DESIGN FRAMEWORK OF GENERIC COMPONENTS FOR COMPUTE POWER MARKET

Anthony Wong Kee Ling, Liew Chee Sun, Gian Chand Sodhy, Chan Huah Yong, Fazilah Haron and Rajkumar Buyya*

School of Computer Science
Universiti Sains Malaysia
11800 Penang, Malaysia

* School of Computer Science & Software Engineering
The University of Melbourne
Victoria 3010, Australia

ABSTRACT

Current computational devices are reaching the limit in terms of meeting the ever increasing needs of high performance computational power. However, there is plenty of idle computational power not being used efficiently since most of this power resides on individual machines, only waiting to be harnessed in an orderly manner. Peer-to-peer (P2P) technology provides the infrastructure and methodology of using this idle power which is otherwise not fully utilized. Compute Power Market (CPM) has been proposed to bring together the providers and users of computing power in a systematic manner. This paper discusses our effort at developing a CPM/P2P framework, based on JXTA which is one of the latest P2P technology. We have managed to develop some essential modules needed to bring together providers and consumers of computational power.

KEYWORDS

Resource Trading, Compute Power Market, JXTA.

1. INTRODUCTION

To handle huge amounts of information and services, more computing power is needed. Computer resources are being grouped together to cater for increasingly power-hungry applications. This demand leads to the design of a market-based resource management system, called Compute Power Market (CPM) [1].

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 [1]. SETI@home [2], 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 [3] 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 [4].

However, the regulation of resource demand and supply is missing in most of these grid computing infrastructures. Accordingly, Compute Power Market was proposed to be a market-based economic paradigm for resource management adopting the Grid Architecture for Computational Economy (GRACE) [5] for high-end computing system into Internet-wide low-end computing machines to create a computational marketplace.

Modern computing has evolved from client-server to web-based computing, and now peer-to-peer [6]. 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 [7]. JXTA Technology, started by Sun Inc., 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 and JXTA technology followed by the architecture design of CPM/P2P using JXTA technology.

2. RELATED WORK

The SETI@Home project is the first attempt to use large-scale distributed computing to perform a sensitive search for radio signals from extraterrestrial civilizations. The SETI, or the Search for Extraterrestrial Intelligence, is a scientific effort seeking intelligent life outside Earth. With the assumption that extraterrestrial civilizations wishing to make contact with other races will broadcast a radio signal, SETI researchers search for signals in a data stream which is collected by a 305-meter radio telescope at National Astronomy and Ionospheric Center in Arecibo, Puerto Rico. The data analysis task involves intensive calculation that requires more computational power than is available in existing supercomputers. Fortunately, it can be easily broken up into little pieces that can all be worked on separately and in parallel. The UC Berkeley SETI team had discovered the potential and unlimited computational resource - the idle personal computers spread around the Internet. Instead of letting screen savers activate and do nothing for most of the time, world wide volunteers were convinced to contribute their computers when they are not using it. The SETI@home participants can get a chunk of data from SETI@home server over the Internet, analyze that data, and then return the results using the SETI@home client program. The SETI@home client program is executed only when the computer is idle. Whenever the participants need their computer back, this program instantly gets out of the way and only continues the analysis task when the computer is idle again. The SETI@home project had been a success story to perform data intensive analysis using distributed idle computers.

The Java Market [8] is another 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. In other words, every Internet user that use any kind of machine, and any kind of operating system can contribute their computational resources or submit jobs to other machines over the Internet without any installation as long as they have a Java-capable Web browser. Conceptually, the 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's in Java language.

Gnutella is a networking protocol, which defines a manner in which computers can speak directly to one another in a completely decentralized fashion [9]. This group membership and search protocol is mainly used for file sharing. Gnutella users connect to each other, and not to a central server. Communication and information searching within the Gnutella network is done through message broadcasting. Users can use Gnutella client (or Gnutella protocol-compatible program) like BearShare [10], LimeWire [11] and Morpheus [12] to participate in the Gnutella network. Gnutella is decentralized - this means there is no central server that the network relies on for its existence. If one server goes down, Gnutella keeps working. Besides, Gnutella also includes other P2P file sharing application features, such as the ability to operate in a dynamic environment, performance and scalability [13].

The conventional client-server model has become a bottleneck in handling the Internet information. P2P computing seems to be the new technology that is able to free the current limitation. Seeing the potential of P2P, Sun Microsystems Inc. started research on P2P technology, The Project JXTA [14]. 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 [15]. 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.

3. GENERAL FRAMEWORK OF CPM/P2P

There are three basic entities in CPM, namely resource provider, resource consumer and the market. These entities communicate with one another via a set of utilities and services such as resource discovery, price negotiation, resource allocation, job deployment, accounting and so on with the market as the mediator. Figure 1 shows the interaction among the different CPM components.

CPM agents will 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 negotiation, job execution and issuing of bills.

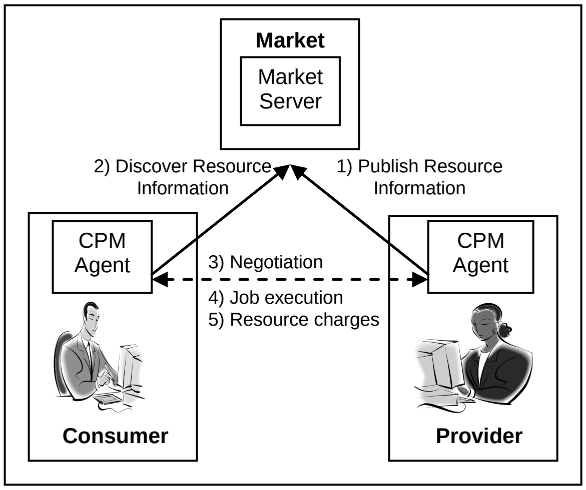

Figure 1: Interaction among CPM entities

The market server contains an administrator. It creates a market and maintains a market repository.

There are four layers in the CPM, as shown in figure 2. Layer 1 connects geographically distributed compute devices through JXTA network across the Internet. Layer 2 is the middleware comprising JXTA protocols and security. It provides a set of APIs that allows developer to implement the JXTA protocol. The Core Engine layer contains generic components, market, trader, job management and accounting modules. Generic components provide the fundamental support to other CPM services. The market module provides the market information and communication facilities, whereas the trader module caters for provider ranking, matching and negotiation issues. After negotiation is over, the job management module sends and executes the consumer's job on provider's compute device. After the job is completed, the accounting service will calculate the charges and bill the consumer. These modules are packaged into market server and CPM Agent applications.

Layer 4 Applications	Market Server & CPM Agent
Layer 3 CPM core engine	Market, Trader, Accounting, Job Generic Components
Layer 2 P2P middleware	JXTA Protocols & Security
Layer 1 P2P fabric	Internet

Figure 2: CPM/P2P Framework

4. CPM CORE ENGINE

An essential part of the CPM framework is the core engine. It consists of several back-end processing engines that run the CPM.

CPM applications
CPM functional modules
CPM generic components
JXTA & Java APIs

Figure 3: Generic Components in CPM/P2P Framework

The generic components provides fundamental support for resource discovery using the JXTA protocol, as shown in figure 3. It provides design transparency to other CPM modules. For example, trader module can call resource discovery component to retrieve the list of available resources. Two generic components have been well defined currently, namely membership control and resource discovery.

The membership control component caters for the regulation of provider and consumer to form a market. Provider who wishes to contribute a resource will use the market discovery service to discover available market in the JXTA network and join the targeted market. In order to use the resources contributed in the market, consumer needs to join the market as well.

The resource discovery component provides the facilities to publish resources (for provider) and discover available resources (for consumer) in the market. It is built on top of the JXTA Discovery Protocol. The existence of a peer, group of peers and services provided is made known via the use of advertisement. Advertisement here refers to an XML-format document that contains necessary information for discovery mechanism. For example, market administrator will create an advertisement that consists of provider name, its IP address, communication endpoint and a description of the provider (such as information on hardware, resource usage charges, available period, etc.), and publish it to the market. Consumers who had joined the market can discover the advertisement and send contract to the provider for trading negotiation.

The market is the place to bring resource providers and consumers together. Basically, it has three functions:

- maintaining information repository of provider resources,
- providing mechanisms for updating the information,
- interacting with others markets.

The trader works on behalf of the consumer and provider to carry out resource trading process. Provider uses its trader to publish resource information and wait for consumer approach. Consumer uses its trader to obtain appropriate resources and search through the market repository. When a suitable resource is selected, consumer agent will send a contract to the provider agent and request for the resource. The provider agent will negotiate with consumer agent and make the decision (whether to receive or reject a contract).

On the provider side, the decision-making is simple. As long as a resource is available, a contract can be accepted.

As for the consumer, it can specify some optimization mode for a particular project, based on cost, speed, deadline, budget, etc.

The job management module manages a complete life cycle of a particular job, which can be divided into three stages namely job deployment, remote job execution and job completion. A job logger will record down the details of the resource that has been consumed, consumer details, job ID, execution time and other related information. This information will be used by accounting module to calculating resource charges.

Accounting module handles the fees and payments for the resource trading and keeps track of the resources that have been consumed using a resource meter. We have identified three processes involved in this module, namely *keep, compute* and *acknowledge*. *Keep process* refers to storing resource charges for every resource that has been discovered. After a job is successfully executed in the provider machine, the *compute process* will calculate the charges for that particular trading based on the usage recorded in resource meter. Lastly, the *acknowledge process* is called to send a bill to the consumer.

5. OUR PROTOTYPE

We have developed some of the core components to provide basic CPM services. Among others, the completed components are market creation, market discovery, market repository, joining market, resource discovery, resource publishing, job transfer, remote job execution, job management and simple accounting. A prototype (in the form of a portal) has also been designed which incorporates all the above mentioned services. In fact, the first version of our generic components has been uploaded to the JXTA CVS server at the end of last year.

Figure 4 shows a scenario of a complete cycle of resource trading in CPM/P2P that we have developed. 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.

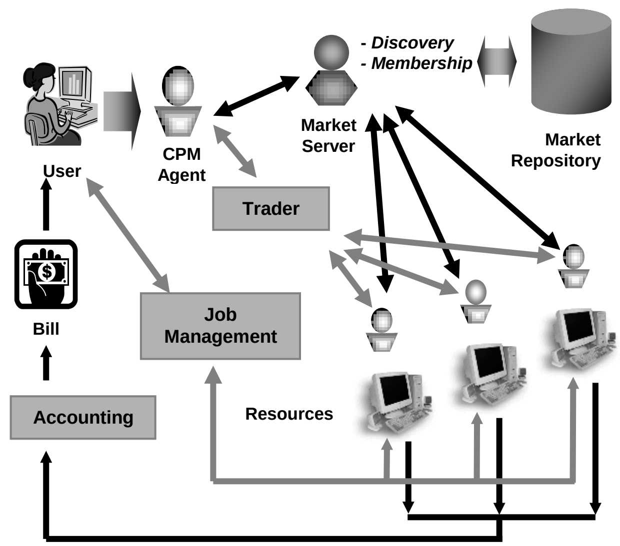

Figure 4: A complete cycle of resource trading in CPM/P2P architecture

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 consumer trader will rank according to consumer defined optimization criteria. Once a provider is selected, a contract will be generated and sent to provider trader. On the provider side, its agent will negotiate with the consumer agent. This is followed by the transfer of the job and executing it on the provider. Once completed, the CPM agent will calculate the charges and bill the consumer accordingly. This completes a cycle of resource renting in CPM/P2P.

Currently, we are working on enhancing the job management module, developing job scheduler and market repository. We are also trying to conduct some experiments using our basic CPM components. In these experiments, we will try various combinations of providers and consumers in selecting and running jobs. Among the evaluation that can be performed is job execution performance (single machine vs. multi machine), resource discovery time and trader efficiency in matching consumer criteria with appropriate provider resources.

6. CONCLUSION AND FUTURE WORK

With the vast need for computing resources, the ability to find and match requirements for a user is very important. It has to be done according to the criteria defined by both consumers as well as providers in order to obtain optimum benefit for all parties concerned. The success story of various resource sharing systems such as Morpheus, Limewire and Bearshare is a great motivation towards the development of the resource sharing system using P2P technology. In this paper, we have proposed a market-oriented resource sharing system using JXTA technology. The CPM/P2P is a market-based resource trading system that is built using P2P technology, making use of distributed processing on idle computers. In the future, we're planning to incorporate various market models (currently only commodity model is used) and also look into the design of super peer scheduler that enables job submission to heterogeneous markets.

6. ACKNOWLEDGEMENT

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

REFERENCES

[1] Rajkumar Buyya and Sudharshan Vazhjudai, Compute Power Market: Towards a Market-Oriented Grid, *The First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, Brisbane, Australia, May 15-18, 2001.

[2] SETI@home - http://setiathome.ssl.berkeley.edu

[3] 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.

[4] 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.

[5] Rajkumar Buyya, Jonathan Giddy and David Abramson, A Case for Economy Grid Architecture for Service-Oriented Grid Computing, *10th IEEE International Heterogeneous Computing Workshop (HCW 2001)* (in conjunction with IPDPS 2001), San Francisco, California, USA, April 2001.

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

[7] Li Gong, Project JXTA: A Technology Overview, Project JXTA Documentation, April 25, 2001 - http://www.sun.com/software/jxta/jxta-doc.html

[8] 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.

[9] Gnutella – http://www.gnutella.co.uk/

[10] BearShare – http://www.bearshare.com/

[11] Limewire – http://www.limewire.com/

[12] Morpheus - http://www.morpheus-os.com/

[13] Matei Ripeanu, Ian Foster and Adriana Iamnitchi, Mapping the Gnutella Network: Properties of Large-Scale Peer-to-Peer Systems and Implications for System Design, *IEEE Internet Computing Journal, 6*(1), 2002.

[14] JXTA - http://www.jxta.org

[15] Sun Microsystems Inc, Project JXTA: An Open, Innovative Collaboration, Project JXTA Documentation, April 25, 2001 - http://www.sun.com/software/jxta/jxta-doc.html