Design Framework of Compute Power Market for Peer-to-Peer and Grid Environments

Gian Chand Sodhy, Fazilah Haron, Tan Tien Ping, Anthony Wong Kee Ling, Wei Hoo Chong, Chan Huah Yong and Rajkumar Buyya*

School of Computer Science
Universiti Sains Malaysia
11800 Penang, Malaysia
[sodhy ,fazilah, tienping, antjwkl, tomweihc, hychan] @cs.usm.my

** School of Computer Science & Software Engineering*
The University of Melbourne
Victoria 3010, Australia
raj@cs.mu.oz.au

Abstract

Peer-to-Peer (P2P) and grid computing are concerned with pooling and coordinating resources across the internet. Similar issues arise in both, for example resource discovery, monitoring, resource management, communications and security. Resource management increases the utilization of resources and enables uniform access to a wide variety of resources over a network of computers. Market-based resource management incorporates the notion of resource trading by regulating the resource supply and demand with necessary charging. Compute Power Market (CPM) is a market-based resource management on Internet-wide computational resources. It transforms computers connected across the internet into a computational market by renting the processing power, storage and special devices from idle resources (computers).

In this paper, we present the design framework of CPM using P2P (specifically JXTA technology). In doing so we highlight the services that might be required if the same framework were to apply on the grid (using Globus technology). We have completed the development of CPM on P2P. We will test the suitability of the proposed framework in scheduling jobs using an image processing application. By comparing the services required in both environments it helps us to further understand how users can distribute their jobs to these two platforms transparently.

Keywords*:* Peer-to-peer, grid computing resource trading, Compute Power Market.

1. Introduction

To handle huge amounts of information and services, more computing power is needed. Computer resources are being grouped together to cater for the increasingly power-hungry applications. At the same time, computers across the world are not being fully utilized. The machines are either idle or occupied by screen savers. This unused computational power available on machines across the Internet can be utilized for solving resource- hungry applications. Peer-to-Peer (P2P) and grid computing are two different technologies pooling and coordinating resource to taking advantage of this [1]. This has led to the development of various Resource Management systems [2, 3, 4, 5, 6] which aims at utilizing these underutilized resources.

The idea of Compute Power Market was first proposed in [7]. Compute Power Market transforms the grid computing environment into a computational market that introduces resource trading on idle computers across the Internet. A number of grid computing projects worldwide have successfully exploited this paradigm for solving specific application areas. SETI@home [8], a scientific experiment that uses Internet-connected computers in the search for extraterrestrial intelligence tries to solve sophisticated problem using geographically distributed idle resources. The construction of research test-bed like Nimrod/G [2] allows scientists and engineers to model whole parametric experiments and transparently stage the data and program at remote sites, and run the program on each element of a dataset on different machines and finally gather results from remote sites to the user site [9].

Modern computing has evolved from client-server to web-based computing, and now peer-to-peer [10].

Peer-to-peer (P2P) technology adopts a network-based computing style that neither excludes nor inherently depends on centralized control points. Thus, it manages to increase the utilization of the information, bandwidth and computing resources of the Internet. P2P is described by Fortune Magazine as one of the four technologies that will shape the Internet's future [11]. JXTA Technology, started by Sun Inc., it is open-source and co-developed by many other contributors from various fields who believed that JXTA will become the lingua franca for distributed computing that also uses peer-to-peer technology.

In the rest of this paper, we will briefly review several related works followed by the architecture design of CPM/P2P and experimental plan.

2. Related Work

Java Market [12] is an effort to utilize wasted computational power using Java programming language and web technology. The goal of Java Market is to allow sharing of computational resources between heterogeneous machines over the Internet. An Internet user with different machine or operating system can contribute their computational resources or submit jobs to other machines over the Internet without having to install the entire system. They should be able to participate as long as they have a Java-capable Web browser. Java Market transforms user submitted Java applications into a Java applet and transfers it to a contributing machine and executes it using the web browser. Users do not need to modify existing code in order to submit it to the Java Market, provided it is in Java language.

Nimrod-G is an economic or market based resource management and scheduling system for a grid-enabled system [3]. Different from other systems, it emphasis on resource trading and quality of service based scheduling. Nimrod-G is built by incorporating the idea of market model, where resource owner can sell out their idle resources for usage by others. It is in contrast with the concept of utilizing free resources as popularize by seti@home. Besides that, Nimrod-G is utilizing the resources in grid, instead of normal generic computer as in seti@home. User of the system can specify deadline or budget that user willing to spend to complete a job. The system can perform resource reservation and negotiation so that user knows whether an experiment can be completed within the requirement set by user. Nimrod-G enables the creation parameter sweep application by using parameter sweep specification language. (Nimrod-G handles tasks related to resource discovery, mapping jobs to appropriate resources, data and code staging and gathering results from multiple Grid nodes back to the home node.) Nimrod-G allows service providers to contribute their resources to be used by others, and receive a reward in return.

Project JXTA helps to create a common platform that makes it simple and easy to build a wide range of distributed services and applications in which every device is addressable as a peer, and where peers can bridge from one domain into another [13]. JXTA technology leverages open standards like XML and Java technology that include the ability for shells to connect commands together using pipes to accomplish complex tasks. It builds up from a set of basic functions to support P2P applications.

JXTA technology comprises a set of protocols. Each protocol is defined by one or more messages exchanged among participants of the protocol. Each message has a pre-defined format, and may include various data fields. These protocols are defined to be independent of programming languages (can be implemented in C/C++, Java, Perl, etc.) and transport protocols (TCP/IP, HTTP, Bluetooth, etc.). The JXTA technology protocols standardize the manner in which peers:

- discover each other,
- self-organize into peer groups,
- advertise and discover network resources,
- communicate with each other, and
- monitor each other.

Globus architecture is the de facto standard for computational grid. Globus research focuses on issues associated with building the computational grid infrastructure and problems that arise in designing and developing applications that use grid services. The Globus project provides software tools that make it easier to build computational grids and grid-based applications. These tools are known as the Globus Toolkit. This toolkit is used by many organizations to build computational grids that can support their applications.

The Globus toolkit is an implementation of a "bag of services" architecture, which provides application and tool developers with a set of services. Each Globus toolkit component provides a basic service, such as security, communication, resource allocation, resource discovery and monitoring. Each core service defines an application program interface (API) that provides a uniform interface to a local service [14]. Meanwhile higher level services use core services to implement more complex global functionality. Different applications and tools can combine these services in different ways to construct "grid-enabled" systems.

3. The Enhanced Framework of CPM

There are three basic entities in CPM, namely resource provider, resource consumer and the market as mentioned in our earlier paper [15]. These entities interact with one another via a set of utilities and services such as resource discovery, scheduling, job monitoring and accounting with the market as the mediator. Figure 1 shows the interaction among the different CPM components.

CPM agent has to be installed in both provider and consumer. It contains the fundamental services of CPM. The provider can use this agent to join a market and publish its resources. The agent on the consumer side will help discover available resources and select the resources that match consumer requirements. Provider and consumer do not interact with each other directly. Everything is handled by the CPM agent. This includes trading, job execution and issuing of bills. The market server contains an administrator which creates a market and maintains a market repository.

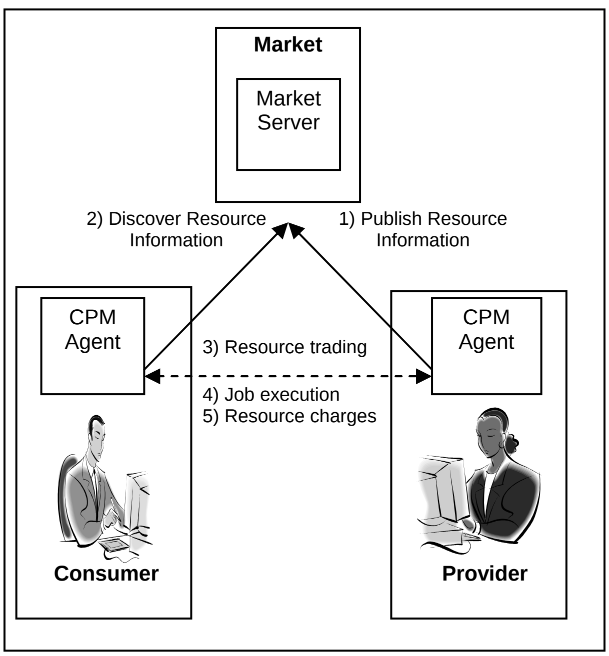

Figure 1: Interaction among CPM entities

There are four layers in the CPM, as shown in figure 2. Layer 1 connects geographically distributed computational devices across the Internet. Layer 2 is the middleware comprising JXTA or Globus protocols and security. Both JXTA and Globus provide similar services, for example resource discovering service and communications service. These services are used by layer 3 – CPM core engine. The Core Engine layer contains the main components of CPM, which are scheduler, market server and accounting modules. The top layer contains application that makes use of CPM core engine.

Layer 4 Applications	CPM Agent, compute intensive application
Layer 3 CPM core engine	Scheduler, Accounting, Market Server
Layer 2 P2P/Grid middleware	JXTA/Globus Protocols
Layer 1 P2P/Grid fabric	Internet

Figure 2: CPM Framework for P2P and Grid Environment

4. CPM Core Engine

An essential part of the CPM framework is the core engine. It consists of three main components, which are market server, scheduler and accounting. We shall describe these components in terms of P2P and grid by highlighting the specific services required for each.

Market is a meeting point where resource providers publish their resources and consumers discover these providers. The implementation of market server in JXTA requires three essential functions, namely market manipulation, advertisement management and rendezvous, and market list management. Resource providers publish their resource information to the market through the advertisement, while consumer discovers these advertisements and selects the appropriate providers. The function of market manipulation module allows the creation, removal and joining of market. The advertisement management module manages the JXTA resource information, in addition to the existing available JXTA functions. For Globus protocol, Market server is managed using Monitoring and Discovery Service (MDS). Resource providers publish their resource information through Grid Resource Information Service (GRIS). GRIS can respond to queries for information about a local machine. GRIS can be configured to register itself with aggregate directory services – Grid Index Information Service (GIIS) so that those services can in turn provide information about the machine to others. Thus, consumers may discover these resources through query to the either GRIS or GIIS.

The job scheduler consists of four main components, namely metascheduler, local scheduler, job monitoring and discovery. The relationship between metascheduler and local scheduler is like a master and worker. Metascheduler decides on the node where a task is allocated, based on a global scheduling policy and algorithm, but local scheduler may accept and chooses the sub job to run on the local system based on the system policy. Scheduling can be done based on deadline or cost constraint [16]. Multiple customers can schedule their job to multiple providers through virtual contract. Each contract describes one or more sub-jobs that customer requires provider to complete. Job Monitor component monitors the status of job and trigger metascheduler to reschedule if failure occurred. The discovery component is used for searching for available resource providers over the net. For JXTA Protocol, the discovery component makes use of JXTA Discovery Service to locate suitable resources. Meanwhile in Globus Protocol, MDS was used to discover the resources.

The accounting module remains the same in P2P and grid. This is due to the similarity of functionality and services. It consists of three components, namely updating database, reporting and billing. The accounting component works closely with scheduler. The scheduler invokes the accounting module before starting a job to log important information (resource cost, consumer information and etc) and after finishing a job to calculate total charges for the job. The calculation is based on resource cost and total execution time on provider machine. After calculating the charges, a summary of usage will be generated and send to consumer. All necessary information is kept in a database in both provider and consumer. This is to allow the provider to retrieve the overall resource rental information and the consumer to check the charges on submitted job. Figure 4 shows a scenario of a complete cycle of resource trading in CPM that we have developed. The scenario is applicable in both P2P and grid environment. The provider uses the CPM agent to discover and join a market. Then the agent will help the provider to publish information about its resources to the market. The resource information is stored in the market repository and maintained by the Market Server.

The consumer uses its agent to discover and join CPM market, as well as discover available resources in the market. The agent will return a list of resources available, which the metascheduler will rank according to consumer defined optimization criteria. Once a provider is selected, a contract will be generated and sent to local scheduler (on provider machine). The trader component in both metascheduler and local scheduler will complete the price negotiation section. Once local scheduler accepts the contract, metascheduler transfers the job to the local scheduler. Local scheduler executes the job and returns the result to metascheduler. The scheduling information is logged (starting from the submission of job until the return of the result to consumer machine), and sent to the accounting service that will calculate the resource charges and bill the consumer. These modules are packaged and refined to create the top layer of CPM framework, the CPM Agent. In the future, the CPM core engine can be reused to create new application or adopted in other applications.

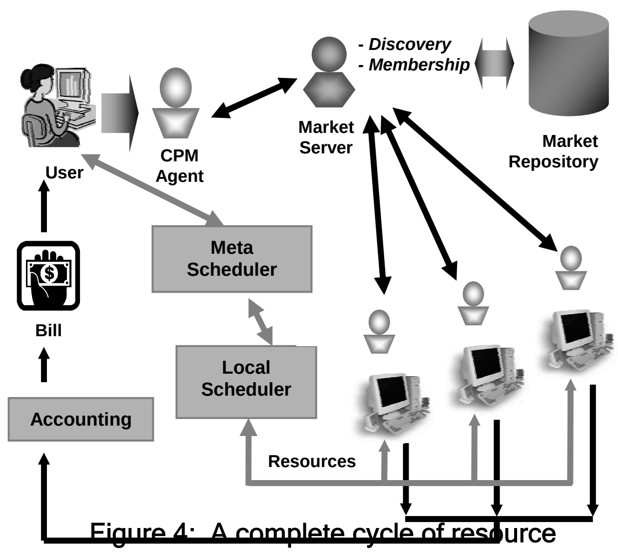

Figure 4: A complete cycle of resource trading in CPM architecture

5. Application and Experimental Plan

We have developed a simple image processing application to test CPM on P2P environment. The application is used for segmenting object from an image using adaptive thresholding method [17].

Figure 5 shows the original image and also the resulted image after being processed with different mask size. The mask size is a parameter that can be changed to produce better segmented object. The larger the mask size the better the segmented object but the longer processing period it takes

The processing can be parallelized by dividing an image into several sections. The whole image is considered to be a job and each sections of the image correspond to a sub-job. For example, an image can be divided into four sections (top right, top left, bottom right and bottom left). This image is considered to be a job with four sub-jobs.

With this application, we hope to demonstrate the suitability of our proposed framework in P2P environment in scheduling job to providers under different cost and deadline, using cost and time optimization scheduling algorithm.

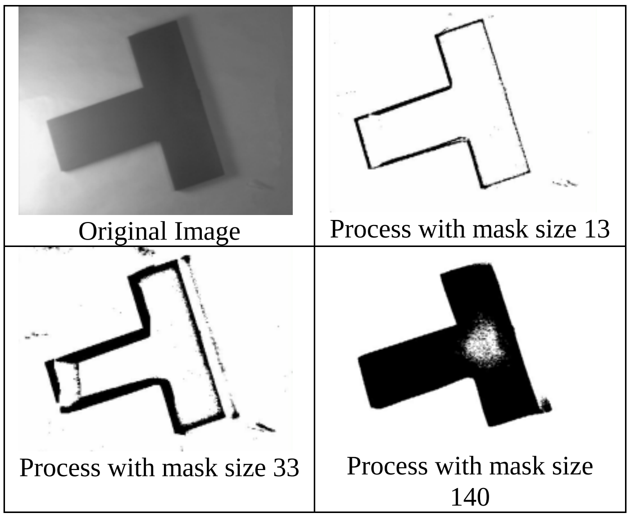

Figure 5. Processed images with different mask size which have been resized.

6. Conclusion and Future Work

We have implemented the proposed CPM framework in P2P environment. The development of the core CPM component has been completed and we will test the whole system using the image processing application. The next task is to implement the system under grid environment. We have also highlighted the suitability of the framework in grid and discussed how the existing system could be adopted on Globus using specific Globus services.

Our future work includes building of a super node which acts like a gateway for user to submit job to the entire platform transparently. This will increase the scalability of the whole architecture. Scheduler's scheduling algorithms need to be adjusted to allow job to be scheduled in these two environments in order to minimize the job completion time. Job monitoring component on the other hand has to be updated to allow management of job, so that job is completed accordingly. This will also involve changes to the accounting module, where the correct amount of cost needs to be recorded regardless of where the job is executed (either in P2P or grid).

On the provider side, local scheduler can be enhanced by allowing job reservation for customer with different price at different time for instance. Besides that, a module for monitoring provider's load will also be beneficial.

7. Acknowledgement

We gratefully acknowledge the research grant provided by Universiti Sains Malaysia, Penang that has resulted in this article.

8. References

[1] Ian Foster, Adriana Iamnitchi, On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing, *2nd International Workshop on Peer-toPeer Systems (IPTPS'03)*, Berkeley, CA, Feb. 2003

[2] Rajkumar Buyya, David Abramson and Jonathan Giddy, "Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid Nimrod/G", *The 4th International Conference on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*, Beijing, China, 2002.

[3] D. Angulo, I. Foster, C. Liu, and L. Yang, "Design and Evaluation of a Resource Selection Framework for Grid Applications", *Proceedings of IEEE International Symposium on High Performance Distributed Computing (HPDC-11)*, Edinburgh, Scotland, July 2002.

[4] K. Czajkowski, A.K. Demir, C. Kesselman, M. Thiebaux, "Practical Resource Management for Grid-based Visual Exploration", *Proceedings of the Tenth International Symposium on High Performance Distributed Computing (HPDC-10), IEEE Press,* August 2001.

[5] I. Foster, C. Kesselman, C. Lee, R. Lindell, K. Nahrstedt, A. Roy, "A Distributed Resource Management Architecture that Supports Advance Reservations and Co-Allocation", *International Workshop on Quality of Service*, London, May 1999.

[6] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, S. Tuecke, "A Resource Management Architecture for Metacomputing Systems", *Proceedings of IPPS/SPDP '98 Workshop on Job Scheduling Strategies for Parallel Processing,* Florida, 1998

[7] Compute Power Market: Towards a Market-Oriented Grid, Rajkumar Buyya and Sudharshan Vazhkudai, *The First IEEE/ACM International Symposium on Cluster Computing*

and the Grid (CCGrid 2001), Brisbane, Australia, May 16-18, 2001.

[8] W. T. Sullivan, D. Werthimer, S. Bowyer, J. Cobb, D. Gedye, D. Anderson. A new major SETI project based on Project Serendip data and 100,000 personal computers, *Proceedings of the Fifth International Conference on Bioastronomy*, Italy, 1997.

[9] Mark Baker, Rajkumar Buyya and Domenico Laforenza, "The Grid: International Efforts in Global Computing", *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet (SSGRR 2000)*, l`Aquila, Rome, Italy, July 31 - August 6, 2000.

[10] Li Gong, "JXTA all About Community", *The O'Reilly Peer-to-Peer and Web Services Conference,* Washington D.C., USA, Sept 18-21, 2001.

[11] Li Gong, "Project JXTA: A Technology Overview", *Project JXTA Documentation*, April 25, 2001

[12] Yair Amir, Baruch Awerbuch and Ryan S. Borgstrom, "The Java Market: Transforming the Internet into a Metacomputer", *Technical Report CNDS-98-1*, Johns Hopkins University, 1998.

[13] Sun Microsystems Inc, "Project JXTA: An Open, Innovative Collaboration", Project JXTA Documentation, April 25, 2001

[14] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A resource management architecture for metacomputing systems", in *The 4th Workshop on Job Scheduling Strategies for Parallel Processing*, Florida, 1998.

[15] Anthony K.L.Wong, Liew Chee Sun, Gian Chand Sodhy, Chan Huah Yong, Fazilah Haron, Rajkumar Buyya, "Design of Generic Components for Compute Power Market", *Proceedings of the IASTED International Conference on Computer Science and Technology (CST 2003),* Cancun, Mexico, 19-21 May 2003.

[16] Rajkumar Buyya, *Economic-based Distributed Resource Management and Scheduling for Grid Computing*, Phd. Thesis, Monash University, Australia, April 2002.

[17] Bob Fisher, Simon Perkins, Ashley Walker and Erik Wolfart, Hyper Media Image Processing Reference, 1994 - http://www.cee.hw.ac.uk/hipr/html/hipr_top.html