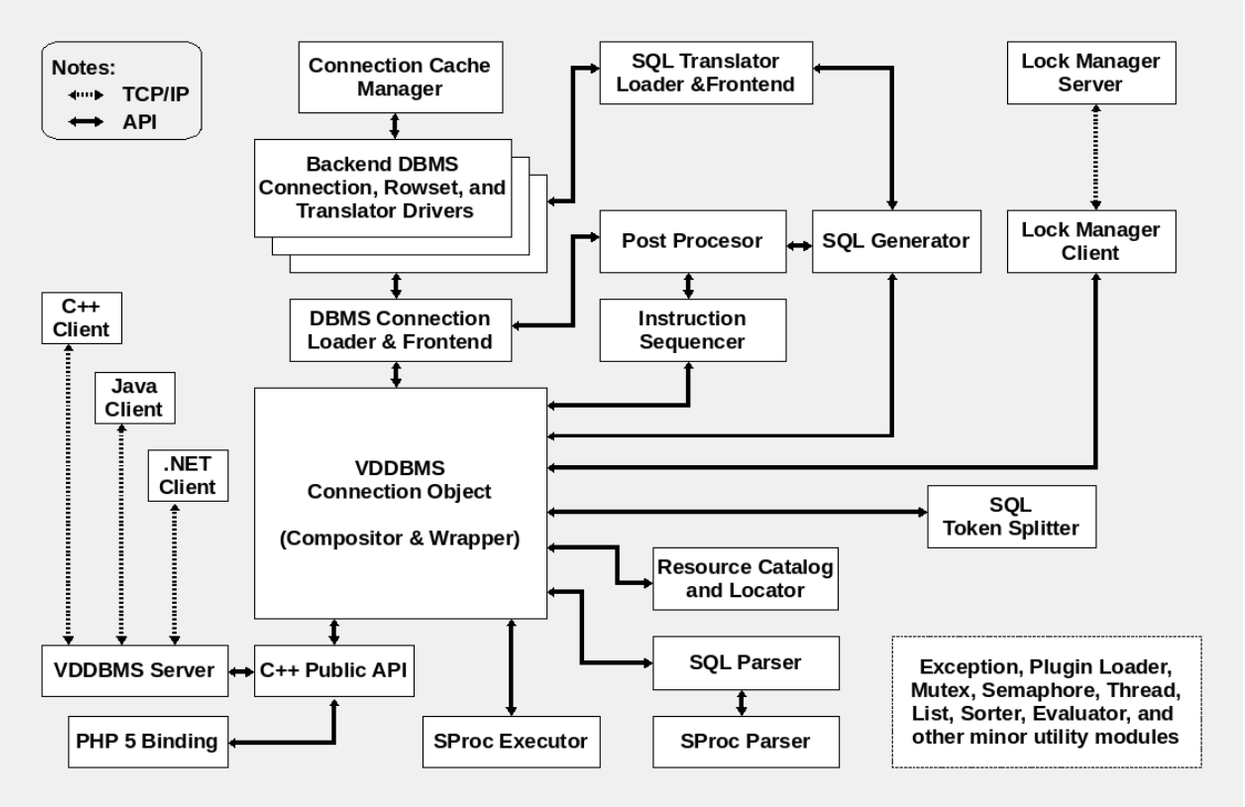


Figure 2: Modules of the VDDBMS system and their simplified dataflows

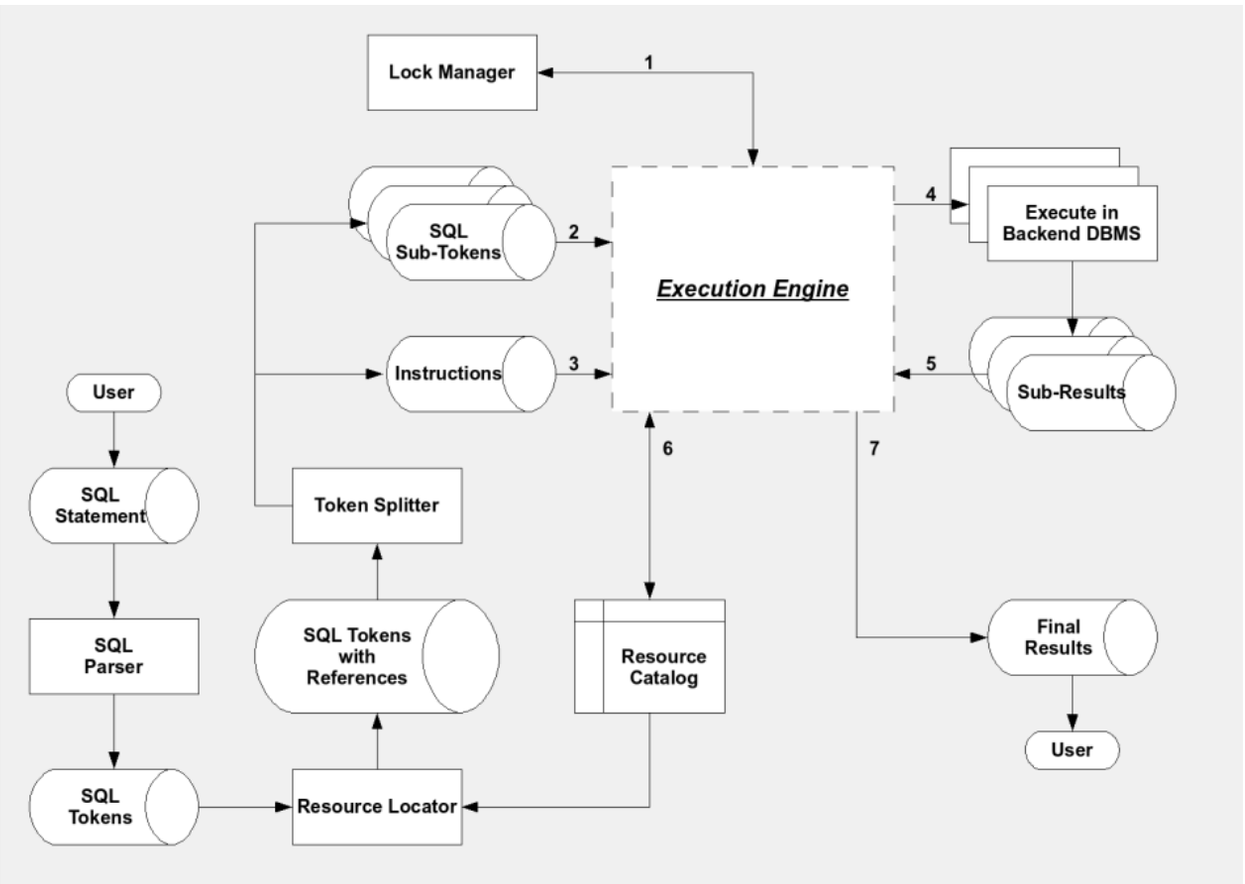


Figure 3: Flow of the VDDBMS Engine

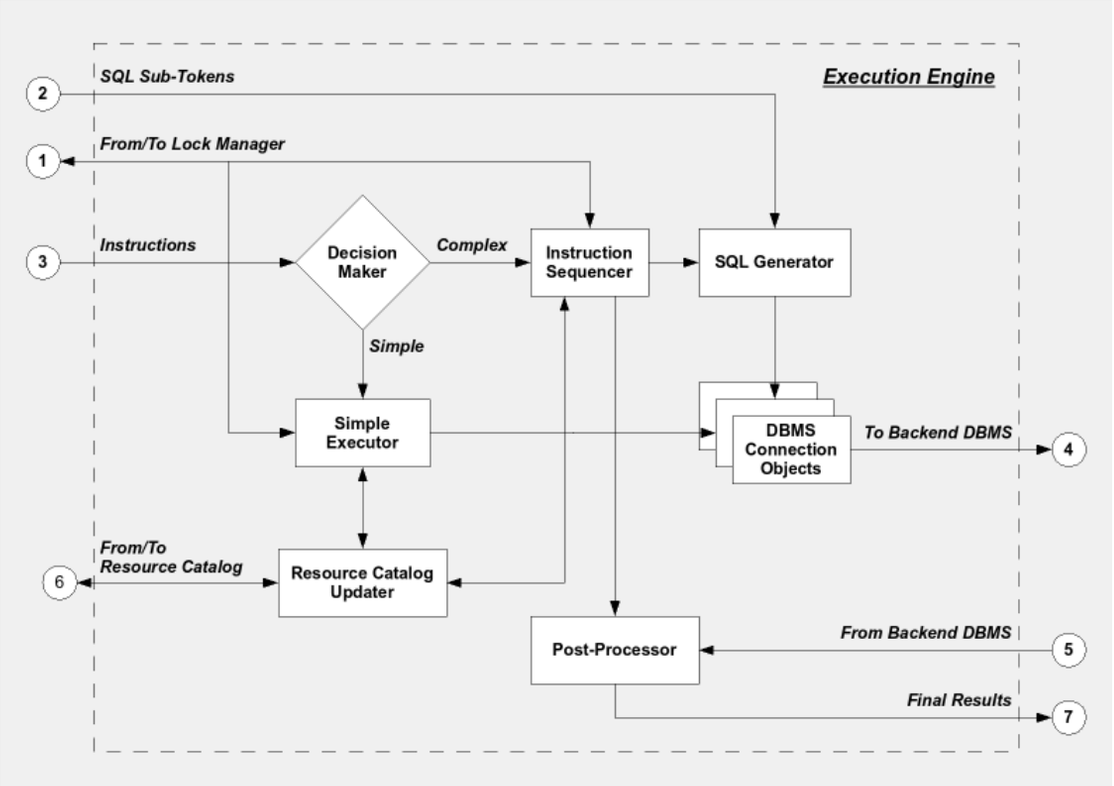


Figure 4: Detailed Flow of the VDDBMS Execution Engine

7. Plugin support (backend RDBMS driver and user-implemented-function in stored procedure).
8. C++ native API.
9. PHP, Java, and .NET binding.
10. Network server is implemented on top of the core engine with user authentication support.

RESULTS & CONCLUSION

The VDDBMS system is able to:

- Display multiple backend RDBMS as if they are one single, virtual database host.
- Perform database and table managements in multiple backend RDBMS.
- Perform queries and updates in multiple backend RDBMS.

Compared with a standalone RDBMS system such as MySQL, the VDDBMS system performance is about 10 to 100 times slower. However, it has the advantage in that it can perform distributed query on multiple RDBMS hosts from different vendors.

FUTURE WORK

- Distributed resource catalogue and distributed lock manager.
- Support for aggregation (bottom-up) approach. In this mode, existing database deployments can be integrated as one virtual database. The user would still be able to use the existing deployments as independent system as well as aggregate them with the VDDBMS system.
- Support for error recovery.
- Performance optimization.
- Integration of a more intelligent algorithm on top of the system for automated data replica or fragment management using an already published/implemented work.

CONTRIBUTION TO THE OPEN SOURCE COMMUNITY

This work has been published as an Open Source Software (OSS) project via the SourceForge code repository. The project homepage can be accessed from the URL `http://vddbms.sourceforge.net`.

PROJECT PUBLICATIONS

Al-Mistarihi, H. H. E. & Chan, H. Y. (2009). On Fairness, Optimizing Replica Selection in Data Grids. *IEEE Transactions on Parallel & Distributed Systems*. (forthcoming).

Al-Mistarihi, H. H. E. (2009). A Data Grid Replica Management System with Local and Global Multi-Objective Optimization. PhD thesis, School of Computer Sciences, Universiti Sains Malaysia.

Chan, H. Y. & Al-Mistarihi, H. H. E. (2008). Replica Management in Data Grid. *International Journal of Computer Science and Network Security*.

Chan, H. Y. & Al-Mistarihi, H. H. E. (2008). Replica Optimization Service in Data Grids. *Sciences Publications*.

Indrayanto, A. (2009). Initial File-Placement in Data Grid Environment using Game Theory and Fictitious Play. MSc thesis, School of Computer Sciences, Universiti Sains Malaysia.

Indrayanto, A. & Chan, H. Y. (2008). Application of Game Theory and Fictitious Play in Data Placement. In *Proceedings of the International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008)*. Penang, Malaysia; pp. 79–83.

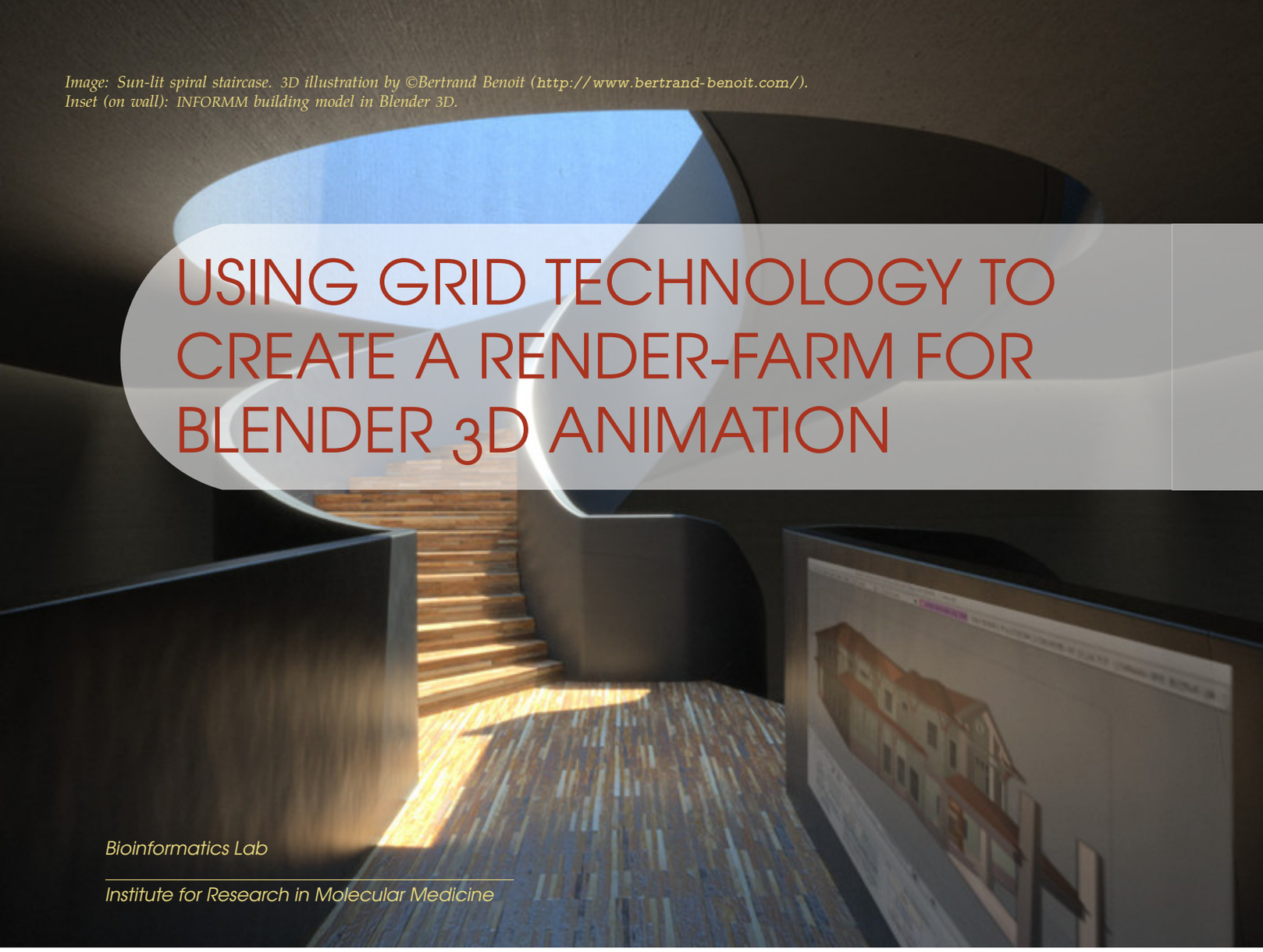


Image: Sun-lit spiral staircase. 3D illustration by ©Bertrand Benoit (http://www.bertrand-benoit.com/).
Inset (on wall): INFORMM building model in Blender 3D.

USING GRID TECHNOLOGY TO CREATE A RENDER-FARM FOR BLENDER 3D ANIMATION

Bioinformatics Lab

Institute for Research in Molecular Medicine

TEAM MEMBERS

Project Leader : AP Phua Kia Ken
Researcher : Zafri Muhammad

INTRODUCTION

Grid computing is becoming more popular today as computers come equipped with multi-core CPUs. Together with innovative operating systems, multi-threaded and parallel processing, a.k.a. grid computing, has become more common, especially for applications that require intensive computations. An attractive feature of grid computing is that it can utilize idle CPUs connected together in a network to create a virtual supercomputer. 'Xgrid' is a distributed computing technology introduced by Apple™ Inc. to create a grid network for parallel processing. Applications that can be controlled via the command-line interface of the Unix operating system should be able to benefit from Xgrid. One such application is Blender 3D. Blender 3D was selected in this research because it is an open source 3D modelling and animation software, and more importantly because it has an expansive architecture that can be extended using Python programming language. Blender 3D version 2.46 is also capable of multi-threading, which enable one to tap into the multi-core capabilities of modern CPUs in addition to the power of grid computing.

Recently, several public-domain animation movies have been produced that showcase Blender 3D's capability as a professional animation tool. However, the rendering time needed for professional animations take considerable time to complete using a conventional computer setup. To overcome this limitation, we explore the use of grid computing to create a Render-Farm that will reduce the burden of rendering Blender 3D animations. To encourage end-users (3D animators) to use the Render-Farm, we explored the use of PHP to develop a web portal for easy access and retrieval of rendered results.

OBJECTIVE

To create a Render-Farm using Xgrid technology for Blender 3D in order to speed up the processor-intensive rendering and compression of animation files for Blender 3D animators.

RESEARCH METHODOLOGY

In our grid environment, we used 10 Apple iMac computers that were available at the Bioinformatics Lab,

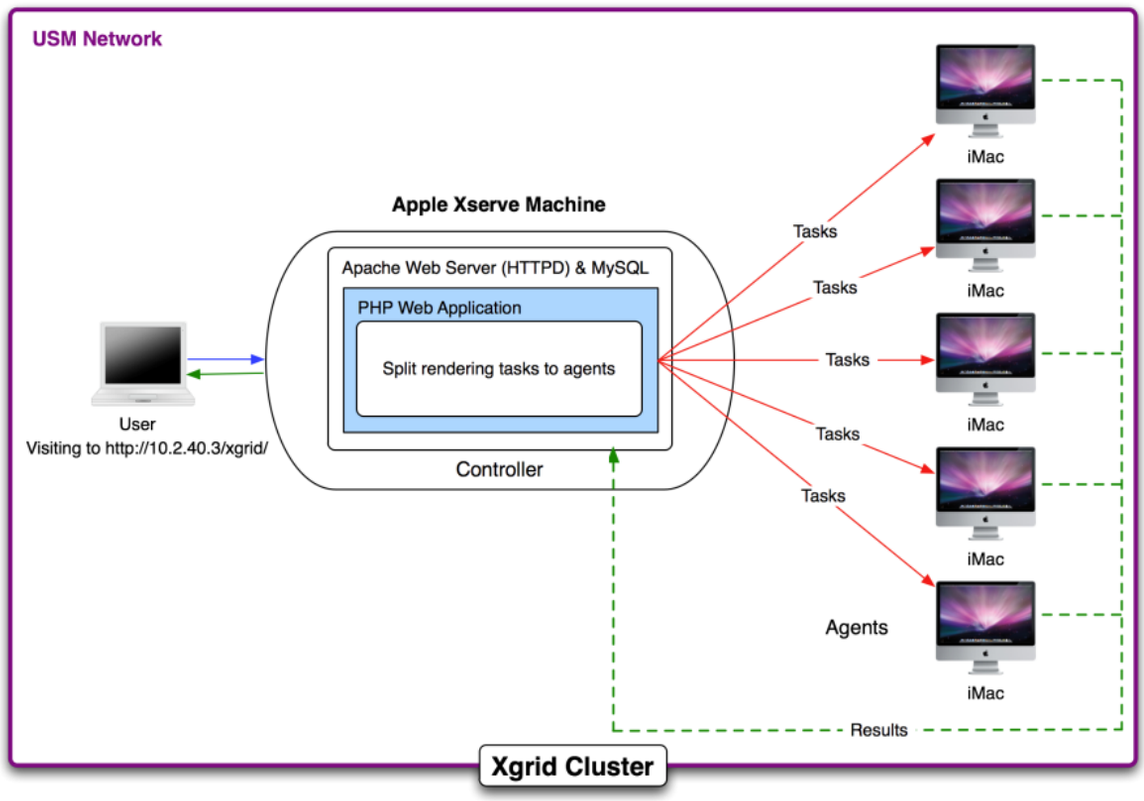


Figure 1: System Architecture showing system flow for Blender 3D Render-Farm Portal developed at INFORMM, USM.

Figure 2: Xgrid Admin showing the combined CPU power of the grid obtained by networking 10 iMac 2.0 GHz (Intel Core 2 Duo CPUs) in a cluster using Apple Xgrid.

Institute for Research in Molecular Medicine (INFORMM), USM. All Apple iMac computers were equipped with Mac OS 10.5, Intel Core 2 Duo 2.0 GHz processors and 1 GB of RAM. These were connected through the USM local area network (Figure 1). An Apple Xserve was used as a controller, and the Apple iMacs served as agents in the grid. A total of 40 GHz CPU power was successfully attained on the grid cluster when all 10 iMac computers in the cluster were used (Figure 2).

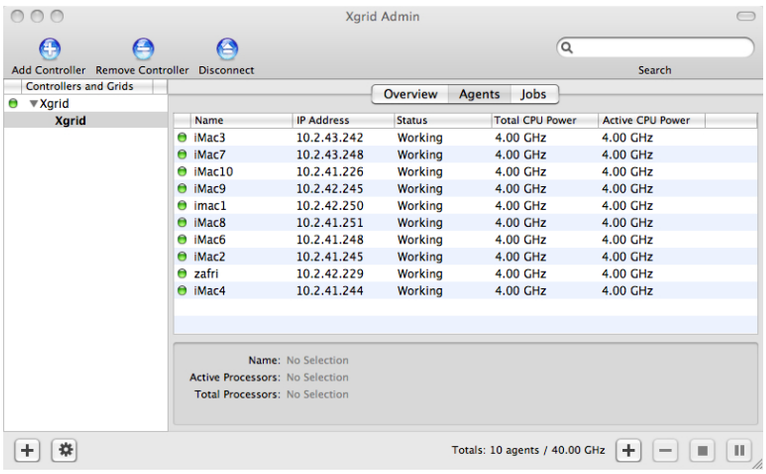
Blender version 2.46 was downloaded from the Blender website (http://www.blender.org) and installed on all agents (see Figure 5(a) for screenshot). A Blender 3D Render-Farm portal was developed using PHP and MySQL to provide easy access for users to submit their Blender 3D rendering jobs (animation and textures files; Figure 3(a)), and to retrieve the rendered results (individual frames) from the controller (Figure 3(b)).

Users can simply 'submit' their jobs by logging-in to the Render-Farm portal; upload their animation and texture files and fill in the start-frame and end-frame in the fields provided (Figure 3(a)). Once the job is submitted, the controller splits the job into multiple tasks and then distributes it to any available agents on the grid. The submitted job contains a Job ID and is saved in the MySQL database running on the server. During the rendering process, Xgrid Administration Tool was used to log-in to the Xgrid controller to monitor its activities (Figures 4(a), (b) and (c)). The easy-to-use interface provides a clean summary of the cluster activities, such as the

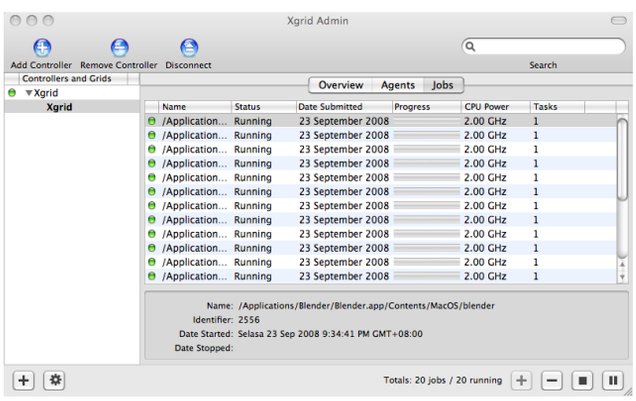
(a) **Job Submission Page.** Users only need to upload their animation file and textures. The render job is given a name and the 'start-frame' and 'end-frame' fields are keyed in before clicking on the 'Submit' button to send the job to the controller, which then splits and delegates the job to the agents on network.

(b) **Results Page.** When the job is completed, the agents return the results to the controller when they are stored in compressed form as a zip file. Users only need to click the 'Download' button to return the results from the controller.

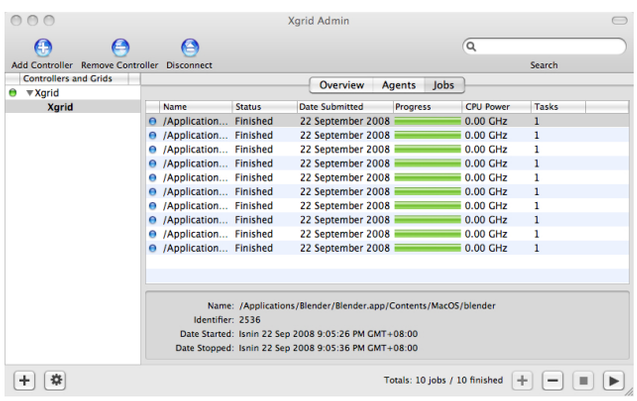
Figure 3: Screenshot of the Render-Farm portal



(a) List of agents available in the grid

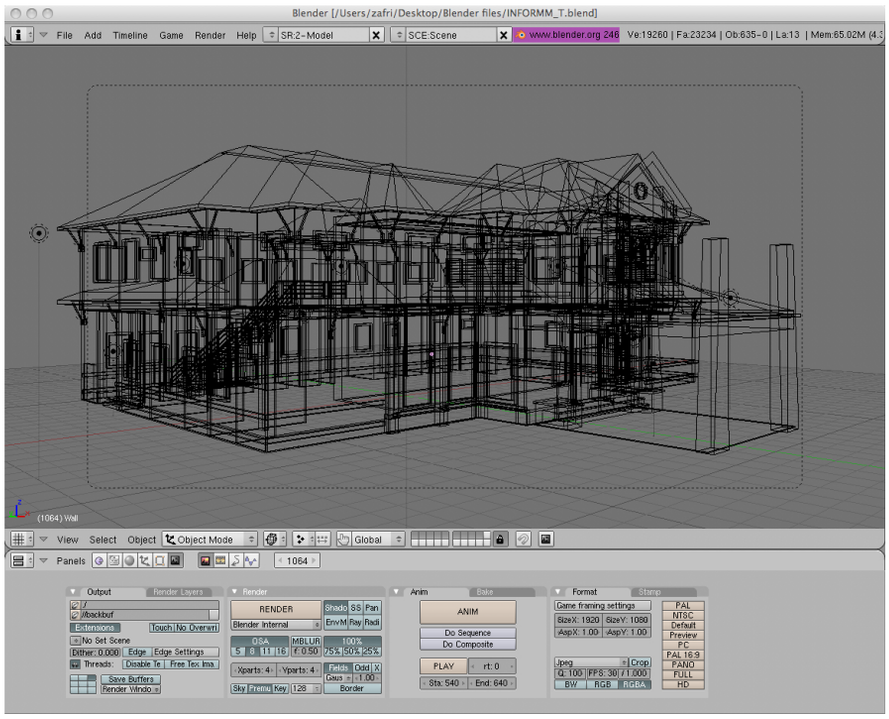


(b) Jobs Status Monitor (Running jobs)

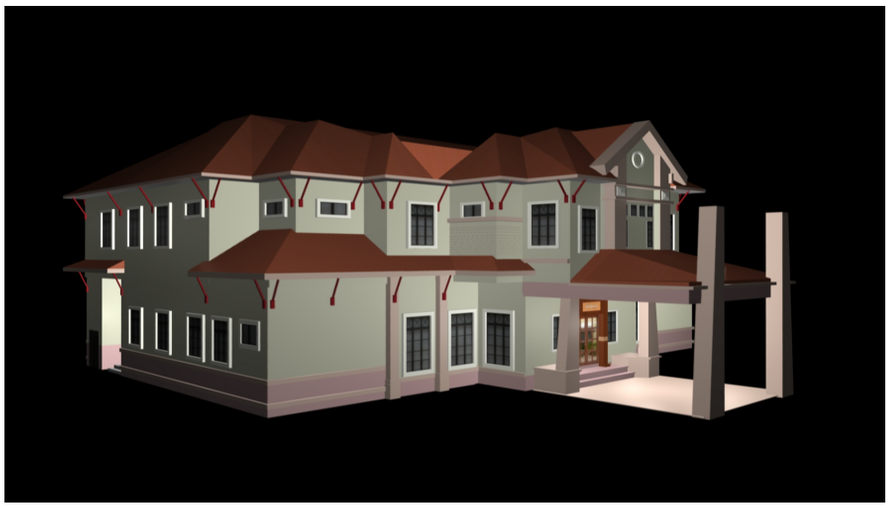


(c) Jobs Status Monitor (Finished jobs)

Figure 4: Xgrid Admin Tools



(a) Wireframe rendered image of INFORMM building at HD resolution (1920 × 1080 pixel) for Blender 3D. Model took less than 1 second to render and save compressed as a JPEG image file.



(b) Final rendered image from Blender 3D using radiocity ray-tracing. Image took 80 seconds to render 1 still frame. 24 still frames are needed to produce 1 second of animation.

Figure 5: Images rendered by Blender 3D

job status and how many jobs are currently running and/or pending, CPU usage, and much more.

On completion of the rendering job, the agents will send back the rendered results (series of static image files) to the controller. The controller then collects the images into a temporary folder and then proceeds to compress each rendered file in the folder. The temporary folder is then packaged as a zipped file and stored in the portal, ready for download. The user can then download the zipped file to his terminal for post processing, such as QuickTime stitching together and temporal or spatial compression to create a .mov file ready to play.

RESULTS

The time periods taken for rendering 10,000 frames of Blender 3D animation on the Render-Farm containing 1 to 10 iMac computers were recorded. The result showed that there is a huge difference in completion time between using one computer and the Render-Farm running on 10 agents. Xgrid with 10 computers (20 CPUs) running a job consisting of 10,000 frames of Blender 3D animation took 2340 seconds (39 minutes) compared with 20,160 seconds (336 minutes) on a single computer (2 CPUs). This represents a 9 fold increase in rendering speed.

Table 1: Reliability of Xgrid Render-Farm

Run No. \ No. of nodes	Time Taken (Seconds) To Complete Rendering 1000 Frames									
	1	2	3	4	5	6	7	8	9	10
1	2435.00	1306.90	963.42	663.78	569.78	435.71	422.64	355.13	327.98	305.87
2	2436.00	1336.00	963.35	663.33	569.40	436.00	421.00	354.00	327.66	303.00
3	2435.00	1306.90	963.42	663.78	560.78	435.71	422.64	355.23	327.98	305.87
4	2435.00	1306.90	963.42	663.78	569.78	435.71	333.00	355.13	327.98	305.27
5	2435.00	1306.90	963.33	660.00	564.00	432.00	421.00	350.95	321.22	301.17
coefficient of variation (%)	1.848	3.061	4.152	0.002	0.007	0.003	0.002	0.005	0.009	0.006

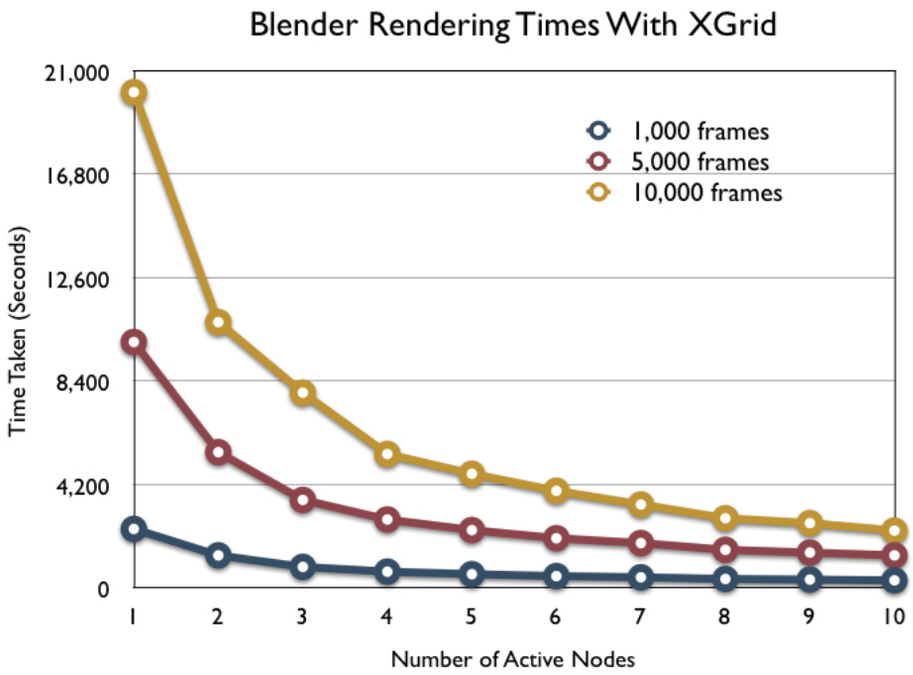


Figure 6: Blender 3D rendering times using Xgrid and increasing number of agents

DISCUSSION & CONCLUSION

In this project, we constructed a grid network of 10 iMac computers using Apple Xgrid middleware to create a Render-Farm for Blender 3D animation. A web portal was also developed using PHP to enable easy submission of render jobs to the grid and to retrieve the results of the rendering. To study the efficiency of the grid, we recorded the time taken to render 1000, 5000, and 10,000 frames of Blender 3D animation with a fixed number of agents in the grid for each study. The results showed that when the number of agents were increased from 1 to 10, the rendering times were reduced by 86.78 %, 86.68 % and 88.39 % for 1000, 5000 and 10,000 frames, respectively. On average the 10-node Render-Farm was able to complete the rendering job by as much as 9 times faster than the conventional single computer setup. As the workload increases, the efficiency also increases, although larger jobs also increase the time taken to return the results to the controller.

To study the reliability (reproducibility) of the Xgrid Render-Farm, we repeated 5 times each run across a fixed number of agents and found that the coefficient of variation of times taken for completing of each rendering job was less than 5 % (Table 1). We conclude that Xgrid can be used to create a reliable low-cost Render-Farm to accelerate CPU intensive tasks, saving time and cost.

FUTURE WORK

Following the PRAGMA 15 exhibition and showcase, we have received many queries regarding our Blender 3D Render-Farm portal. We are preparing to organise a workshop on Blender 3D and grid computing.

PROJECT PUBLICATIONS

Phua, K. K. & Wong, V. C. (2007). Developing A Compelling 3D Animation and Multimedia Presentation Using Blender for INFORMM's Opening Ceremony. *Malaysian Journal of Medical Sciences*, *14*(Supplement 1).

Zafri, M., Sanjay, K. C., & Phua, K. K. (2009). Implementation Methods for Estimating Haplotypes with GRID Computing Technology. In *Compendium of Abstracts for the 14th National Conference on Medical and Health Sciences (NCMHS)*.

Zafri, M., Fadhilah, N. K., & Phua, K. K. (2008). Using Xgrid to Improve FASTA Efficiency for Alignment of Multiple DNA and Protein Sequences. *Malaysian Journal of Medical Sciences*, *15*(Supplement 1).

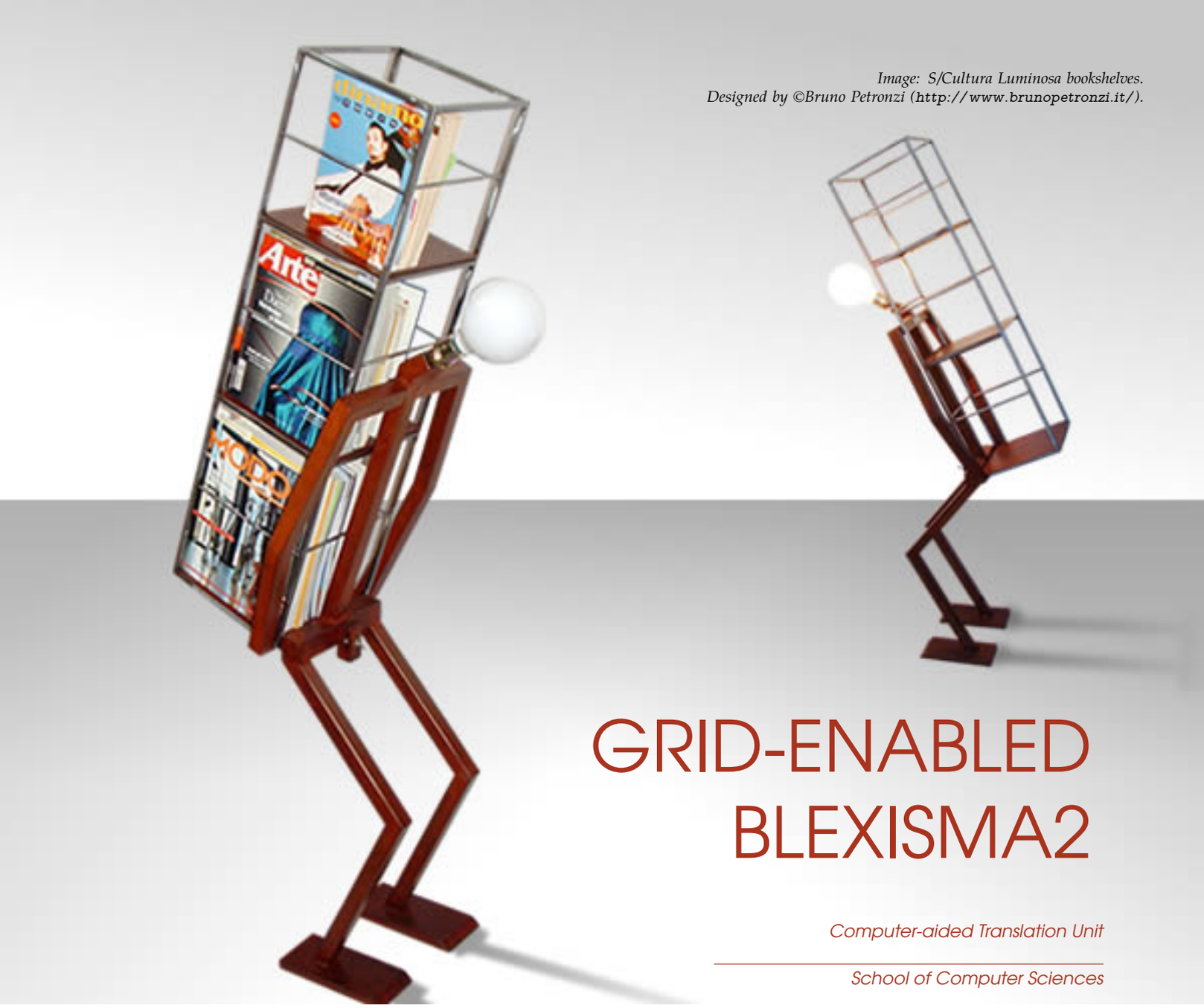


Image: S/Cultura Luminosa bookshelves.
Designed by ©Bruno Petronzi (http://www.brunopetronzi.it/).

GRID-ENABLED BLEXISMA2

Computer-aided Translation Unit

School of Computer Sciences

TEAM MEMBERS

Project Leader : AP Tang Enya Kong
Researchers : Lim Lian Tze, Ye Hong Hoe
Collaborator : Dr Didier Schwab,
Groupe d'Étude en Traduction Automatique/Traitement Automatisé des Langues et de la Parole (GETALP),
Laboratoire d'Informatique de Grenoble, Université Pierre Mendès France (Grenoble 2), France

NATURAL LANGUAGE PROCESSING & GRID COMPUTING

Efforts to create computer programs that can understand human languages, or *natural languages*, has a long tradition dating back to the invention of the computer. Much work is still on-going in the field of Natural Language Processing (NLP) today.

On one hand, there are still many pertinent research questions as natural language is intricate, ambiguous, dynamic and continuously evolving. On the other hand, novel needs for NLP applications are discovered everyday as people from different communities interact in this increasingly globalised world, across national, linguistic and cultural borders. Machine translation, automatic summarisation, intelligent search and targeted advertising are some examples that spring to mind. Suffice to say that NLP is an exciting research field to be in.

NLP systems are notorious for being resource-hungry. Intricate grammatical structures, large vocabularies, the complex interactions between these linguistic layers, as well as the sheer amount of documents to be processed, all call for high processing power, speed and storage. Distributed processing in grid environments have proved to be a helpful solution: the Tsujii Lab at the University of Tokyo had successfully performed deep grammatical parsing of the entire MEDLINE corpus[1] in 8 days on PC

[1] MEDLINE is an online database of 11 million citations and abstracts from health and medical journals and other news sources. It contains about 10 GB of uncompressed data.

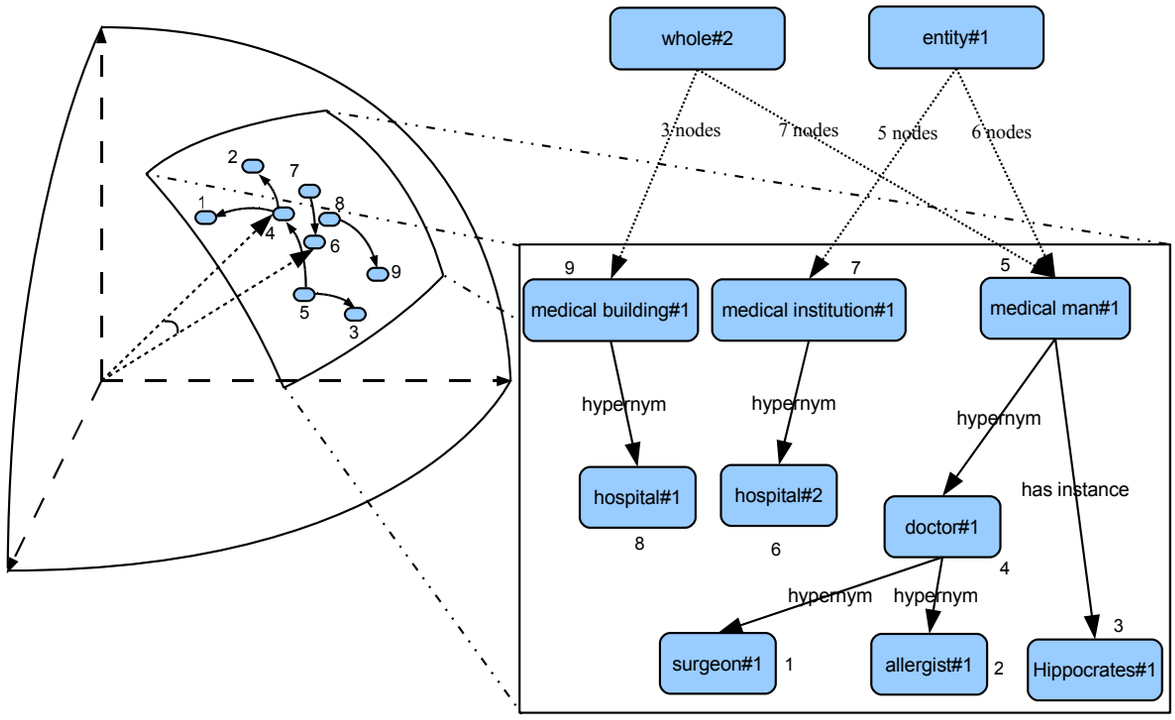


Figure 1: Discrete relations and non-discrete neighbourhoods of lexical items and their meanings. Hypernymy (the *'is-a'* relation) links *surgeon* to *doctor* etc, but does not link *hospital* to *surgeon* nor *doctor* directly. Instead, all lexical items related to the medical profession are within nearby neighbourhood of each other by the conceptual vector model.

clusters consisting of 340 CPUs. Closer to home, we had previously deployed our machine translation system on a grid cluster for load balancing.

An NLP application would need to understand – or *seen* to be capable of understanding – 'meaning' as conveyed by natural language discourse, if it was to perform anything deemed 'intelligent'. While the exact definition of 'meaning' is still debated among linguists and philosophers, we attempted to create a simple computational representation of word (*lexical item*) meanings. This model is used by *Blexisma2*, our distributed multi-agent system, to construct a semantic lexical database and to perform semantic processing of natural language texts.

REPRESENTING LEXICAL MEANING WITH COMPUTERS

We designed a Semantic Lexical Base (SLB) to store computer representations of word meanings. The meaning of a word or lexical item can be modelled by its relations to other lexical items. For example, the English word *heavy* is used to convey the intensity of *rain* – we do not say *big rain* or even *strong rain* in English. In addition, *rain* intuitively calls to mind other words like *snow*, *sunny*, *wet*, *water*. We model such explicit linguistic relations with discrete links between lexical items (and between their meanings), and implicit 'conceptual neighbourhood' with mathematical vectors. Essentially, the conceptual vector model projects words and their meanings onto a spherical space, in which meanings with similar 'conceptual themes' (e.g. WATER, WEATHER) are located close to each other. Figure 1 illustrates how this hybrid representation scheme captures multiple facets of lexical item meanings.

The SLB is designed to handle *polysemy*, i.e. that a surface lexical form (e.g. *mouse*) can have multiple meanings (a small, furry animal; a computer accessory; a timid person; etc). In other words, the SLB will store a record for each acknowledged meaning. One of the key aims of *Blexisma2* is to automatically acquire as many meaning records (called *acceptions*) as possible from multiple sources, including dictionaries, terminology lists and Web articles. We chose to use multiple learning sources because no single source can claim exhaustive coverage of a language's lexicon. Moreover, if the information or definition given by a particular source is badly written to the point of introducing noise, the negative effect can be automatically reduced when data of better quality from another source is processed. Therefore, we also store a record (called a *lexie*) for each meaning entry acquired from a different source. The overall SLB organisation is shown in Figure 2.

Blexisma2 is expected to run continuously and iteratively, so that newly-coined terms or new usages of existing words will be captured in the SLB as they appear. The quality of acception and lexie objects will also be refined with each iterative cycle, as each refined record would 'feed' positive inputs into the learning process of other lexical meanings. We implemented the SLB as a PostgreSQL database, hosted on UTMKBLEX in the USM Campus Grid.

BLEXISMA2 AGENTS

Blexisma2 is a distributed multi-agent system, implemented with the *MadKit* platform (`http://www.madkit.net`). Each *Blexisma2* agent has a specific role or function to perform. They may communicate with each other by sending messages, managed by the *MadKit* kernel. An agent may request the help of other agents to complete its task. Conversely, an agent may

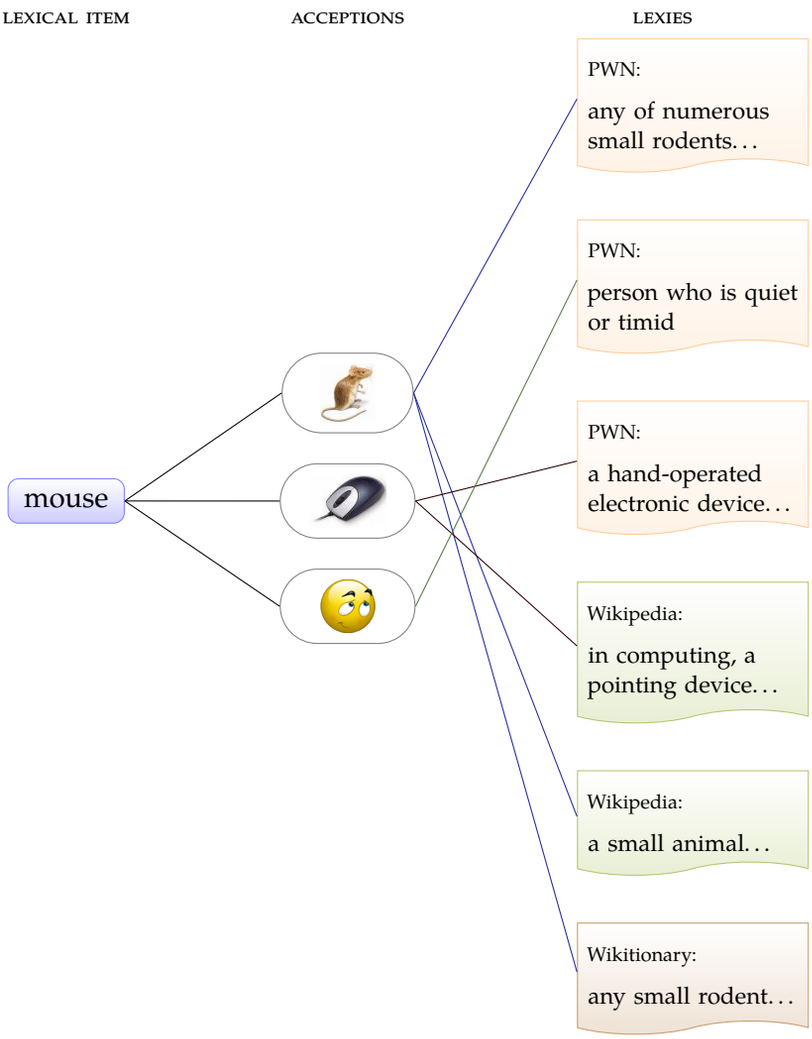


Figure 2: Schematic organisation of lexical items, acceptions and lexies in the SLB. Lexies are mined from various sources, and are then aligned or clustered into acceptions.

respond to other agents' requests by supplying information after performing its own processing, or setting off further chains of agent interactions.

The first implemented agents in our *Blexisma2* system prototype attempts to compute acception and lexie objects for English lexical items, based on Princeton WordNet (PWN), a wide-coverage, freely available online lexical database for the English language. PWN Analyser agents would request input data from Lexicon Dispenser agents, whose role is to interface with the SLB on behalf of all other agents. After performing their computations, the Analysers would pass their results to the Dispensers to be stored in the SLB. (See Lim and Schwab 2008; Schwab and Lim 2008 for details of the computation method). We deployed instances of these agents on the USM Campus Grid using the *Globus* job submission toolkit (Figure 3), thus freeing up UTMKBLEX to devote its resources to the running and maintenance of the SLB PostgreSQL server.

RESULTS AND APPLICATION

We were able to process as many as 1,902,080 lexical items in 5 days, averaging around 263 items per minute. Performance-wise, we noted bottleneck effects i.e. hitting the performance ceiling at around 8 nodes due to frequent connections, high I/O and overall heavy load on the SLB, which we aim to overcome in the future.

To demonstrate how the SLB might be used, we also implemented a simple Sense-Tagger and a Syntactic Parser. The Sense-Tagger takes an English sentence as input, and requests the Syntactic Parser to analyse the grammatical properties of the words in the sentence. It then uses semantic data from the SLB to identify the most likely meanings of each word. For instance, when given the inputs

(1) Spica is a *star* in the Virgo constellation.
(2) She is the *star* actress in our play.
(3) A *school* of fish swam past.
(4) The *school*'s classrooms are spacious.

the Sense-Tagger identifies *star* as a celestial body in (1), and as an actor playing a principal role in (2). *School* in (3) is tagged as a large group of fish; and as an education institution in (4). This capability to discern between different meanings of a word in different contexts is important in helping other NLP applications provide accurate results.

WHAT'S NEXT?

While the *Blexisma2* prototype caters only for the English language, we are

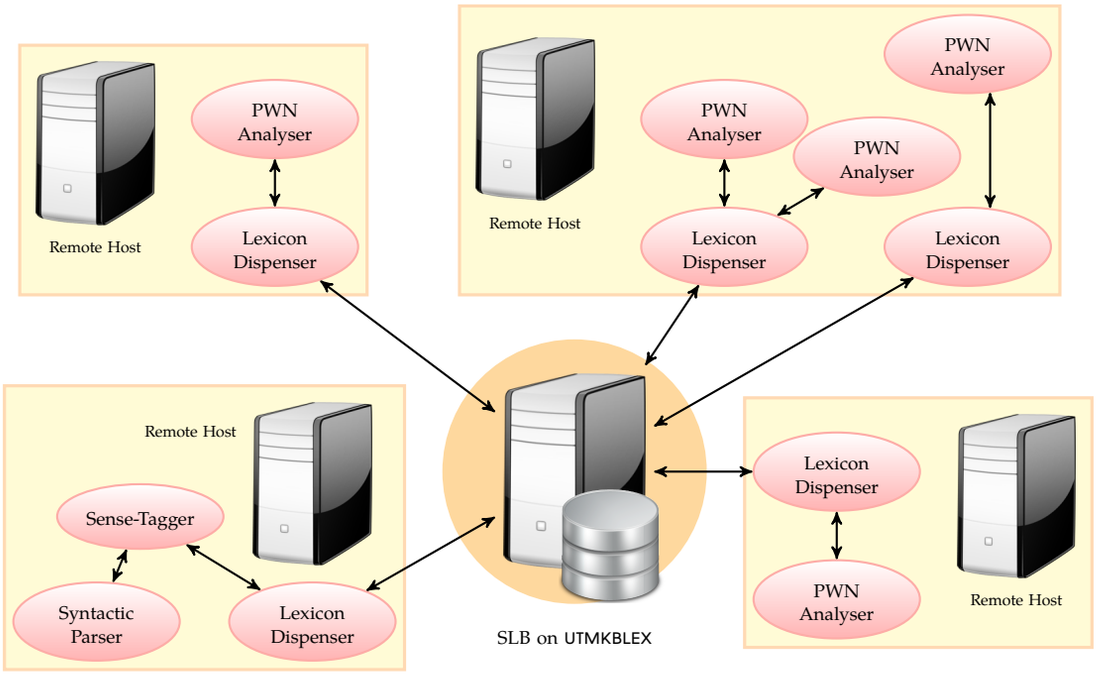
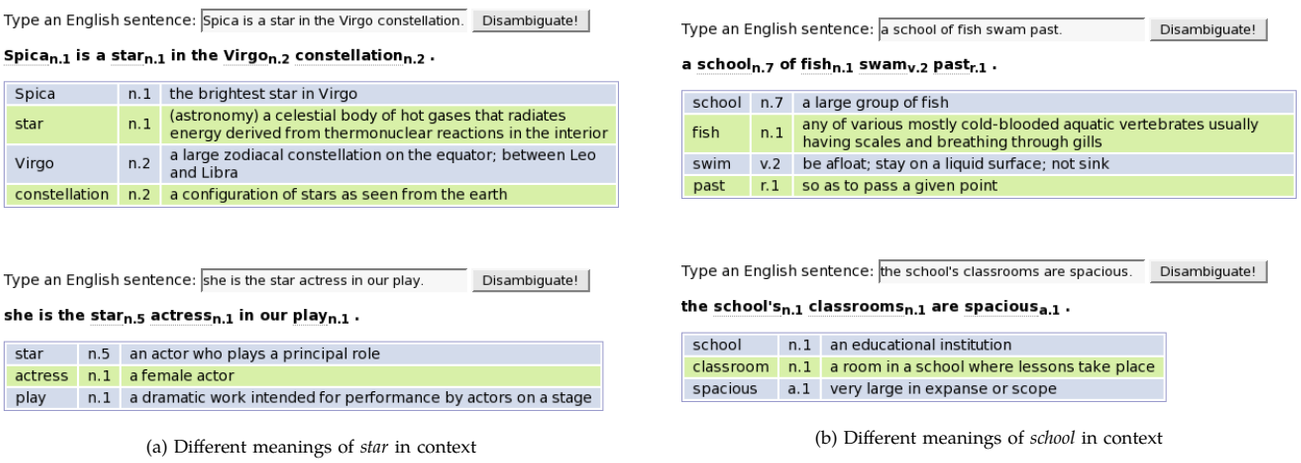


Figure 3: *Blexisma2* agents are deployed on USM Campus Grid nodes using the Globus job submission toolkit.



(a) Different meanings of *star* in context

(b) Different meanings of *school* in context

Figure 4: SLB data generated by *Blexisma2* agents used to determine most probable meanings of ambiguous lexical item. A Web interface to the Sense-Tagger is available at `http://utmkblex.usmgrid.myren.net.my:8080/Blexisma2Servlets/CVSenseTagger`.

interested to improve its design to a multilingual setting, where more complicated cross-lingual phenomena will have to be considered. We also hope to improve the agent communication mechanisms to reduce latency. Apart from creating more agents that implement different learning algorithms and heuristics, we are also planning agents responsible for other NLP tasks such as deep parsing, detecting named entities, etc. In other words, we hope to have agents of various responsibilities so that they can be 'mixed-and-matched' to construct new NLP applications, such as those mentioned in the introduction.

On the performance issues, we may attempt several solutions for the SLB bottleneck problem mentioned earlier:

- hardware (RAM) upgrades,
- further optimisation of PostgreSQL server settings,
- database connection pooling,
- load balancing,
- parallelising queries.

PROJECT PUBLICATIONS

Lim, L. T. & Schwab, D. (2008). Limits of Lexical Semantic Relatedness with Ontology-based Conceptual Vectors. In *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science (NLPCS'08)*. Barcelona, Spain; pp. 153–158.

Schwab, D. & Lim, L. T. (2008). Blexisma2: a Distributed Agent Framework for Constructing a Semantic Lexical Database based on Conceptual Vectors. In *Proceedings of the International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008)*. Penang, Malaysia; pp. 102–110.

Image: A business handshake. Photograph by ©FOTOCROMO (http://www.sxc.hu/profile/FOTOCROMO).

B2B STANDARDS COMPONENT MODELING

Information Systems Engineering Research Group

School of Computer Sciences

TEAM MEMBERS

Project Leader : Dr Vincent Khoo Kay Teong
Researchers : Ting Tin Tin, Rinki Yadav, Johnson Foong, Kor Chan Hock

INTRODUCTION

This report is about the research, design, development and implementation of two B2B communications models on a USM networked non-Grid architecture, and also an attempted implementation of the models on a MyREN networked Grid architecture.

B2B INTEGRATION STANDARDS

A major challenge in integrating the business processes among the trading partners is the enhancement of the documents interchange processes. Traditional business-to-business (B2B) process integration is based on a 'Push' model through which documents are pushed from the senders to the receivers, as in e-mailing and electronic data interchange. However, as the volume of document increases, the 'Push' model suffers from diminishing data quality, which could be due to the low personalizability in the communication subject. Pushing a large volume of pre-defined standard documents among the traders usually results in data redundancy. In addition, low personalizability in communication channel also incurs a higher cost and longer time in developing and deploying the infrastructure. In this research, we explored the 'Pull-only' model and the 'Push and Pull' model to increase the personalizability and thereby promote a more pervasive use of the B2B integration standards.

'PULL-ONLY' MODEL

The 'Pull-only' model is a style of communication in which the receiver initiates the data request to be responded by the sender accordingly. The process that involves the sender is automated by the Web services. After the receiver has initiated the information request, the stored authentication information will be sent by the Web services to the sender's server for authentication, and connecting the Web services to the sender's database view. Once it has been authenticated, the sender's database view will allow the receiver to 'pull' data through the Web services. The raw data will be transformed into a XML format message based on the predefined standard schema. Once the XML message is validated, it can be transformed back into raw data and inserted into the receiver's database automatically.

The purpose of the transformation of raw data into XML message is to ensure that the message conforms to the standard schema. This will ease the data insertion process.

'PUSH AND PULL' MODEL

A 'Push and Pull' model is a style of communication in which the sender pushes the data to the receiver but the amount of data received is filtered by a standard schema pre-defined by the receiver (pull). The receiver controls what to receive through the filter which does not require the receiver to manually select what to be inserted

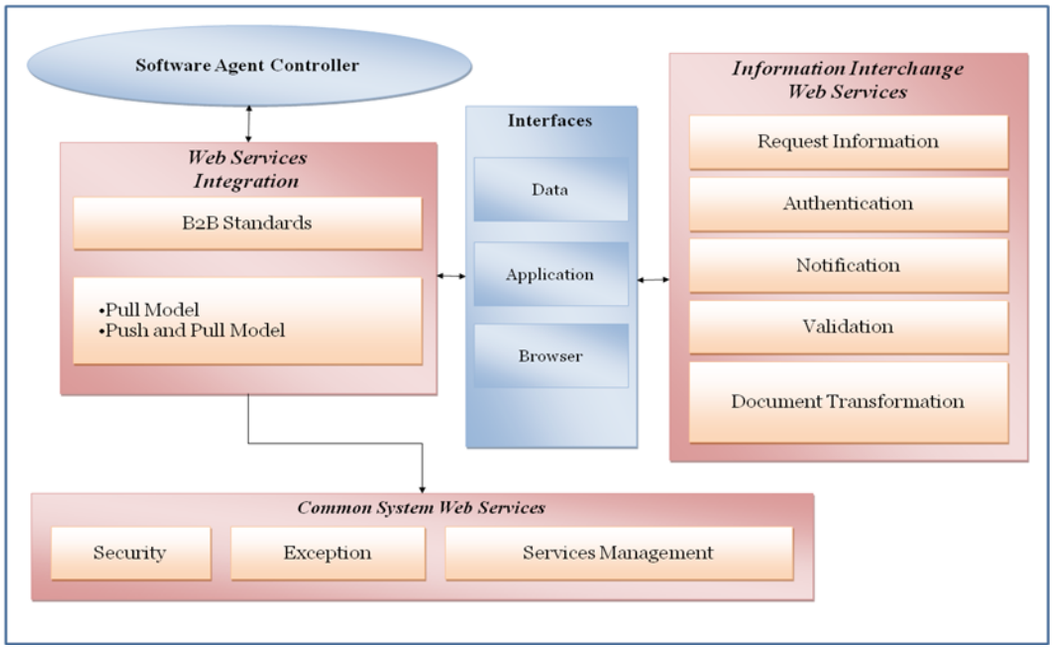


Figure 1: B2B Web Services Integration Framework

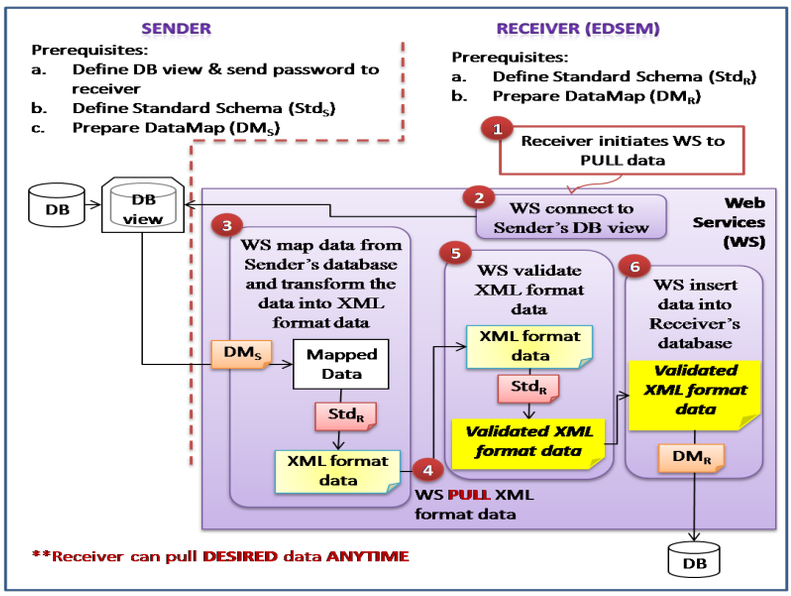


Figure 2: 'Pull' Model Mechanism

into their database during the data insertion process.

ENTERPRISE GRID ARCHITECTURE FOR COLLABORATIVE B2B INFORMATION INTERCHANGE

The Grid infrastructure layer supports a collaborative database sharing middleware. All the virtual databases sharable among the partners are maintained at a centralized physical database. The grid middleware used in this implementation controls and manages the retrieval of the databases. The 'Push' and 'Pull' modeling in Education Service Mashup (EdSeM) helps the partners to interchange confidential information more effectively. EdSeM is the partner of choice for business partners that are interested in building long-term relationship among the educational institutions worldwide. EdSeM has been set up to deploy, leverage and integrate many different types of emerging technologies including the Web 2.0 content extraction, agent-based, and decision-support technologies; knowledge dissemination and knowledge representation technologies. The EdSeM infrastructure complements the middleware by sharing the information interchange Web services across the network. Every partner can use the services to interchange confidential information.

RESULTS & APPLICATION

InfoX is a data mapping tool that maps raw data among the files in a database. It uses the 'Push and Pull' models for transforming data between the trading partners. Each time a trading partner needs to send or receive

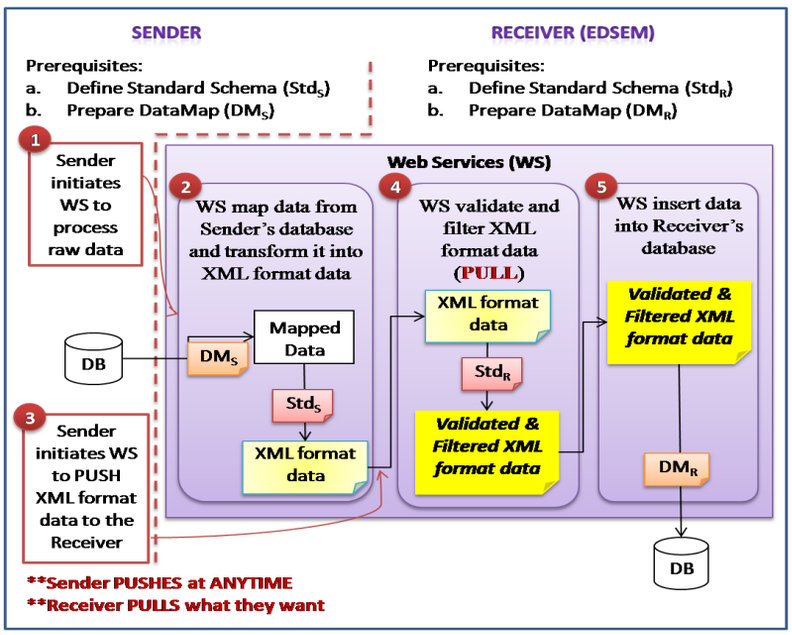


Figure 3: 'Push and Pull' Model Mechanism

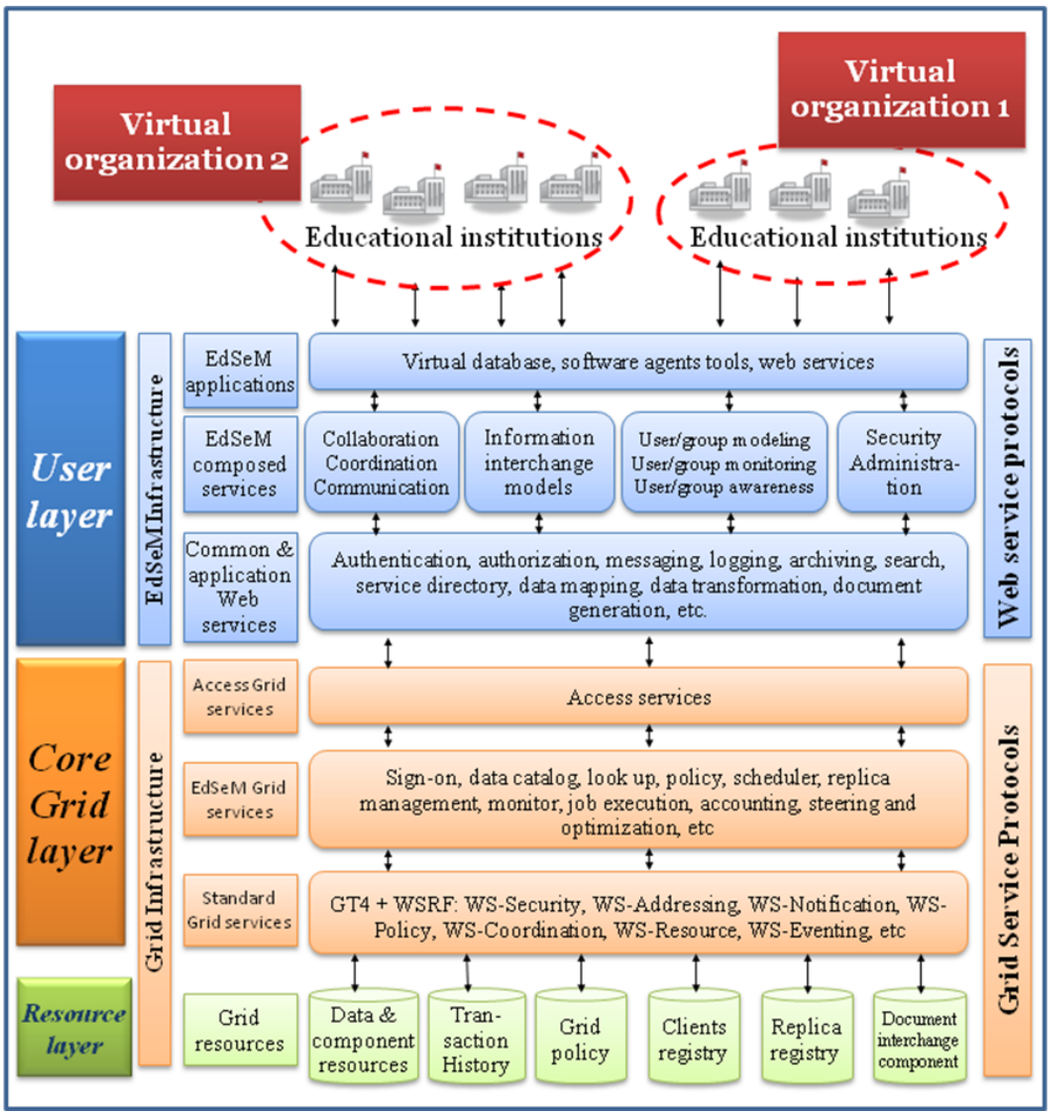


Figure 4: Enterprise Grid Architecture

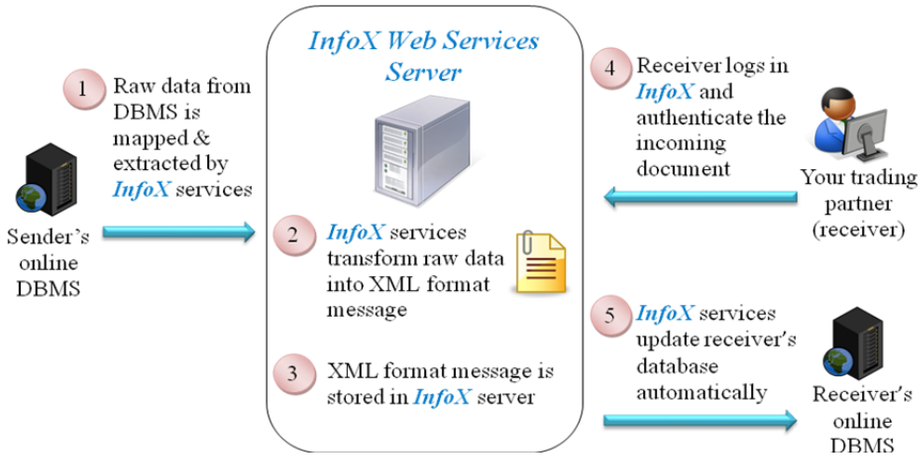


Figure 5: *InfoX* Architecture

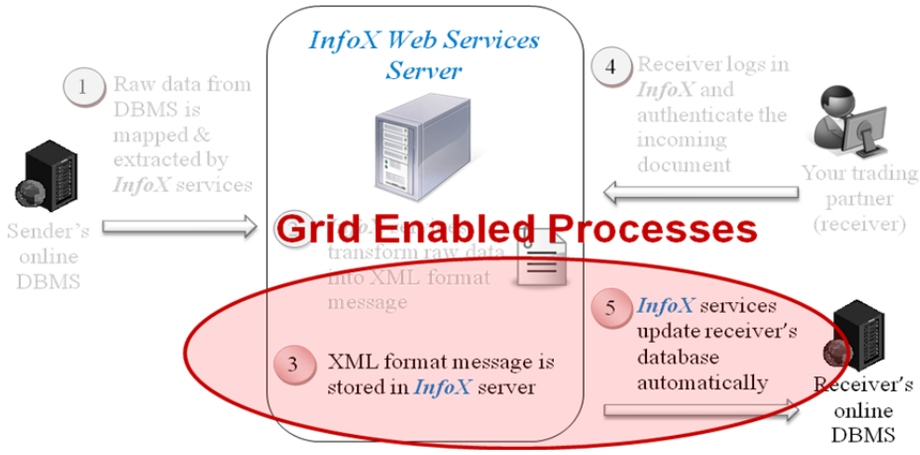


Figure 6: *InfoX* Infrastructure Incorporating Grid Architecture

data, it has to be connected to the *InfoX Web Services Server* to request for the Web service to perform the required action. All the raw data will be transformed into the XML format to ease the data transformation process. The XML format message will be stored in the *InfoX* server which is sharable throughout the network through a grid middleware. The grid middleware controls and manages the retrieval of the database. The management is transparent to the users. The end users only need to query the desired information through the middleware. The grid services will subsequently retrieve the data from various databases based on a data catalog. The distribution of data at various databases is carried out by the grid services.

By deploying the *InfoX* system in a grid environment, it is able to send 45 records per second on an average. However, it was observed that some 340 records per second on an average could be achieved in a non-grid environment.

In order to deploy Infox in a grid environment through a grid middleware, both the *InfoX* and the middleware must be located in a grid MyREN network. Due to the incompatibility of *InfoX* in the MyREN network, it takes a much longer time to process the data and eventually affects the response time.

We have also implemented a self-service evaluation system for the potential educational institutions to estimate the possible time required for the development and deployment of an industry-wide standardised architecture.

WHAT'S NEXT?

As part of the future work, *InfoX* and the grid middleware have to be enhanced from various perspectives. First of all, the *InfoX* system must have the ability to compress the data before sending to the grid middleware for further processing. It is hoped that the response time especially when sending large files will eventually be enhanced. From the grid system perspective, we need to further investigate the existence of a botteneck in the MyREN network.

PROJECT PUBLICATIONS

Ting, T. T. & Khoo, V. K. T. (2009). B2B Standardized Information Interchange Challenges – A Study on Standardization versus Personalization. In *Proceedings of the 2009 International Conference on Advanced Management Science (ICAMS 2009)*; pp. 244–248.

Ting, T. T. & Khoo, V. K. T. (2008). Personalizable Information Interchange Architecture for Educational Institutions. In *Proceedings of the 3rd International Conference on e-Commerce with Focus on Developing Countries (ECDC'08)*. Isfahan, Iran.

Image: A programmer hard at work. Photograph by ©Milen Yakimov (http://www.sxc.hu/profile/todorov40).

AN AUTOMATED JAVA TESTING TOOL ON THE GRID

Software Engineering Research Group

School of Electrical & Electronic Engineering

TEAM MEMBERS

Project Leader : Dr Kamal Zuhairi Zamli
Researchers : Dr Nor Ashidi Mat Isa, Mohammed Issam Younis, Saidatul Khatimah binti Said

INTRODUCTION

Nowadays, we are increasingly dependent on software to assist as well as facilitate our daily chores. In fact, whenever possible, most hardware implementation is now being replaced by the software counterpart. From the washing machine controllers, mobile phone applications to the sophisticated airplane control systems, the growing dependent on software can be attributed to a number of factors. Unlike hardware, software does not wear out. Thus, the use of software can also help to control maintenance cost. Additionally, software is also malleable and can be easily changed as the need arises.

Covering as much as 40 to 50 % of the total software development costs, testing can be considered one of the most important activities for software validation and verification. Lack of testing can lead to disastrous consequences including loss of

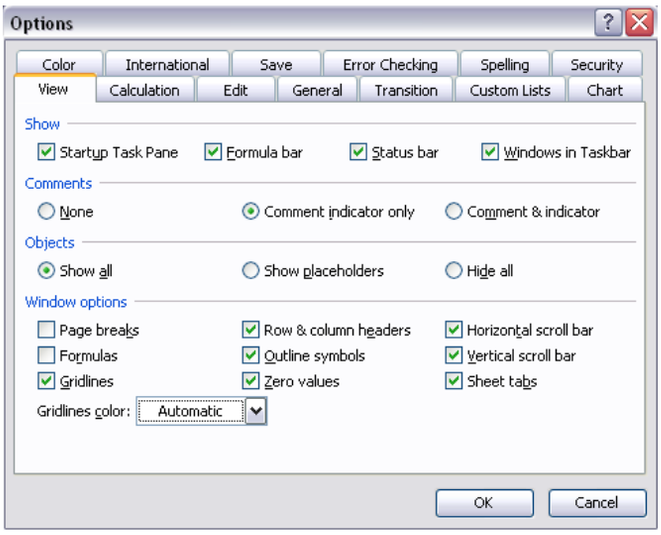


Figure 1: Option dialog in Microsoft Excel

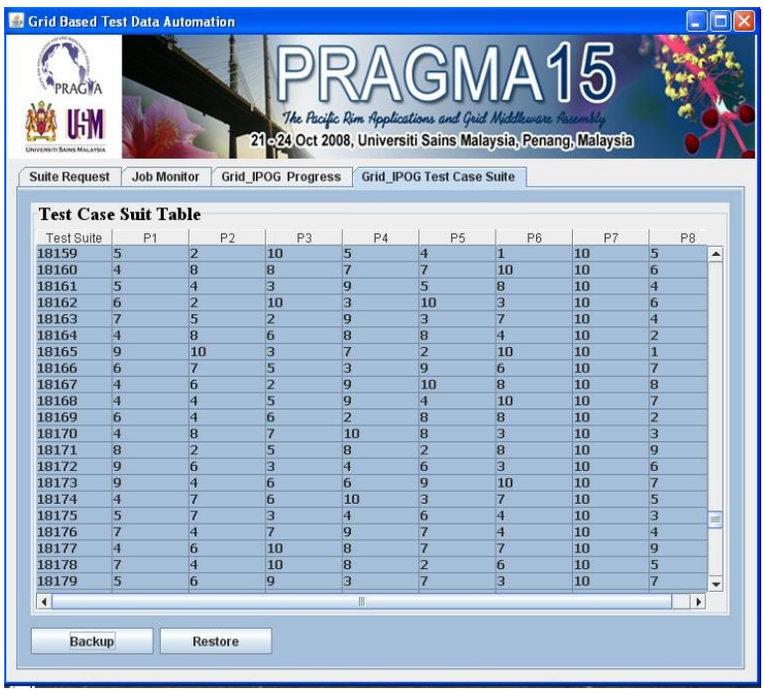


Figure 2: G_MIPOG Graphical User Interface

data, fortunes and even lives. Many combinations of possible input parameters, hardware/software environments, and system conditions need to be tested and verified against for conformance based on the system's specification. Often, this results into combinatorial explosion problem.

As illustration, consider the option dialog in Microsoft Excel software (see Figure 1). Even if only the View tab option is considered, there are already 20 possible configurations to be tested. With the exception of Gridlines color which takes 56 possible values, each configuration can take two values (i.e. checked or unchecked). Here, there are $2^{20} \times 56$ (i.e. 58,720,256) combinations of test cases to be evaluated. Assuming that it takes 5 minute for one test case, then it would require nearly 559 years for a complete test of the View tab option. As highlighted above, given limited time and resources, exhaustive testing is next to impossible. Obviously, there is a need for a systematic strategy in order to reduce the test data into manageable ones.

Earlier work has indicated that pairwise testing (i.e. based on 2-way interaction of variables) can be effective to sample the test data appropriately. While such conclusion may be true for some systems, it cannot be generalized to all software system faults (i.e. especially when there are significant interactions between variables). This argument is supported by the fact that software applications grew tremendously in size from kilobytes to terabytes in the last 5 years. Driven by the advancement of technologies and market demands, the net effect of this massive software growth can often lead to intertwined dependency between parameters involved, thus, justifying the need to consider for high interaction strength (i.e. often indicated as t).

Although desirable, the consider-

Table 1: Results For 7 to 10 5-Valued Parameters in 6-Way Testing

# Parameters	7	8	9	10
Test Case Size	15,625	28,125	40,146	45,168
Time (MIPOG) (Single Computer)	118.703	485.047	1637.097	4657.457
Time (G_MIPOG) (6-Computers)	55.234	183.184	626.797	1552.125
Speedup	2.149	2.648	2.838	3.203

Table 2: Results using 1 to 11 Computers for 10 10-valued Parameters in 4-way Testing

# Computers	Time	Speedup	Test Case Size
1 (MIPOG)	77,882.16	1	
2 (G_MIPOG)	43,680.404	1.783	
3 (G_MIPOG)	31,140.4	2.501	
4 (G_MIPOG)	24,089.749	3.233	
5 (G_MIPOG)	17,150.883	4.541	
6 (G_MIPOG)	15,193.554	5.126	27,306
7 (G_MIPOG)	14,928.537	5.217	
8 (G_MIPOG)	14,789.624	5.266	
9 (G_MIPOG)	12,928.645	6.024	
10 (G_MIPOG)	9570.184	8.138	
11 (G_MIPOG)	9352.967	8.327	

ation for higher interaction strength (e.g. from $t = 3$ onwards) can be problematic. When the parameter interaction coverage t increases (i.e. as $t > 2$), the number of t-way test set also increases exponentially. For example, consider a system with 10 parameters, where each parameter has 5 values. There are 1125 2-way tuples (or pairs), 15,000 3-way tuples, 131,250 4-way tuples, 787,500 5-way tuples, 3,281,250 6-way tuples, 9,375,000 7-way tuples, 17,578,125 8-way tuples, 19,531,250 9-way tuples, and 9,765,625 10-way tuples. From this example, it is evident that for large system with many parameters, considering higher order t-way test data invites massively notorious computation.

Facing these challenges, our research investigates a new strategy and its implementation, called G_MIPOG. Here, G_MIPOG (see Figure 2) originates from a well known sequential t-way strategy called IPOG. Unlike its predecessor strategy (IPOG), G_MIPOG permits harnessing of grid based architecture, allowing the support for higher t than that of IPOG.

OUR EXPERIENCE WITH THE G_MIPOG STRATEGY

As the first step to the development of G_MIPOG, we investigate an intermediate strategy, called MIPOG[2], based on the modification of IPOG. Our experimental result demonstrates that the control and data dependencies in MIPOG can be removed to permit distributed as well as parallel processing.

As part of our research, we intend to investigate whether there is a speedup gain from distributing MIPOG in G_MIPOG. Three experiments are applied to both MIPOG and G_MIPOG in order to gauge the speedup. Here, the speedup is defined as ratio of the time taken by single computer to the time taken by multiple computers. The experimental goals are to investigate:

- the speedup as the number of parameters increases;
- the speedup as the number of computer nodes increases;
- the speedup against the increase of parameter coverage;
- the comparison in terms of test size against other strategies.

To achieve the first goal, we apply MIPOG, and G_MIPOG (consists of 6 computers) to 5-valued parameters, and vary the number of parameters from 7 to 10, and fix $t = 6$. The results are demonstrated in Table 1. To achieve the second goal, we have fixed $t = 4$, with 10 10-valued parameters. Then, we determine the speedup using 2 to 11 computers. The results are shown in Table 2. To achieve the third goal, we apply G_MIPOG to the Traffic Collision Avoidance System (TCAS) module. Here, the TCAS module is an aircraft collision avoidance system from the Federal Aviation Administration which has been used as case study in other related works. The TCAS module has twelve parameters: seven parameters have 2 values, two parameters have 3 values, one parameter has 4 values, and two parameters have 10 values. In this case, we use 2 computers, and vary t from 2 to 11 to determine the speedup, as given in Table 3. Finally, to achieve the last goal, we compare the test size generated by G_MIPOG and other strategies for the aforementioned TCAS module. The results are tabulated in Table 4. We have darkened the entries to show the best result. Entry with 'NS' refers to as not supported by the implementation.

DISCUSSION

In Table 1, we note that the speedup increases linearly as the number of parameters increases. Here, extra overhead is added for the fifth parameters due to the need to start and shut down the corresponding threads. Referring to Table 2, we note the speedup gain also increases quadratically as the number of values increases. Extrapolating and performing curve fitting of the results from Ta-

[2]The details of MIPOG can be found at http://gclportal.usmgrid.myren.net.my/trac/engpjtst

Table 3: Results Using $t = 2$ to 11 for TCAS Module

t-Way	Time (MIPOG)	Time(G_MIPOG)	Speedup	Test Case Size
2	0.156	0.292	0.534	100
3	0.609	0.547	1.113	400
4	3.234	2.578	1.2544	1265
5	36.797	29.125	1.263	4196
6	301.128	214.091	1.406	10,851
7	1772.407	1086.7	1.631	26,061
8	10,242.09	5839.276	1.754	56,742
9	36,284.7	19,124.78	1.897	120,361
10	41,481.672	21,832.46	1.9	201,601
11	14,939.891	7821.932	1.91	230,400

Table 4: Comparative Test Size Results Using TCAS Module for $t = 2$ to 12

t	G_MIPOG	IPOG	IPOG_D	IPOF1	IPOF2	ITCH	Jenny	TConfig	TVGII
2	100	100	130	100	100	120	106	108	101
3	400	400	480	402	427	2388	411	472	434
4	1265	1361	2522	1352	1644	1484	1527	1476	1599
5	4196	4219	5306	4290	5018	NS	4680	NS	4773
6	10,851	10,919	14,480	11,234	13,310	NS	11,608	NS	NS
7	26,061	NS	NS	NS	NS	NS	27,630	NS	NS
8	56,742	NS	NS	NS	NS	NS	58,865	NS	NS
9	120,361	NS	NS	NS	NS	NS	NS	NS	NS
10	201,601	NS	NS	NS	NS	NS	NS	NS	NS
11	230,400	NS	NS	NS	NS	NS	NS	NS	NS
12	460,800	NS	NS	NS	NS	NS	NS	NS	NS

ble 3, we observe that the speedup increases logarithmically as the strength of coverage increases. In this case, there is also no speedup gain for this strategy when $t = 2$, possibly due to the overhead required for creation, synchronization, and deletion of threads for small degree of interaction. Concerning comparison with other strategies in Table 4, G_MIPOG, IPOG, IPOF1, and IPOF2 gave the most optimum test size at $t = 2$. At $t = 3$, both G_MIPOG and IPOG gave the most optimum test size. For all other cases, G_MIPOG always outperforms other strategies. Putting G_MIPOG aside, only Jenny can go more than $t > 6$ for the TCAS module. TVG implementation do not give any result when $t > 5$ whilst TConfig does not produce any result when $t > 4$. ITCH can not support $t > 4$. We also note that in the case of ITCH, the test size for $t = 3$ is greater than that of $t = 4$.

CONCLUSION & FUTURE WORK

In short, our experimental results demonstrate that G_MIPOG scales well against its sequential strategy (MIPOG) with the increase of the computers as computational nodes. Moreover, G_MIPOG produces minimal test set as compared with other existing strategies (whilst keeping the test set of MIPOG the same). Finally, G_MIPOG is the only strategy that supports high t even against all other IPOG variants (i.e. IPOG_D, IPOF1, and IPOF2).

As t-way testing approach is still in infancy within the software engineering community, novel research and findings are highly welcomed. The work discussed here is indeed significant to facilitate the testing activities by systematically minimizing the test data for test consideration. As human are more and more dependent toward software, this work presents a small leap toward the betterment of the way engineers do testing.

As a part of our future work, we are trying to integrate test execution and reporting along with parallelizing the data structure for storing the interaction element using tuples-space technology.

PROJECT PUBLICATIONS

Younis, M. I., Zamli, K. Z., Klaib, M. F. J., Che Soh, Z. H., Che Abdullah, S. A., & Mat Isa, N. A. (2009). Assessing IRPS as an Efficient Pairwise Test Data Generation Strategy. *International Journal of Advanced Intelligence Paradigms*. (forthcoming).

Younis, M. I., Zamli, K. Z., & Mat Isa, N. (2008). IRPS – An Efficient Test Data Generation Strategy for Pairwise Testing. *Lecture Notes in Artificial Intelligence, 5177*.

Younis, M. I., Zamli, K. Z., & Mat Isa, N. A. (2008). A Strategy for Grid Based t-Way Test Data Generation. In *Proceedings of the International Conference on Distributed Frameworks & Applications (DFmA 2008)*. Penang, Malaysia; pp. 73–78.

Zamli, K. Z., Che Abdullah, S. A., Younis, M. I., & Che Soh, Z. H. (2009). *Software Testing Module*. (forthcoming). OpenUniversity Malaysia and Pearson Publishing.

Image: Ethernet crossover cable. Photograph by ©Elia Schmidt (http://The-Condor.deviantart.com/)

INET-GRID

Network Research Group

School of Computer Sciences

TEAM MEMBERS

Project Leader : Prof. Sureswaran Ramadass

Researchers : AP Rahmat Budiarto, AP Chan Huah Yong, Dr Ahmed M. Manasrah

GRID COMPUTING

A Grid environment is a complex globally distributed system that involves large sets of diverse, geographically distributed components used for a numerous type of applications. The components discussed here include all of the software and hardware services and resources needed by applications. Due to the diversity of the components and enormous number of users, the Grid environment has become more vulnerable to fault, failure and excessive tools. Thus, a suitable mechanism is needed to monitor the components, their usage and detecting conditions that may lead to bottlenecks, faults or failures. Grid monitoring is a potential platform towards a robust, reliable and efficient environment in networking whereby it is crucial to maintain network reliability during a peak environment. In conclusion, a combination between network monitoring and grid monitoring are required in order to enhance network reliability and performances.

Network monitoring in Grid environment plays an important role in network management and is essential in Grid utilization. According to the Global Grid Forum schema, the management of network infrastructure observations is divided into three main activities: their production, using network monitoring tools, their publication, by way of powerful databases, and their utilization by administration and workflow analysis tools.

We are embarking on an intelligent real time grid monitoring system called *iNet-Grid*. The purpose of this research is to enable an intelligent tool to assist network and system administrators intelligently by providing information for preventive measures to be taken to minimize damages due to system or network down time that can be very costly. Real-time network analysis helps in detecting and resolving network faults and performance problems in a minimum amount of time. This system has the power to analyze multi-topology, multi-protocol networks, automatically.

TECHNICAL OBJECTIVE

To prove the ability of the proposed work to be integrated with any open source application for grid monitoring (*Ganglia*) and thus, building a platform that allow all other grid monitoring system to be integrated together for better monitoring results. For example, the platform allows a network monitoring solution and a grid monitoring solution to work hand in hand to provide network monitoring information and therefore, accurate grid monitoring results.

GENERAL BENEFITS

Figure 1 depicts the general benefits from monitoring grid environments.

1. To understand the performance, identify problems and to tune a grid system for better overall performance,
2. Fault detection and recovery mechanisms; to determine if there are any units within the grid environment that are not functioning properly, and thus, to decide whether to restart a component or redirect a service requests elsewhere,
3. For debugging purposes,
4. Resource utilization,
5. Management decisions,
6. Performance utilization,
7. Accounting,
8. Security.

INET-GRID

iNet-Grid was built on top of Eclipse Platform as a Rich Client Platform

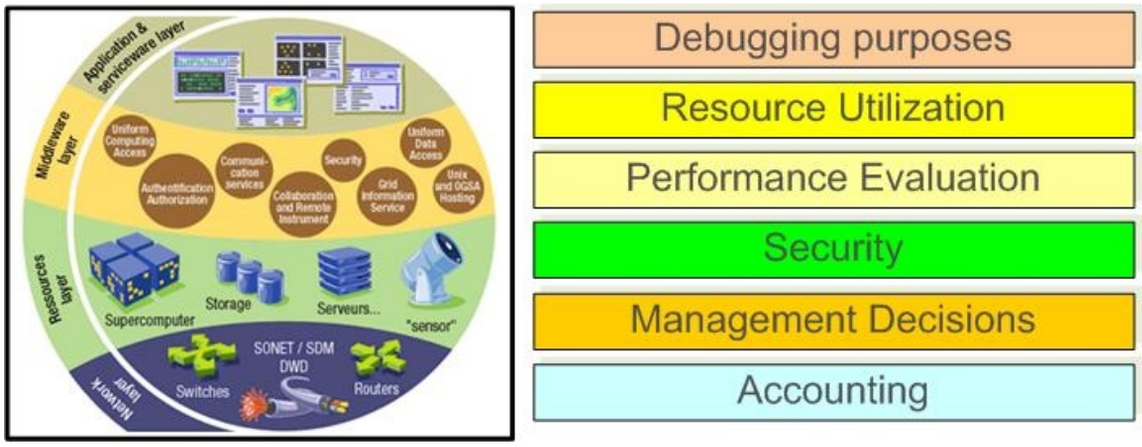


Figure 1: Grid monitoring benefits

(RCP) application that gives the flexibility in deploying the end solution into various/any grid environments regardless of the grid architecture or the operating system. The RCP provides the power of plugins that can extend the system functionality easily and quickly without refactoring the whole core component from the scratch. Moreover, these plugins also provide extension points for other plugins to be integrated or add/enhance any existing service (plugin) such as interfacing to *Ganglia* and any other monitoring tools.

ARCHITECTURE

iNet-Grid is one of the core modules of *iNet-Enterprise Suite* and plays a role as the centralized grid monitoring module that retrieves and manages all the nodes information within a cluster send by a grid monitoring agent. Only one grid agent needs to be installed in a Linux-based machine within a cluster equipped with any grid monitoring solution such as *Ganglia*. This monitoring agent will look into *Ganglia* database, retrieve the data and send it to the central monitoring Server a specific defined time interval (15 seconds by default). Figure 2 best visualizes the *iNet-Grid*'s architecture followed by its monitoring agent architecture in Figure 3.

Furthermore, *iNet-Grid* consists of three components: grid agent, grid data manager and grid data viewer. A grid agent is installed in one machine for each cluster. It collects and processes data loaded from the *Ganglia* database and sends it to grid data manager which resides in *iNet-Server*. Figure 4 portrays *iNet-Grid*'s functional architecture.

The monitoring information consists of the number of all incoming/outgoing packets, CPU speed, CPU number, CPU usage load and hard disk used/free size. On the other hand, the network monitoring information consists of; bandwidth utilization, viruses/worms infecting or propagating within the network, the applications that are consuming bandwidth, traffic analyzers, etc. The values are read from Ganglia database which kept in round robin database (RRD) format files in the location `/usr/share/ganglia/rrdtool/file/libs`. Below is the list of the files:

- boottime.rrd
- cpu_idle.rrd
- cpu_system.rrd
- disk_total.rrd
- mem_buffers.rrd
- mem_total.rrd
- proc_run.rrd
- bytes_in.rrd
- cpu_nice.rrd
- cpu_user.rrd
- load_fifteen.rrd
- mem_cached.rrd
- part_max_used.rrd
- proc_total.rrd
- bytes_out.rrd
- cpu_num.rrd
- cpu_wio.rrd
- load_five.rrd
- mem_free.rrd
- pkts_in.rrd
- swap_free.rrd
- cpu_aidle.rrd
- cpu_speed.rrd
- disk_free.rrd
- load_one.rrd
- mem_shared.rrd
- pkts_out.rrd
- swap_total.rrd

CONNECTION FUNCTIONALITY

The grid monitoring agent sends a request to the central monitoring server. The central monitoring server then checks/verifies whether the request is a valid request or not (*the monitoring agent will verify his identity using a unique ID that is pre-defined in the database*), if it is valid, the server will give the authorization to the monitoring agent and allows him to send the data so it can be used for real-time or historical monitoring information. When the data reaches to the server, it will be stored into the database. "Grid" table is where the data is stored (Table 1 shows the table data dictionary).

iNet-Console (which acts as a viewer) will send a grid data request to the central monitoring server requesting for grid monitoring information. Finally, the received grid information will be displayed to the viewer. The format of the data sent to the client is as shown in Figure 5.

RESULTS

The implementation of *iNet-Grid* prototype has been done. We tested it with 5 clusters and 90 nodes in total. From our evaluation we can conclude the effectiveness of the system. The screenshots taken during the prototype are shown in Figure 6.

EFFICIENT GRID MONITORING

Grid monitoring is considered efficient if it is a combination of network and grid monitoring, the network monitoring in term of bandwidth, anomaly, security, user misusage/misdealing with the network resources which obviously will result in slow network, thus, minimizing the grid productivity specially if it requires communication between different parties over the network. On the other hand, grid monitoring will ensure that the Grid resources are used and managed wisely and efficiently

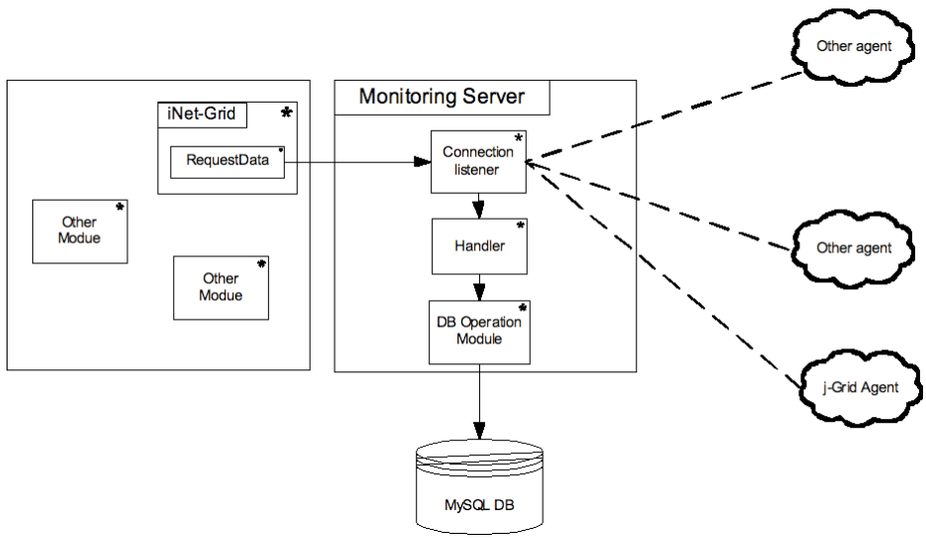


Figure 2: Grid monitoring general architecture

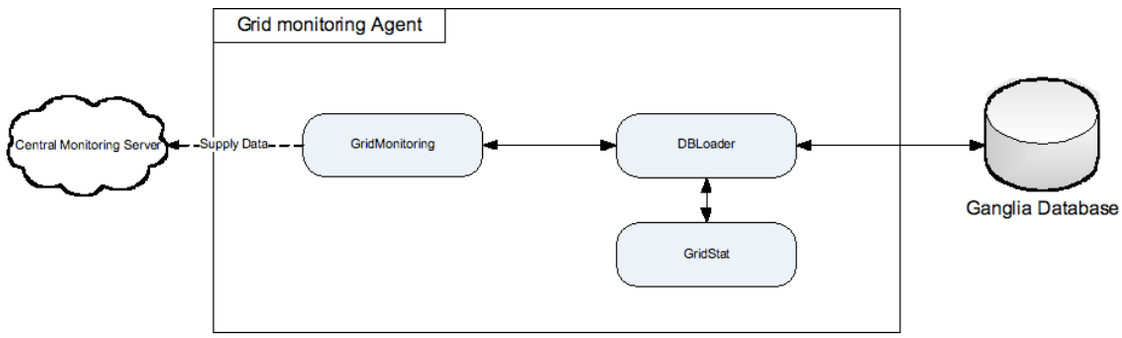


Figure 3: Grid agent architecture

Table 1: Data dictionary of the "Grid" table

Column Name	Comment	Column Name	Comment
Node_name	Holds the CPU ID value	Time	Holds the time value
Boot_time	Holds the boot time value	Cpu_idle	Holds the CPU idle value
Cpu_system	Holds the CPU system value	Disk_total	Holds the disk total value
Mem_buffers	Holds the memory buffers value	Mem_total	Holds the memory total value
Proc_run	Holds the number of running processes	Bytes_in	Holds the bytes in value
Cpu_nice	Holds the CPU nice value	Cpu_user	Holds the CPU user value
Load_fifteen	Holds the load fifteen value	Mem_cached	Holds the cached memory value
Part_max_used	Holds the part max used value	Proc_total	Holds the total processes value
Bytes_out	Holds the bytes out value	Cpu_num	Holds the number of CPUs value
Cpu_wio	Holds the CPU I/O value	Load_five	Holds the load five value
Mem_free	Holds the free memory value	Pkts_in	Holds the packets in value
Swap_free	Holds the swap free value	Cpu_aidle	Holds the CPU idle value
Cpu_speed	Holds the CPU speed value	Disk_free	Holds the free disk value
Load_one	Holds the load one value	Mem_shared	Holds the shared memory value
Pkts_out	Holds the packets out value	Swap_total	Holds the total swap value

by keeping an eye on each individual CPU available in the grid and alerting when certain threshold is triggered. This is illustrated in Figure 7.

The Grid and network monitoring is one tool that all modules are pluggable to each other on top of one core engine that manages the communication between the different available modules controlled by XML manifest. This XML manifest makes it easier to add/remove/enhance any existing functionality. As shown in Figure 8, each module is working/operating hand in hand with others. The core engine is hardwired with a circular buffer using *iNetmon* technology to support high speed network with high amount of traffic efficiently with a minimal packet loss. *iNet-Grid* is the core of the Grid monitoring where it has another pluggable interface for existing grid monitoring tool integration (such as *Ganglia*) without any special

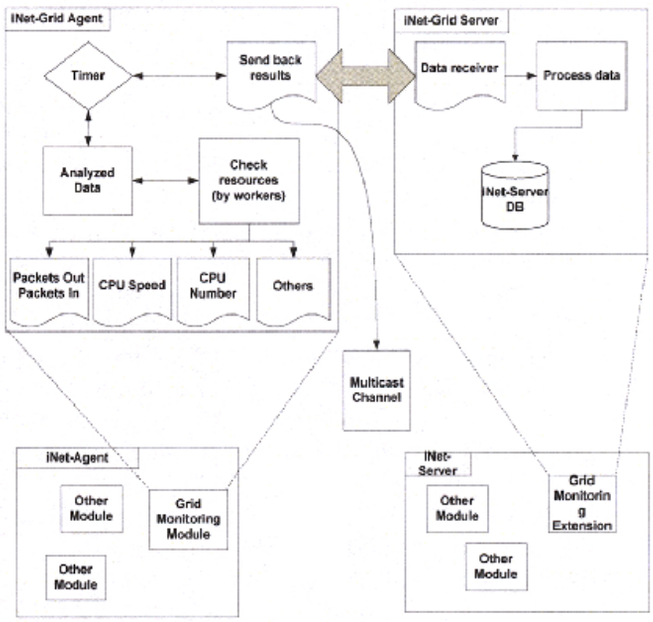


Figure 4: Functional architecture of *iNet-Grid*

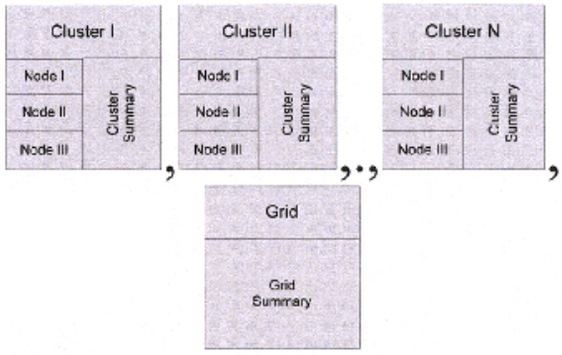


Figure 5: Data format sent to the client

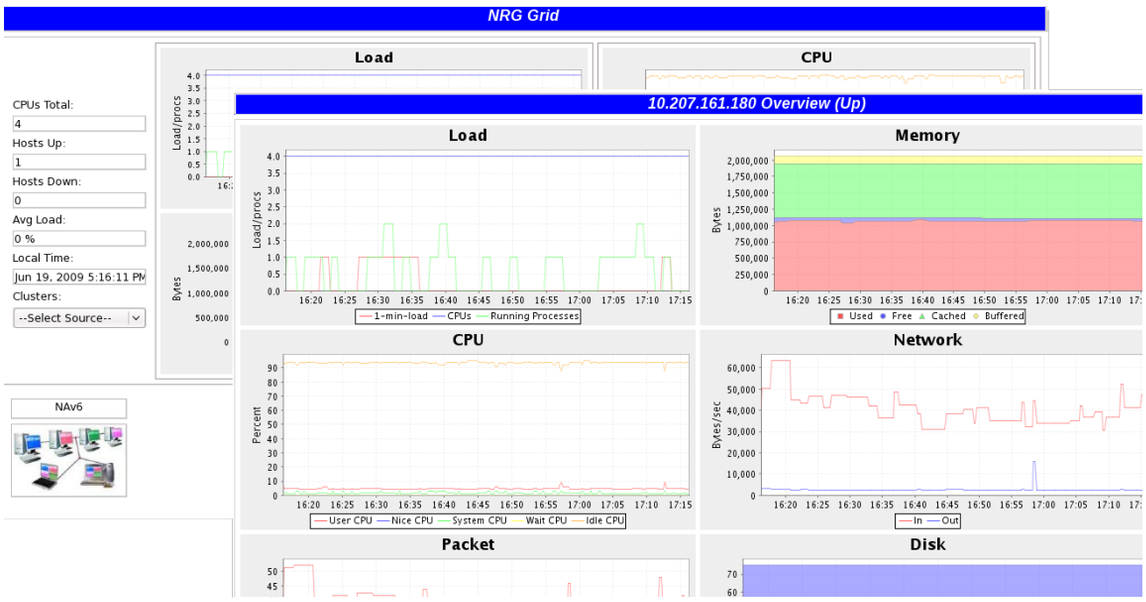


Figure 6: Snapshots of the system

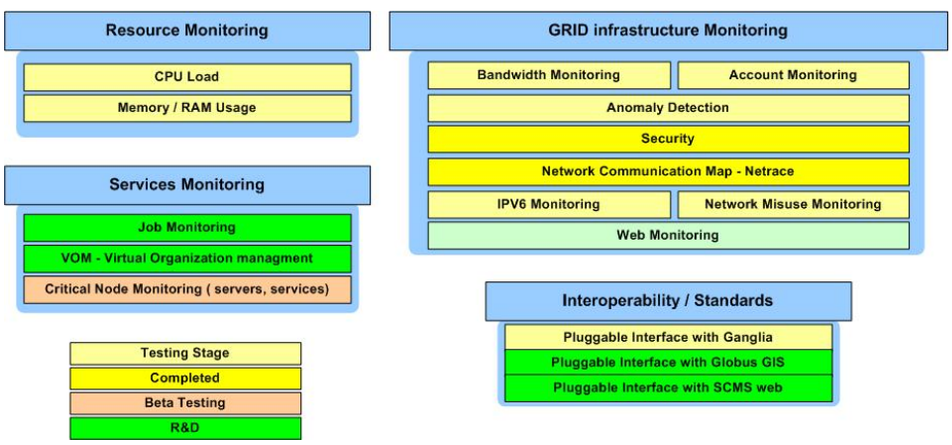


Figure 7: Efficient Grid Environment Monitoring Features

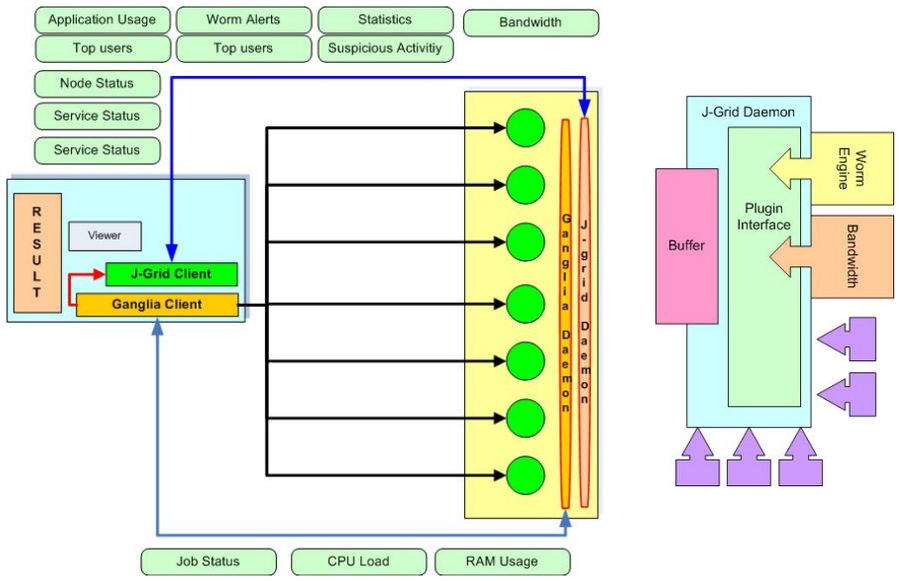


Figure 8: Efficient grid environment monitoring components and architecture

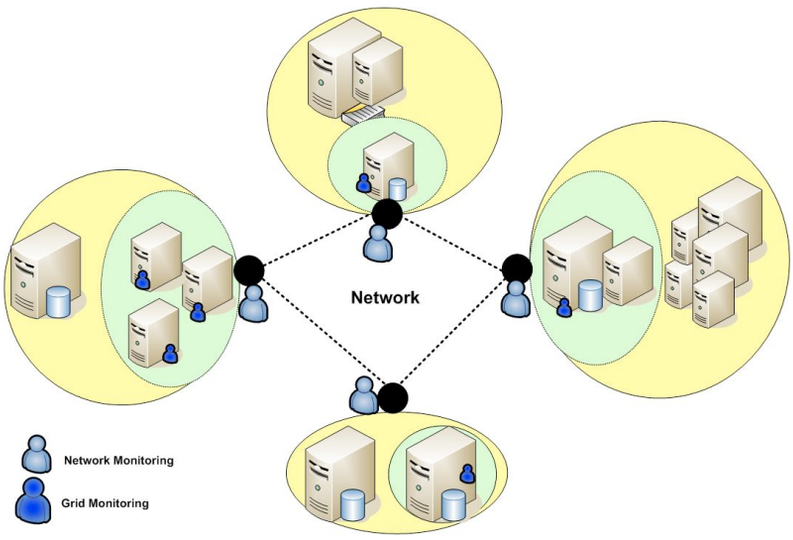


Figure 9: Grid environment monitoring and deployment

configuration accept that for *Ganglia* (client) which is assumed to be ready and functioning as required. *iNet-Grid* client is installed with *Ganglia* client to avoid installing the whole *Ganglia* package. Instead, the entire system will provide as a whole:

1. Application usage statistics,
2. Worm activity alerts,
3. Bandwidth usage,
4. Top users/usage,
5. Nodes status,
6. Disk usage,
7. Packets per second,
8. Cluster statistics,
9. CPU load,
10. RAM usage.

DEPLOYMENT

As we described earlier, the grid-network will have an intelligent agent for the network part that is responsible for any network activity hand in hand with a grid intelligent agent that is in charge of each grid environment resources. Figure 9 portrays the combination between network monitoring and grid monitoring as well as illustrating the deployment plan.

CONCLUSION & FUTURE WORK

Resource monitoring is very important for Grid environment. Many mature systems exist today. However, none of them provides complete solution, which covers all requirements needed for efficient grid monitoring. With this we propose grid monitoring architecture namely *iNet-Grid*. We focus on following issues: integrating existing systems on top of *Ganglia*, providing component for the management of monitoring component and designing *iNet Console* for the GUI. Deployment of the described monitoring systems has already been successfully accomplished and the preliminary results have shown positive outcomes.

PROJECT PUBLICATIONS

Ahmed, M. (2009). A Real Time Distributed Network Monitoring and Security Monitoring Platform (RTDNMS). PhD thesis, School of Computer Sciences, Universiti Sains Malaysia.

Ahmed, M. & Talib, N. A. (2008). iNet-Grid: A Real-Time Grid Monitoring and Troubleshooting System. In *Proceedings of the International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008)*. Penang, Malaysia; pp. 68–72.

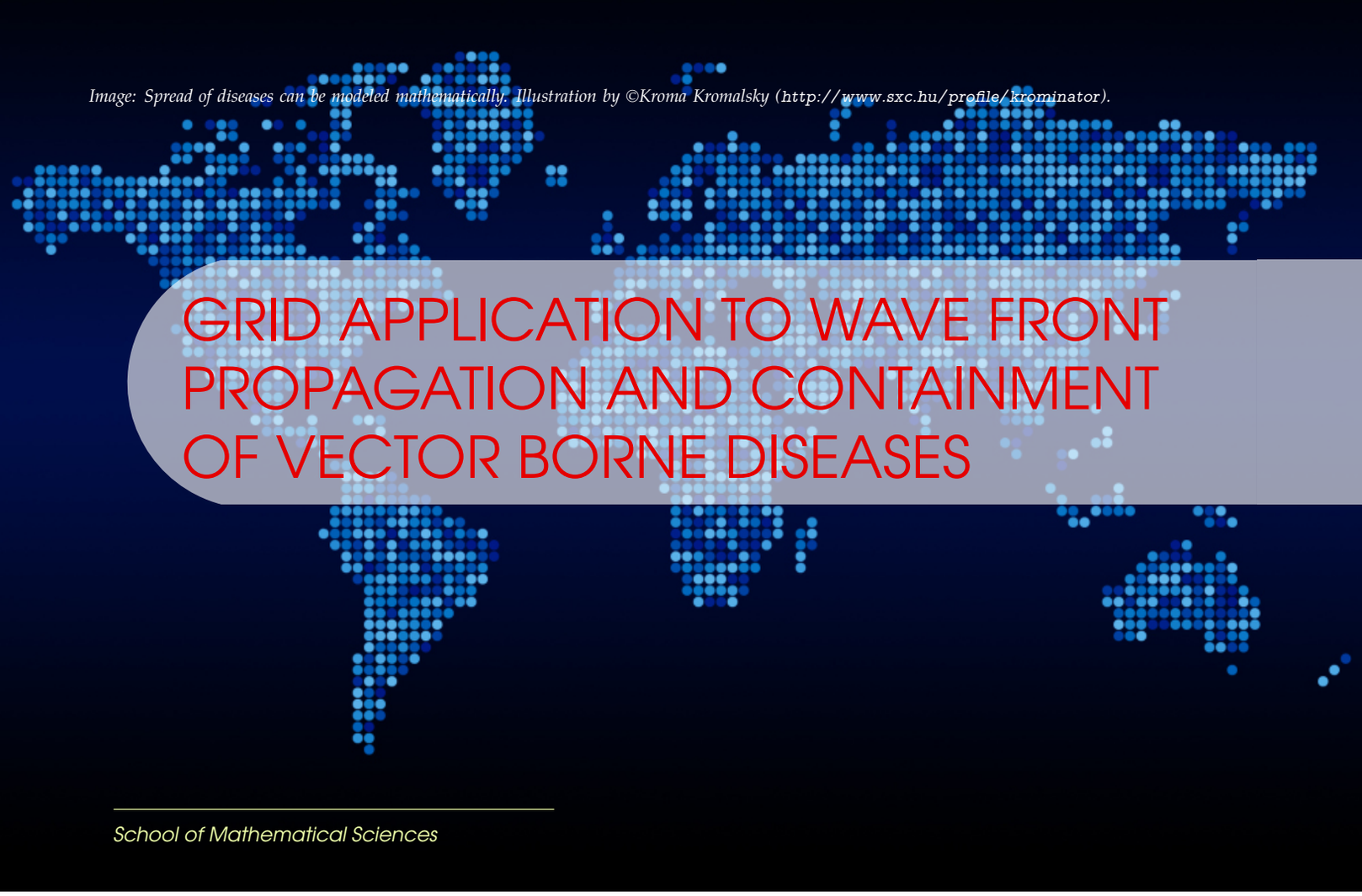


Image: Spread of diseases can be modeled mathematically. Illustration by ©Kroma Kromalsky (http://www.sxc.hu/profile/krominator).

GRID APPLICATION TO WAVE FRONT PROPAGATION AND CONTAINMENT OF VECTOR BORNE DISEASES

School of Mathematical Sciences

TEAM MEMBERS

Project Leader : Prof. Koh Hock Lye
Researchers : Dr Teh Su Yean, Tan Kah Bee

INTRODUCTION

Vector borne diseases such as dengue, Yellow fever, malaria, West Nile Encephalitis (WNE), Japanese Encephalitis (JE), St. Louis Encephalitis (SLE) and Western Equine Encephalitis (WEE) pose serious public health hazard worldwide. Nearly half of the world population is infected with at least one type of vector borne diseases (WHO 2000). Approximately 1.4 million lives are lost each year due to vector-borne diseases. For most of these vector-borne diseases, vaccines are currently not available and are unlikely to be available in the near future. Therefore, the focus is diverted to controlling the spread of a disease by controlling the vector of that disease. For this purpose, a mathematical model is developed by the team to study the mechanisms involved in the invasion and persistence of vector borne diseases. Through model simulations, qualitative and quantitative assessments can be made to determine the factors affecting the dispersal of vector population and the transmission of the disease from the vector to human. Further, control methods can be experimented numerically on its effectiveness before implementation.

CLIMATE CHANGE IMPACT ON DENGUE

Dengue fever is the world fastest growing vector borne disease which is currently endemic or intermittently epidemic in many tropical and subtropical regions. Current estimates suggest that up to 50 million dengue cases occur annually, including 500,000 cases of dengue haemorrhagic fever (WHO 2001). Dengue virus is transmitted by arthropods of the species *Aedes aegypti*, a mosquito found throughout the world where hot and humid climate is predominant. The transmission of the dengue virus has only one epidemiological cycle linking the human host and female *Aedes aegypti* mosquito. Four major serotypes of dengue virus have been identified. A person who recovers from an infection is immune to only that serotype and may become secondarily infected with a different serotype virus (Atkinson et al. 2007). In the absence of vaccine, the efforts to control dengue disease are focused on controlling the mosquito vector. Further, global warming may increase the outbreak risk of dengue disease associated with weather or precipitation pattern modifications (Schaeffer, Mondet, and Touzeau 2008). Climate change may be irreversible, and is predicted to become more extreme in the future (IPCC 2001). To examine the global-scale relationships between climate, *Aedes aegypti* populations and dengue disease, grid technology is appropriate, where a grid will simulate each local habitat.

APPLICATION OF GRID TECHNOLOGY

WNE is a vector borne disease with more than one epidemiological cycle linking the human host, the mosquito vector and another mammal such as birds and pigs. WNE started in New

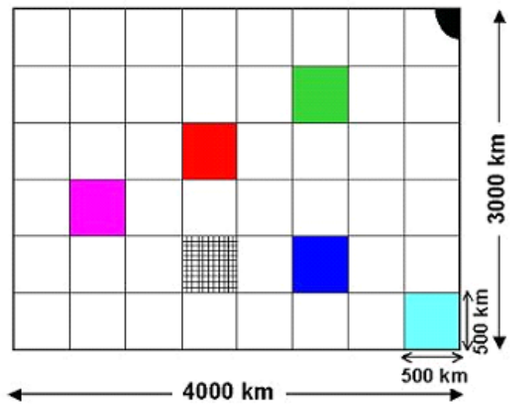


Figure 1: Conceptual computation grid

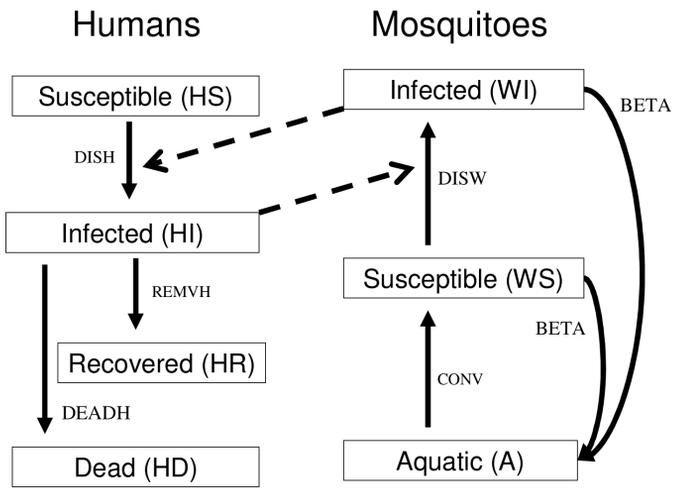


Figure 2: Compartment model for Dengue disease transmission

$$
\begin{aligned}
\frac{\partial WS}{\partial t} &= DIFF \times \frac{\partial^2 WS}{\partial x^2} - v \times \frac{\partial WS}{\partial x} + CONV \times A \times \left(1 - \frac{WN}{CCW}\right) - ALPHAW \times WS - DISW \times BITR \times WS \times \frac{HI}{HN} \\
\frac{\partial WI}{\partial t} &= DIFF \times \frac{\partial^2 WI}{\partial x^2} - v \times \frac{\partial WI}{\partial x} - ALPHAW \times WI + DISW \times BITR \times WS \times \frac{HI}{HN} \\
\frac{\partial A}{\partial t} &= BETA \times WN \times \left(1 - \frac{A}{CCA}\right) - ALPHAA \times A - CONV \times A \\
\frac{\partial HS}{\partial t} &= ALPHAH \times HN - ALPHAH \times HS - DISH \times BITR \times HS \times \frac{WI}{WN} \\
\frac{\partial HI}{\partial t} &= DISH \times BITR \times HS \times \frac{WI}{WN} - REMVH \times HI - ALPHAH \times HI - DEADH \times HI \\
\frac{\partial HR}{\partial t} &= REMVH \times HI - ALPHAH \times HR \\
\frac{\partial HD}{\partial t} &= DEADH \times HI
\end{aligned}
\tag{1}
$$

York in northeast USA in 1990 but quickly spread to the southwest in a matter of 3 to 4 years. The local scale of mosquito dispersion cannot account for this speed with which the disease spread. The spread was indeed enhanced by infected birds that may fly over large distances of tens of kilometers over a season or a few months. This pattern of dispersion may be conceptualized in Figure 1, in which uneven patches of distributions (squares) spreading over a rectangle of 3000 km by 4000 km are used to represent WNE distribution on the scale of continental USA. Over a local patch (a square) of the order of a few km, the PDE for mosquito will be solved by finite difference method with mesh size of the order of 0.1 km, each patch requiring a total number of meshes of the order of 10 to 100 thousands nodes.

For large system simulations, normal computing power offered by the regular PC would not be adequate. We therefore use the high power computing such as parallel or grid computing to provide the needed power. In this project, we develop grid technology for general applications to wave front propagation simulations, with particular reference to vector borne disease dispersal and transmission. The grid application developed has been extended to tsunami wave propagation and will be extended to other areas.

DEER MODEL

Dengue and Encephalitis Eradication Routines (DEER) is a numerical simulation model developed in this project to simulate the dynamics of vector borne diseases transmission. DEER is developed to simulate the distribution of *Aedes aegypti* mosquito population and the dynamics of dengue disease transmission, which has only one epidemiological cycle linking the human hosts and the vector mosquitoes.

DEER is based upon a mass balance equation for two phases of mosquito lifecycle: the winged female mosquito (represented by W) and the aquatic form (represented by A), which includes eggs, larvae and pupae. For disease transmission, the winged female mosquitoes (W) are divided into two different classes, susceptible wing (WS) and infected wing (WI). Humans are another state variables and it has 4 different sub-variables, susceptible human (HS), infected human (HI), recovered human (HR) and dead human (HD). The compartment model for dengue disease transmission is illustrated in Figure 2. The equations governing the spatial and temporal evolution of the disease in DEER are given in (1).

Let the mortality rate of the mosquitoes and aquatic forms be respectively $ALPHAW$ (day^{-1}) and $ALPHAA$ (day^{-1}). The specific rate of development of the aquatic forms into wing mosquitoes will be $CONV$ (day^{-1}). The rate of oviposition by female mosquito is $BETA$ (day^{-1}). We will consider the *Aedes aegypti* dispersal as the result of a ran-

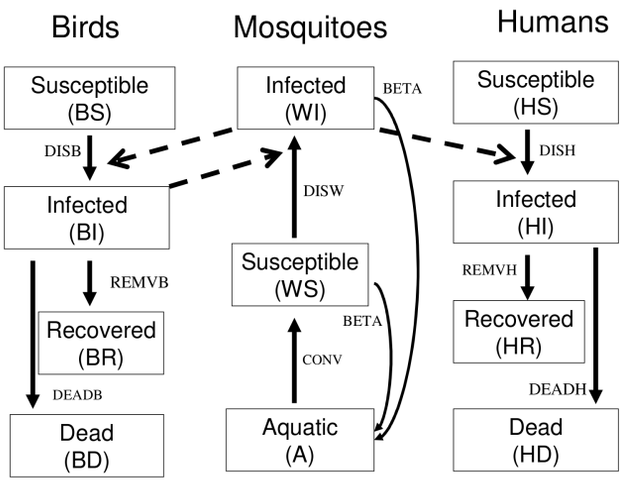


Figure 3: Compartment model for West Nile Encephalitis

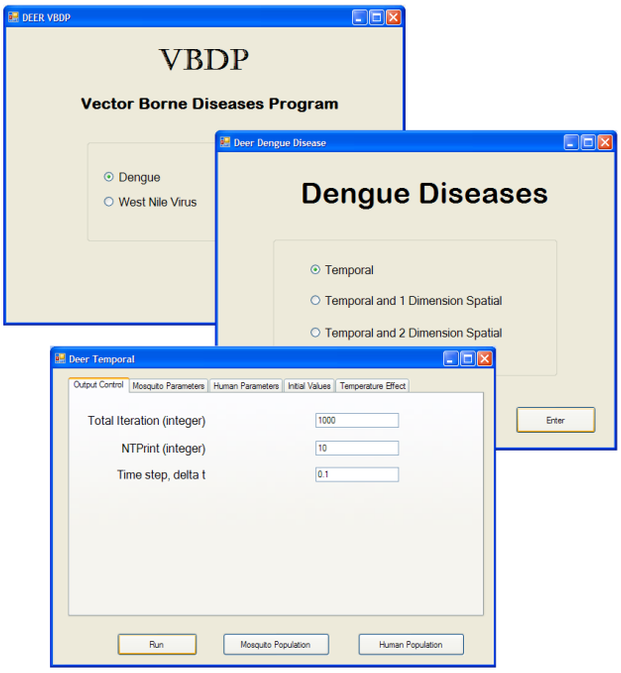


Figure 4: Graphical user interface (GUI) for DEER

dom flying movement represented by a diffusion process with coefficient *DIFF* (km^2/day) and the constant velocity flux of the wind advection is v (km/day). The density of wing mosquitoes and aquatic forms is limited by carrying capacity of wing mosquitoes *CCW* (mosquito) and aquatic forms *CCA* (mosquito). Wing female mosquitoes become infected when they bite infected humans. The transmission probability from infected human hosts to susceptible vector mosquitoes is *DIS*. Similarly, humans get dengue virus infections from the bite of infected wing female mosquitoes. *DISH* is the transmission probability from infected vector mosquitoes to susceptible human hosts. *REMVH* is recovery rate of human (day^{-1}). *DEADH* is the mortality rate of human caused by the virus. *ALPHAH* is the natural mortality rate and birth rate of human (day^{-1}). *BITR* is biting rate of mosquito (day^{-1}). *AHOST* is number of alternative hosts available as blood sources. *HN* is the total human population ($HN = HS + HI + HR$).

DEER is further enhanced to enable the simulation of the transmission of other vector borne diseases with more than one epidemiological cycle such as WNE and JE. WNE disease transmission involves another mammal like birds or horse in the epidemiological cycle linking human host and mosquito. To enable simulation of WNE transmission, bird is included as a state variable in our model. Similar to the human state variable, it has four sub-variables such as susceptible birds (*BS*), infected birds (*BI*), recovered birds (*BR*) and dead birds (*BD*). Figure 3 shows the life cycle of West Nile Virus Disease that mainly circulates between mosquitoes and birds. The virus is transmitted to human by infected mosquito bites. Humans are the end host in the life cycle of the encephalitis virus.

GRID APPLICATION

MPI commands are incorporated into the DEER algorithm so that some routines can be performed in parallel. We have successfully run DEER using our USM Campus Grid with 22 nodes, the results of which are presented and demonstrated in PRAGMA 15 and 16.

GRAPHICAL USER INTERFACE

A web-based application will enable easy access to job submission and results retrieval by users. The user can submit jobs using a web browser running either from Windows, Linux or Mac OS X operating systems. Web-based application enables simple retrieval of results and reduces data storage size. The Microsoft .Net Framework is a platform that provides tools and technologies we need to build Networked Applications as well as Distributed Web Services and Web Applications. For creating a user-friendly model for the web services, we develop the GUI for DEER model in C#.NET. Figure 4 shows the simple GUI panel for DEER.

AN EXAMPLE APPLICATION

For the purpose of presentation, a study domain with 400×400 grid cells is used for the simulations. The spatial and temporal distribution of *Aedes aegypti* mosquito population is shown in Figure 5. Model simulation begins with a small concentration of wing mosquito. The wing mosquitoes spread out to locations with water containers, which provide sites for the mosquitoes to lay eggs. A small number of mosquitoes are frequently transported by vehicles to other locations. Here, we investigate the effectiveness of insecticide spraying on controlling the dispersal of mosquitoes. We assume that insecticide is applied to kill the mosquitoes at a certain time. The amount of insecticides applied is then gradually reduced to zero. Upon application of insecticide, the density of mosquito population decreases with their dispersal contained. However, the density and dispersal of the mosquitoes eventually return to the pre-treatment level. Figure 6 shows the physical computational time consumed subject to the

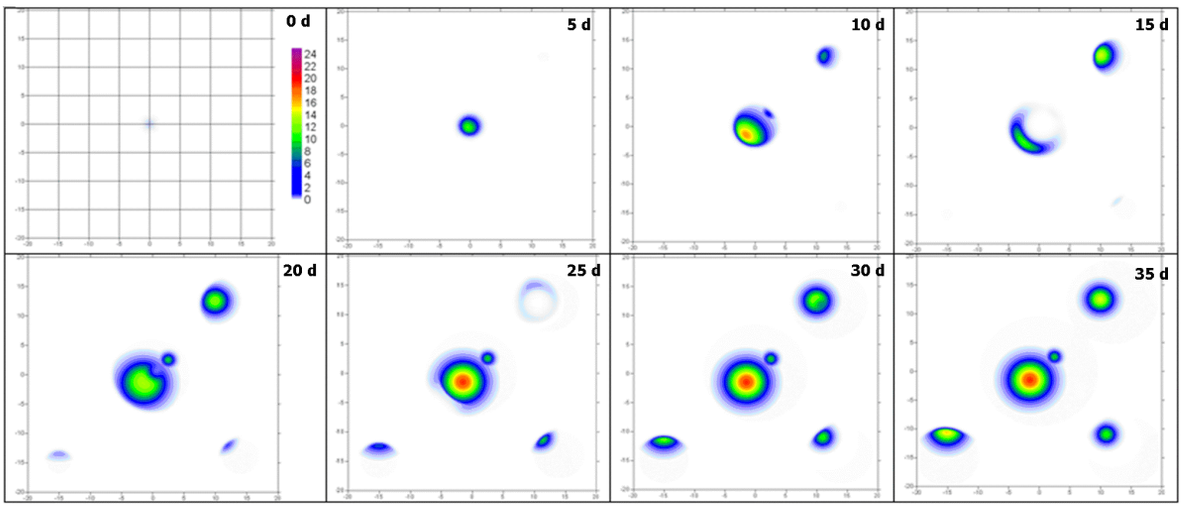


Figure 5: Mosquito Population Distribution

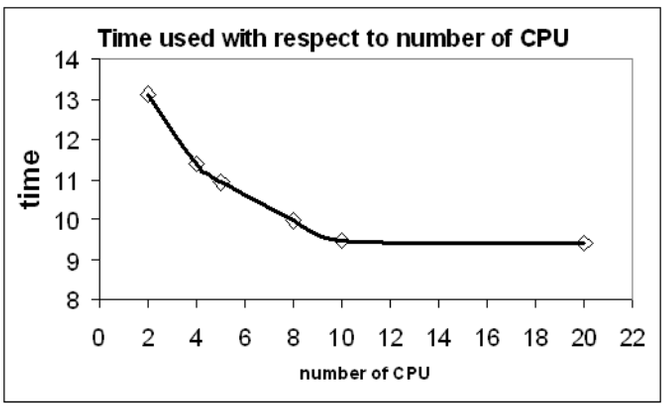


Figure 6: Computational time using different number of nodes

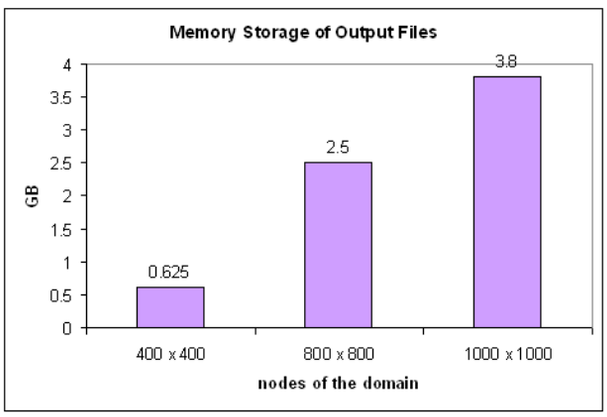


Figure 7: Memory required for different research area

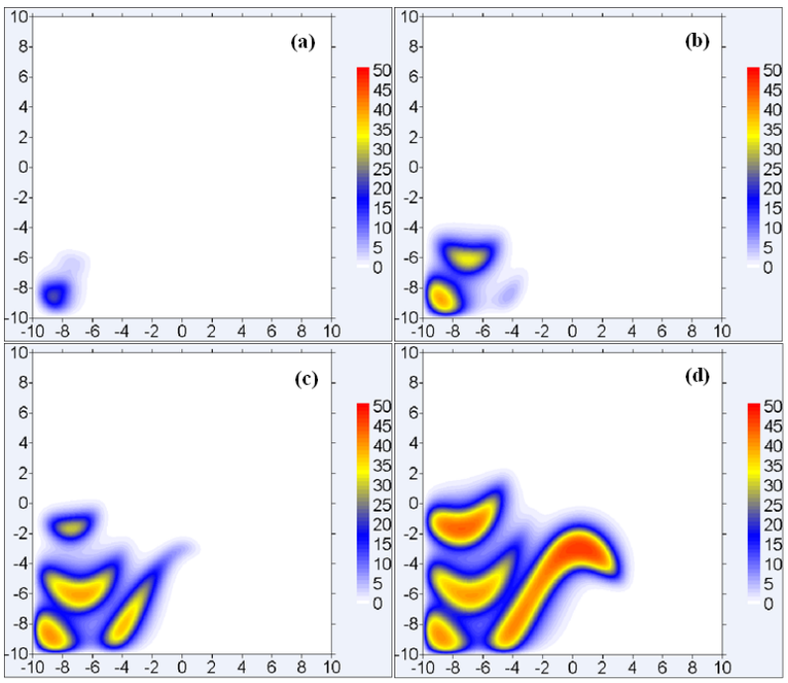


Figure 8: Dengue disease transmission

Table 1: DEER computational time on a single PC and cluster with 10 and 20 nodes

Segments	Computational time		
	Single PC	10 nodes	20 nodes
Calculation	6 hrs	50 mins	40 mins
Printing	3 hrs	60 mins	20 mins
Total	9 hrs	1 hr 50 mins	1 hr

various numbers of nodes used. The reduction of time used becomes saturated after 12 nodes. This is because the execution speed of the program is limited by the system input and output latencies such as file access and message passing over network. Figure 7 shows the memory storage needed for several sizes of study domain. The memory size needed for data storage increases exponentially as the calculation domain increases.

To examine the global-scale relationships among climate, *Aedes aegypt*i populations and transmission of dengue disease, a large study domain typically in terms of thousands of kilometers is needed. This incurs excessive demands on the computational time and storage size as the computational grid size of the study domain must be small enough to reflect the local distribution and flight reach of *Aedes aegypti*, which is only a few hundred meters (Reiter et al. 1995). Therefore, grid technology is used to speed up the computational time and to reduce storage size. This is achieved by dividing and allocating sections of program computation to several computers.

REGIONAL DENGUE SIMULATION BY GRID

A global scale of dengue disease transmission is considered in this study to demonstrate the capability of grid computing in reducing the computational cost of simulating global transmission of dengue. Table 1 shows the computational time used to run a global case of dengue transmission for a single PC and clusters with 10 and 20 nodes. A single simulation takes about 9 hours to run on a single normal PC of Intel® Pentium® 1.60 GHz and 768 MB RAM. Using a cluster with 20 nodes reduces the computational time to about half an hour; a total time reduction of more than 80 %.

DIRECTION OF FUTURE RESEARCH

DEER is undergoing continuous enhancements to upgrade its capability in simulating the dynamics of dengue disease transmission. Further, DEER is undergoing revisions for application on the PRAGMA Grid network. We hope that this project would stimulate active research and collaboration in this region. Further we have successfully extended grid technology to tsunami simulations, which typically require large computational resources, made available by Grid technology.

PROJECT PUBLICATIONS

Kew, L. M., Teh, S. Y., & Koh, H. L. (2009). Optimization of Tsunami Model TUNA by Grid Technology. In *Proceedings of the 5th Asian Mathematical Conference (AMC)*. Kuala Lumpur, Malaysia.

Koh, H. L., Lee, H. L., Teh, S. Y., & Izani, A. (2009). Dengue and Tsunami Modeling: Application of Grid Technology. In *Proceedings of 2nd Regional Conference on Ecological and Environmental Modeling (ECOMOD 2007)*. Penang, Malaysia; pp. 22–28.

Tan, K. B., Teh, S. Y., Koh, H. L., Sui, L. L., Bahari, B., & Izani, A. (2009). Modeling West Nile Virus with Grid Technology. In *Proceedings of the 16th Pacific-Rim Application And Grid Middleware Assembly (PRAGMA 16)*. Daejeon, Korea.

Teh, S. Y., Kew, L. M., & Koh, H. (2008). Application of Grid Computing in Modeling Tsunami and Dengue. In *ICTP Advanced School in High Performance and GRID Computing*. Trieste, Italy.

REFERENCES

Atkinson, M. P., Su, Z., Alphey, N., Alphey, L., Coleman, P. G., & Weln, L. M. (2007). Analyzing the Control of Mosquito-borne Diseases by a Dominant Lethal Genetic System. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, *104*(22), pp. 9540–9545.

Intergovernmental Panel for Climate Change. (2001). *Summary for policy makers climate change 2001: the scientific basis*. Cambridge University Press.

Reiter, P., Amador, M. A., Anderson, R. A., & Clark, G. G. (1995). Short Report: Dispersal of Aedes aegypti in Urban Area after Blood Feeding as Demonstrated by Rubidium-Marked Eggs. *American Journal of Tropical Medicine and Hygiene*, *52*, pp. 177–179.

Schaeffer, B., Mondet, B., & Touzeau, S. (2008). Using a Climate-Dependent Model to Predict Mosquito Abundance: Application to Aedes (Stegomyia) africanus and Aedes (Diceromyia) furcifer (Diptera: Culicidae). *Infection, Genetics and Evolution*, *8*, pp. 422–432.

World Health Organization. (2001). Chiang Mai Declaration on Dengue/Dengue Haemorrhagic Fever. *Wkly Epidemiol Rec*, pp. 29–30.

World Health Organization. (2000). *World Health Report 2000 – Health Systems: Improving Performance*. World Health Organization.

Image: Fireworks. Photograph by ©G Schouten de Jel (http://www.sxc.hu/profile/gabriel77).

PROJECT ACTIVITIES

Image: A conference room. Photograph by ©Justyna Furmanczyk (http://www.sxc.hu/profile/just4you).

ORGANISED EVENTS

APPLE CONTENT CREATION WORKSHOP

This workshop was jointly organised by the School of Computer Sciences (PPSK) and OpenSOS Sdn. Bhd. on 26–28 February 2008 at PPSK. The objective of the workshop is to learn how technology engages students in exploring their natural curiosity of the world while learning the core curriculum. Mr Ahmad Shaharuddin Amin b. Sahar, the facilitator, guided the participants from familiarising themselves with a Mac to using applications in the iLife '08 suite.

INTRODUCTORY WORKSHOP TO GRID COMPUTING

AP Chan Huah Yong organised and facilitated this one-day workshop on 1–2 March, 2008 at PPSK, assisted by researchers from the Grid Computing Lab (GCL). All ROs and RAs from the sub-project teams were invited to get a first taste of Grid computing technologies, especially from an end-user point perspective. The workshop included an introductory talk on Grid computing and hands-on sessions on grid credentials, the *Globus* toolkit and the *GridSphere* portal framework.

WORKSHOP ON GRID COMPUTING STRATEGIES & SECURITY ISSUES

Held in Pulau Perhentian on 2–4 July, 2008, this workshop focusing on grid security and monitoring. During a co-located session, ROs and RAs from the respective sub-project teams updated each other on their progress, and received training on the *Trac* Wiki system. We would later use *Trac* to post information and progress updates on their respective sub-projects in one centralized repository.

THE 2ND INTEL MULTICORE WORKSHOP

This event was co-organised with Intel and held in Bangunan Eureka on 22–25 July 2008. AP Chan Huah Yong hosted a workshop including a hands-on session there.

PRAGMA INSTITUTE, PRAGMA 15 & DFMA 2008

The PRAGMA 15 Workshop was held and hosted by USM on 22–24 October, 2008, while PRAGMA Institute met at the same venue earlier on the 21 & 22. All sub-projects were selected for poster exhibition, while AP Chan Huah Yong, AP Phua Kia Kien and Prof. Koh Hock Lye's projects were chosen for showcase demonstrations. In addition, The International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008) was also held in conjunction on the 21 & 22. DFmA 2008 accepted 4 papers from Grid@USM members.

MICROSOFT HIGH PERFORMANCE COMPUTING (HPC) WORKSHOP

This 3-day workshop was held at the School of Computer Sciences, USM on 8–10 April 2009, organised by AP Chan Huah Yong. He was also joined by Mr Ang Sin Keat and Ms Kheoh Hooi Leng.

Participants were introduced to the infrastructure of Windows High Performance Computing (HPC) Server 2008 infrastructure, as well as deploying and administrating a Windows HPC Server 2008 Cluster. Topics on the second day included network configuration, diagnostics, performance tuning and scheduling jobs. The third and final day saw the participants developing C/C++ MPI applications with Visual Studio and MPI.NET.

Image: a conference room. Photograph by ©Ann Petersen (http://www.sxc.hu/profile/Lilja).

OTHER EVENTS

PRAGMA 14

AP Bahari Belaton was present at PRAGMA 14 held in March 2008 in Taichung, Taiwan. He and Dato Prof. Muhd. Idiris, then USM's Deputy Vice Chancellor for Research and Innovation, were invited as co-chairs and took the opportunity to promote and raise awareness about PRAGMA 15, which was to be held in USM in October the same year.

COURSE ON ENABLING GRIDS FOR E-SCIENCE

Mr Aloysius Indrayanto and Ms Kheoh Hooi Leng attended this training course at MIMOS Berhad in Kuala Lumpur on 10–14 December, 2007.

PBS TRAINING AND GRIDING OF UNIVERSITIES

Mr Tan Choo Jun and Ms Tan Chin Min participated in this training held at International Islamic University Malaysia (IIUM) on 5–7 May 2008.

ICTP ADVANCED SCHOOL IN HIGH PERFORMANCE AND GRID COMPUTING

Dr Teh Su Yean attended this course at the Abdus Salam International Centre for Theoretical Physics (ICTP), Trieste, Italy on 3–14 Nov 2008.

THE 1ST KNOWLEDGE GRID MALAYSIA FORUM 2009

This event was held on 18-19 March, 2009 at the IIUM. AP Chan Huah Yong was invited to conduct a hands-on session entitled "USM Grid (Globus)". Three USM ROs i.e. Mr Ang Sin Keat, Ms Tan Kah Bee and Ms Sui Lee Lin also attended the two-day forum.

SCOPE ICT CAREER FAIR

This event was held in conjunction with the SCoPE ICT Expo on 18 March 2009 at Suntech Penang Cybercity. Grid@USM members exhibited 6 posters about their sub-projects at rented booths at the fair, including:

- AP Chan Huah Yong
  - Virtual Distributed Database Management System
  - Mobile Desktop Grid
  - Campus Grid Portal
- Dr Vincent Khoo Kay Teong
  - Enterprise Grid Architecture for Collaborative B2B Information Interchange
- Prof. Koh Hock Lye
  - In-house Tsunami Propagation Model TUNA-M2: Application of Grid
  - Grid Application: Modelling Mosquitoes Distribution

ROs attending the event include Mr (Mr) Johnson Foong, Mr Kor Chan Hock, Mr Aloysius Indrayanto and Mr Tan Choo Jun.

PRAGMA 16 & IDRIC

PRAGMA 16 was held on 23–26 March 2009 at DCC, Deajon, Korea, in conjunction with the International Workshop on Infectious Disease Researches in Cyberinfrastructure (IDRiC). 4 posters from USM participated in PRAGMA 16:

- Prof. Koh Hock Lye
  - Modeling Dengue Infection for Malaysia
  - Modelling West Nile Virus with Grid Technology
- AP Chan Huah Yong – Virtual Distributed Database Management System
- Mr Mohd. Azam Osman – Mobile Desktop Grid

Image: Books at a library. Photograph by ©Carlo Lazzeri (http://www.sxc.hu/profile/mvttley).

PUBLICATIONS

JOURNAL ARTICLES

Phua, K. K. & Wong, V. C. (2007). Developing A Compelling 3D Animation and Multimedia Presentation Using Blender for INFORMM's Opening Ceremony. *Malaysian Journal of Medical Sciences*, *14*(Supplement 1).

Al-Mistarihi, H. H. E. & Chan, H. Y. (2009). On Fairness, Optimizing Replica Selection in Data Grids. *IEEE Transactions on Parallel & Distributed Systems*. (forthcoming).

Chan, H. Y. & Al-Mistarihi, H. H. E. (2008). Replica Management in Data Grid. *International Journal of Computer Science and Network Security*.

Chan, H. Y. & Al-Mistarihi, H. H. E. (2008). Replica Optimization Service in Data Grids. *Sciences Publications*.

Younis, M. I., Zamli, K. Z., Klaib, M. F. J., Che Soh, Z. H., Che Abdullah, S. A., & Mat Isa, N. A. (2009). Assessing IRPS as an Efficient Pairwise Test Data Generation Strategy. *International Journal of Advanced Intelligence Paradigms*. (forthcoming).

Younis, M. I., Zamli, K. Z., & Mat Isa, N. (2008). IRPS – An Efficient Test Data Generation Strategy for Pairwise Testing. *Lecture Notes in Artificial Intelligence*, *5177*.

Zafri, M., Fadhilah, N. K., & Phua, K. K. (2008). Using Xgrid to Improve FASTA Efficiency for Alignment of Multiple DNA and Protein Sequences. *Malaysian Journal of Medical Sciences*, *15*(Supplement 1).

CONFERENCE & WORKSHOP PROCEEDINGS

Zafri, M., Sanjay, K. C., & Phua, K. K. (2009). Implementation Methods for Estimating Haplotypes with GRID Computing Technology. In *Compendium of Abstracts for the 14th National Conference on Medical and Health Sciences (NCMHS)*.

Ahmed, M. & Talib, N. A. (2008). iNet-Grid: A Real-Time Grid Monitoring and Troubleshooting System. In *Proceedings of the International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008)*. Penang, Malaysia; pp. 68–72.

Indrayanto, A. & Chan, H. Y. (2008). Application of Game Theory and Fictitious Play in Data Placement. In *Proceedings of the International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008)*. Penang, Malaysia; pp. 79–83.

Kew, L. M., Teh, S. Y., & Koh, H. L. (2009). Optimization of Tsunami Model TUNA by Grid Technology. In *Proceedings of the 5th Asian Mathematical Conference (AMC)*. Kuala Lumpur, Malaysia.

Koh, H. L., Lee, H. L., Teh, S. Y., & Izani, A. (2009). Dengue and Tsunami Modeling: Application of Grid Technology. In *Proceedings of 2nd Regional Conference on Ecological and Environmental Modeling (ECOMOD 2007)*. Penang, Malaysia; pp. 22–28.

Lim, L. T. & Schwab, D. (2008). Limits of Lexical Semantic Relatedness with Ontology-based Conceptual Vectors. In *Proceedings of the 5th International Workshop on Natural Language Processing and Cognitive Science (NLPCS'08)*. Barcelona, Spain; pp. 153–158.

Schwab, D. & Lim, L. T. (2008). Blexisma2: a Distributed Agent Framework for Constructing a Semantic Lexical Database based on Conceptual Vectors. In *Proceedings of the International Conference on Distributed Frameworks & Applications 2008 (DFmA 2008)*. Penang, Malaysia; pp. 102–110.

Tan, K. B., Teh, S. Y., Koh, H. L., Sui, L. L., Bahari, B., & Izani, A. (2009). Modeling West Nile Virus with Grid Technology. In *Proceedings of the 16th Pacific-Rim Application And Grid Middleware Assembly (PRAGMA 16)*. Daejeon, Korea.

Teh, S. Y., Kew, L. M., & Koh, H. (2008). Application of Grid Computing in Modeling Tsunami and Dengue. In *ICTP Advanced School in High Performance and GRID Computing*. Trieste, Italy.

Ting, T. T. & Khoo, V. K. T. (2009). B2B Standardized Information Interchange Challenges – A Study on Standardization versus Personalization. In *Proceedings of the 2009 International Conference on Advanced Management Science (ICAMS 2009)*; pp. 244–248.

Ting, T. T. & Khoo, V. K. T. (2008). Personalizable Information Interchange Architecture for Educational Institutions. In *Proceedings of the 3rd International Conference on e-Commerce with Focus on Developing Countries (ECDC'08)*. Isfahan, Iran.

Younis, M. I., Zamli, K. Z., & Mat Isa, N. A. (2008). A Strategy for Grid Based *t*-Way Test Data Generation. In *Proceedings of the International Conference on Distributed Frameworks & Applications (DFmA 2008)*. Penang, Malaysia; pp. 73–78.

THESES

Ahmed, M. (2009). A Real Time Distributed Network Monitoring and Security Monitoring Platform (RTDNMS). PhD thesis, School of Computer Sciences, Universiti Sains Malaysia.

Al-Mistarihi, H. H. E. (2009). A Data Grid Replica Management System with Local and Global Multi-Objective Optimization. PhD thesis, School of Computer Sciences, Universiti Sains Malaysia.

Indrayanto, A. (2009). Initial File-Placement in Data Grid Environment using Game Theory and Fictitious Play. MSc thesis, School of Computer Sciences, Universiti Sains Malaysia.

BOOKS

Zamli, K. Z., Che Abdullah, S. A., Younis, M. I., & Che Soh, Z. H. (2009). *Software Testing Module*. (forthcoming). OpenUniversity Malaysia and Pearson Publishing.

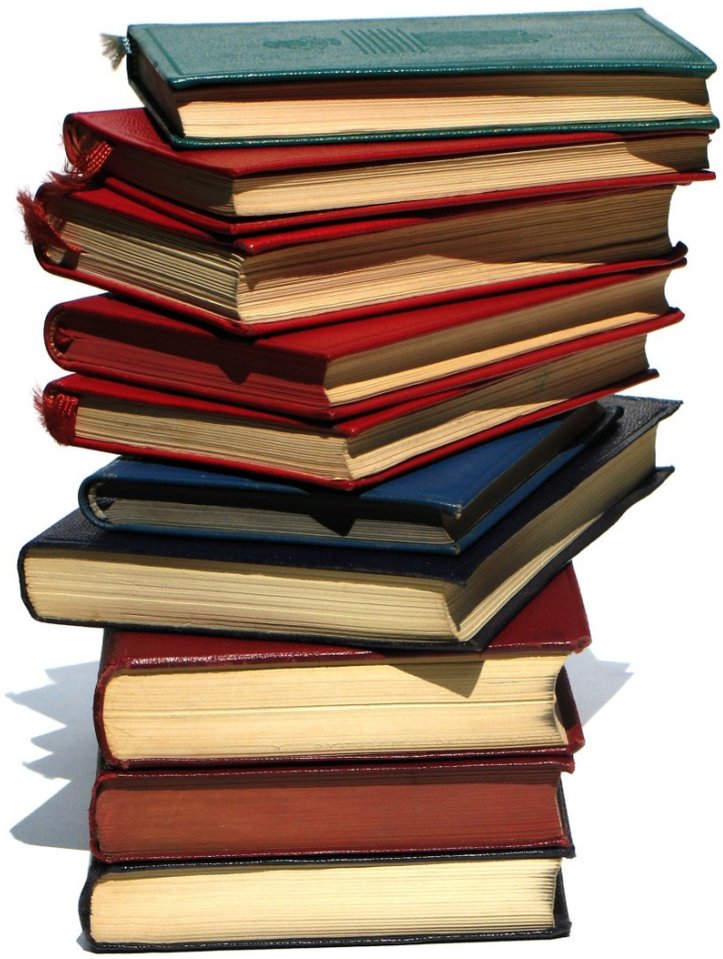


Image: A huge pile of books. Photograph by ©Sanja Gjenero (http://www.sxc.hu/profile/lusi).

Grid@USM

Grid computing can provide easy and transparent access to high performance computing by aggregating geographically distributed resources to act as a single powerful resource. With grid computing technologies, users can access to any virtual resources such as computers, special instruments, software applications and databases in a seamless manner.

Grid computing is by its nature highly distributed geographically, consists of highly specialised equipment (storage, grid engines and management tools), as well as expensive. It is as such an ideal example of common, core computational resources to be created and pooled among aspiring researchers who require high performance resources. Equally important, Universiti Sains Malaysia (USM) as a research university (RU) needs a focused, holistic and dedicated effort to build a sustainable human capital in grid computing.

It is with USM's R&D needs and sustainability in mind that Grid@USM, a grid computing research cluster to address the problems mentioned above, came into being. Grid@USM's pilot project, led by AP Bahari Belaton and entitled *Grid Computing Cluster – the Development and Integration of Grid Services and Applications*, was initiated under an RU research grant. This book records our achievements thus far, in the form of technical reports on 8 sub-projects. These sub-projects include both grid middleware and tool enablers, as well as applications from various domains, spanning grid infrastructure setup, grid monitoring, virtual distributed databases, 3D rendering, natural language processing, B2B standards component modeling, software testing and disease modeling.

Platform for Information & Communication Technology Research
Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

ISBN 978-983-3986-58-3