A Survey of Market-Based Approaches to Distributed Computing

Shashank Shetty, Pradeep Padala and Michael P Frank
Computer & Information Science & Engineering
University of Florida
Gainesville, Florida 32611–6120
Email: {sks,ppadala,mpf}@cise.ufl.edu

Abstract— **Distributed and grid computing has become the choice of computing in high performance applications. Various projects have developed software infrastructure to harness the enormous power of distributed resources. Applying market approaches to computing, though not new, is an active field of research. Various key issues like QoS of matching, protocols for negotiation and service level agreements are not fully understood. Academic projects like Nimrod/G and commerical projects like GridMP have made good contributions in answering these key questions. In this survey, we explore the distributed computing paradigm and issues. We present an overview of methods used by various projects and how they do resource management. We conclude with a list of open research problems in this field.**

I. Introduction

Distributed computing strives to harness the power of resources distributed in a network. There are millions of computers connected to Internet that are usually idle and can be part of collective computing. In today's world, distributed computing has attained new heights with complex projects like SETI@home[1]. But, there are various challenges for achieving the noble goal of a global computer. Recently, there has been many advances in solving these problems using the grid. Grid[2], as it is known, enables the sharing, selection, and aggregation of a wide variety of geographically distributed resources including supercomputers, storage systems, data sources and specialized devices owned by different organizations administered with different policies.

Grid computing has become the choice of distributed computing especially in academic disciplines. To harness the power of the grid, there are some key issues to be solved in resource management. In the last few years, a number of exciting projects like Globus[3], Legion[4] and UNICORE[5], developed the software infrastructure and protocols needed for grid computing. Various distributed computing issues have been solved using these tools and libraries. But, we do not yet have a scalable, high-performance, self-serving resource management architecture.

Surprisingly, we can apply the resource management strategies used in economic markets in distributed computing. In this paper, we provide an overview of market approaches and how they are applied to distributed computing. First, we provide a background of market approaches including review of some existing distributed computing projects. Then, we explain various issues involved in resource management. Next, we present a detailed survey of various projects using market based approaches for resource management. We conclude by summarizing our findings and open research issues in this active field.

II. Background Research

Economic systems in human society can be broadly classified into two models: Central Planning Model and Free Market Model. In Central Planning Model, a single institution has total authority and decides what to produce, how to produce and to whom. In a free market model, producers and consumers make these decisions suiting their benefits. Central Planning Model is aimed at the overall welfare of all individuals at the cost of some individuals' freedom. Such an economy makes free trade tougher. Thus, the free market model wins in this case even though it has its share of disadvantages.

Auctioning plays a major role in a free market model. Auctioning has history dating back to 500B.C., when men used to bid for women, whom they wanted to marry on the streets of Babylon. Auctioning has evolved constantly over years. Christie's and Sotheby's are few of the popular auction houses currently, where people trade things. Change in technology also brings change in our economic environment. Advent of computers and Internet brings new dimension to auctioning. Online auctioning services like *e-bay* and *Yahoo Auctions* have major impact on how we buy various commodities.

Now, researchers are trying to apply the same techniques to distributed computing. It all started back in 1968 when Sutherland demonstrated how auction methods were used to allocate time to users of the PDP-1 Computer in Aiken Computation Laboratory at Harvard University[7]. In this system, hours of the day were divided into time slots. Users were assigned different amounts of currency based on the importance of their projects. Users bid for a particular time slot and a user who submits the highest bid gets the privilege of using the computer for that time-slot. This is one of the first applications of economics to problems of computer resource allocation.

But, Sutherland's work failed to address the problem in distributed system. Drexler and Miller approached this problem in paper[8], which describes auction mechanisms for allocating distributed resources. This paper made two important contributions to computational market. First, market based mechanisms to allocate distributed resources is provided and second, the problem of stability of pricing in computational market is addressed.

The advent of personal computers and Internet technology caused computational economy to attain new dimensions. Increase in high performance computations that require huge amount of resources like processing power, memory etc. makes it impossible for an average person to get such computations done on a normal computer. The user may want to get the computations done on other idle machines that are connected to Internet.

Usually, a personal computer sits idle for most of the time, other than doing simple tasks like browsing the net or editing a letter which only consumes a fraction of total CPU power. This has led people to think about harnessing the computing power of millions of idle computers around the world. A big step towards this direction was SETI@home project which uses computing power spread across the world to analyze radio waves from outer space in search of extraterrestrial intelligence. Anyone interested in participating in SETI@home has to download a SETI screen-saver that takes care of using the idle cpu cycles of the machine. It is amazing that the creators of the Toy Story used 117 multiprocessor Sun Sparc workstations working around the clock for more than nine months to generate the films' images. SETI@home could have accomplished this task in less than seven hours[9].

SETI@home is not a market-based approach, rather it is a simple cycle stealing technique where a user lets the SETI@home project to use the computing power of his machine free of cost. But SETI@home opened doors to the development of new market based approaches in this area by different players. With lot of research going on in this area, there are lot of private companies and academic institutions working on this. People have tried to apply different economic models existing in the human economic society. Some of the most common ones are auction model, commodity market model, contract-net model, bargaining model and bartering model[6].

Few of the popular projects that use market-based approach are Nimrod/G[12], United Devices' Grid MP Platform[14], Frontier (Parabon Computation)[16], Compute Power Market[15], Mojo Nation[17], Entropia's DCGrid[18] projects. We will give an insight into resource management in the next section before going through some of the projects in detail.

III. Resource Management

Grid Systems allow applications to assemble and use collections of resources on an as-needed basis, without regard to its physical location. Grid middleware and other software architecture that manage resources have to locate and allocate resources according to application requirements. They also have to manage other activities like authentication and process creation that are required to prepare a resource to use.

Some of the challenges involved in this task include site autonomy, heterogeneous substrate, policy extensibility, co-allocation and online control[10].

Site autonomy means resources are owned and operated by different organizations which may have different policies, security mechanisms etc. Owner of these resources has total control over the usage of resources.

Different sites may use different local resource management systems like Condor, CODINE, and LoadLeveler etc, which leads to heterogeneous substrate problem. Even if two different sites use the same local resource management system they might differ in configuration leading to difference in functionality.

Resource Management should support frequent changes of domain specific management structures without any requirement to change the code thus supporting policy extensibility. Resource management should adapt itself to new user requirements and should be capable of evolving itself to meet future demands.

Lots of applications need resources that can be satisfied only by allocation of resources simultaneously at several different sites, leading to co-allocation problem.

Resource requirements and characteristics may change during execution, thus making it necessary for having an online control because substantial negotiation may be required to adapt to application requirements.

Grid Resource Allocation Manager (GRAM)[10] API developed in the context of Globus project addresses all the five issues listed above. GRAM allows programs to be executed on remote resources regardless of any heterogeneity. It provides a standard interface for executing jobs using remote resources. Requirements are communicated using resource specification language (RSL).

Legion[11], another grid middleware developed at University of Virginia, has similar resource management architecture. Features of the architecture include global, shared namespaces, wide area data sharing, heterogeneity, security, efficient scheduling and resource management. Legion has an object-based model where each of its components is an object. These objects are independent of each other and they communicate using remote method invocation.

There are other projects that employ market based appraoches for resource management. These include Nimrod/G, OCEAN, Compute Power Market etc. For example Nimrod/G supports Commodity market and Contract-net models.

IV. Survey

Finding optimal mechanisms for distributed computing using market-based approaches is an active research problem. Various academic and commercial projects like Nimrod/G, United Devices' Grid MP Platform, Parabon Computation's Frontier, Compute Power Market, OCEAN, Mojo Nation, and Entropia's DCGrid are the result of research in this area. Some of these projects are explained in detail with a brief overview on the rest.

A. Criterion for choosing the projects

We have selected few projects based on the following criteria. First, it should be distributed computing project, second it should have mechanisms to solve various issues in resource management and third, market-based approach should be used directly or indirectly.

B. Nimrod/G

Nimrod/G[12], a grid enabled resource management and scheduling system that uses Globus middleware services for resource discovery and dispatching jobs over grid. Nimrod/G is modular, extensible, portable, provides interoperability of independently developed components and provides a simple declarative parametric modeling language for expressing a parametric experiment.

1) System Architecture: Nimrod/G has 5 key components as shown in the figure 1.

- *Client or User System*: It provides a user-interface that allows user to vary parameters and monitor the status of the submitted jobs. It allows running multiple instances of the same client at different locations
- *Parametric Engine*: Heart of the system that manages the whole experiment. As a persistent job control agent, it is responsible for creation of jobs, maintenance of job status, interacting with clients and scheduling dispatcher. Declarative parametric modeling language is used to give input to the experiment. Experiment is recorded in a persistent storage, so that the experiment can be restarted in case of failure.
- *Scheduler*: It is responsible for resource discovery, resource selection and job assignment. Resource discovery algorithm interacts with grid-information service discovery (MDS) [13]and identifies the list of authorized systems. Resource selection algorithm selects those resources that meet the deadline and minimize the cost of computation.
- *Dispatcher*: It initiates execution of a task on the selected resource. This is done by starting a remote component named job wrapper.
- *Job-Wrapper*: It is responsible for starting execution of a task on the selected resource and sending the results back to parametric engine.

2) Selection of computational resources: Selection of computational resources is handled in two ways as shown below:

- The work is completed within a given cost and deadline.
- User is allowed to negotiate for resource in the grid. The system can employ resource reservation or trading technique to identify suitable resources. The user is then allowed to renegotiate with a different deadline or cost.

C. United Devices - Grid MP Platform

Grid MP platform [14] aggregates computer resources across clusters, servers, workstations and PCs to enhance the performance of compute-intensive applications. Main features are

- supports wide variety of applications from message passing interface jobs to data parallel jobs that run on clusters, servers and desktops.
- enables multi-step parallel or pipelined applications to be executed as single job on the grid.
- includes Linux Binary Emulator that executes Linux applications on Windows platform without any modifications.

1) Grid MP Interfaces: It includes tools and interfaces to create and manage various functions in the grid.

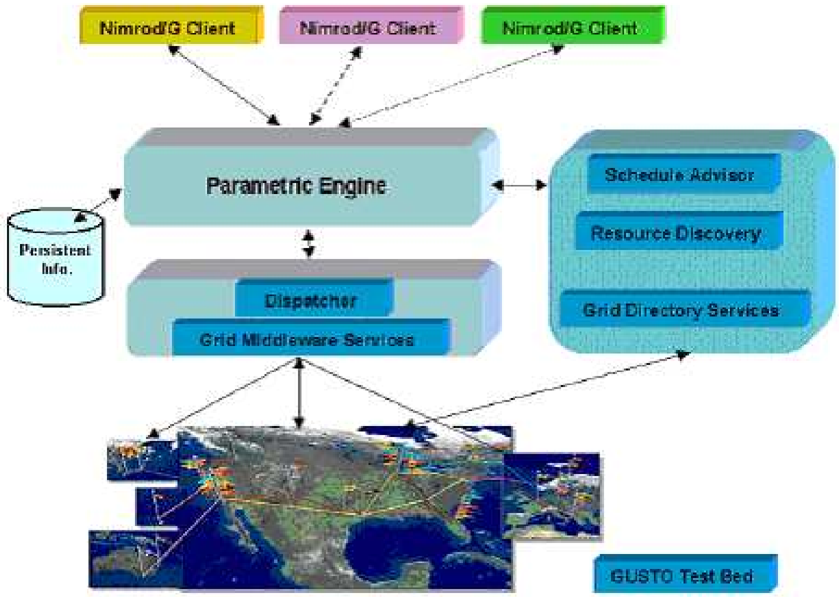

Fig. 1. Nimrod/G architecture

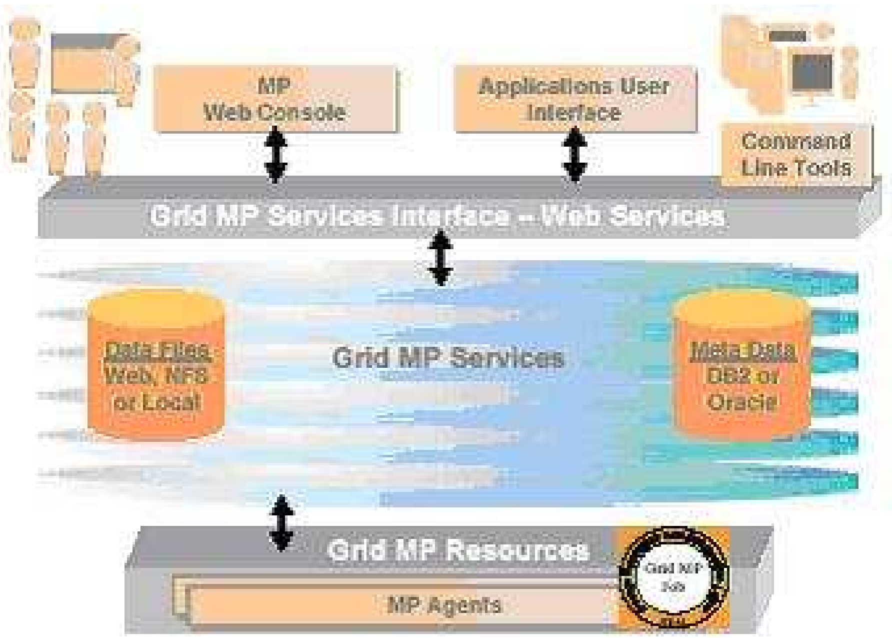

Fig. 2. Grid MP 4.0 architecture

- *MP Web Console*: User interface for administrators to manage users, devices and jobs. It also allows users to monitor the progress of their jobs.
- *MP Grid Service Interface (MGSI)*: It is a XML/SOAP based web services interface that enables applications to be grid enabled.
- *MPSUB*: It provides a command line interface to submit applications to resources.
- *MPIRUN*: It enables parallel jobs to run on the resources that are capable of running them.
- *MP Software Development Kit (SDK)*: Development kit for programmers and users to develop custom grid applications.

2) Grid MP System Components: It is a collection of components that perform specific functions to run and manage grid applications.

3) Grid MP Agent: It is the software that runs on all the participating machines. It is responsible for identifying and authenticating itself to the MP Services, downloading jobs and returning results after executing the jobs. It takes care of securing the job application from other applications executing on the same machine. It also executes Linux libraries on Windows without any modifications to the code.

4) Grid MP Services: It includes the following core services that operate on the grid:

- *Realm Service*: Registers, authenticates and licenses devices on the grid. MP Agents authenticate themselves with Realm Services before starting execution.
- *Monitoring Service*: Uses polling to periodically process state information of all active devices on the grid.
- *File Service*: It is responsible for sending and receiving files from MP Agent. It uses an underlying file system like NFS or AFS. It also provides garbage collection for those files that are no longer needed.
- *Dispatch Service*: As the name says, it dispatches jobs to those devices that are idle connect to it. It uses a centralized scheduling engine that selects a job based on device's capability, job priority, device availability, usage preferences and other defined constraints. It also reschedules failed jobs.
- *Metadata Repository*: It is a database that stores information on devices, device configurations, user accounts, user roles, applications, jobs and data. Currently, it supports DB2 and Oracle.

5) Grid MP Management Functions: Grid MP incorporates a set of management functions for administrators and users to apply on the grid.

- *Application Management*: It enables administrators to register applications and manage them.
- *Device Management*: It enables administrators to separate devices based on their capability or geographic location.
- *Data Management*: It enables the sharing and managing application data across the system.
- *User Management*: It allows administrators to manage users on the system.
- *Workload Management*: It allows both users and administrators to manage the jobs running on the grid.

Grid MP has been most successful among organizations that want to harvest the spare cycles of their own organization.

D. Compute Power Market

Compute Power Market (CPM)[15] is a market based resource management and job scheduling system for grid computing. CPM is comprised of Markets, Resource Consumers and Resource Providers. It supports three major economy models - Commodity market, Contract-net/tendering and Auction models.

1) System Architecture: Three major components of CPM are Market, Market Resource Agent, and Market Resource Broker as shown in the figure 3.

Market is the point of interaction for both consumers and producers. Consumers and producers have to register with a market to buy or sell computational power. Market provides the following services.

- Repository of information on providers
- Agents for consumers and providers
- Mechanisms for updating the information
- Interaction with other markets

Market Resource Index is a database that has information on providers and consumers within that markets' domain. It also stores information on other markets like addresses of other markets, capabilities etc.

Market Control Unit is the Central processing unit of a market. It is responsible for channelizing and regulating requests, monitoring the market behavior and synchronizing with other markets.

Provider domain is associated with resource providers in the CPM. It is made up of Market Resource Download Unit and Update Unit. Market Resource Download Unit keeps track of download of market resource agents by providers. On completion of download, an entry is made in Market Resource Index for that provider. Update unit is responsible for updating the provider information, whenever the provider specification changes. Consumer

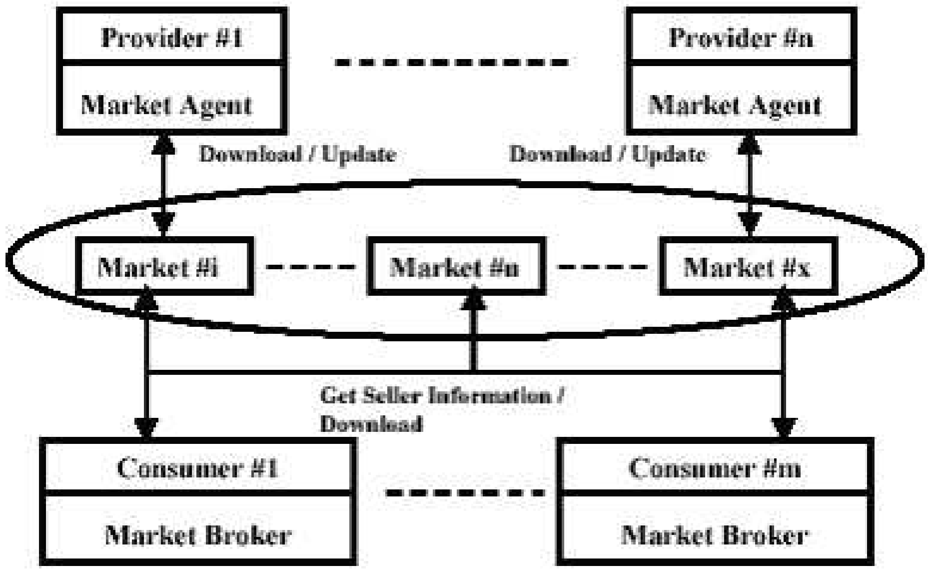

Fig. 3. Compute Power Market

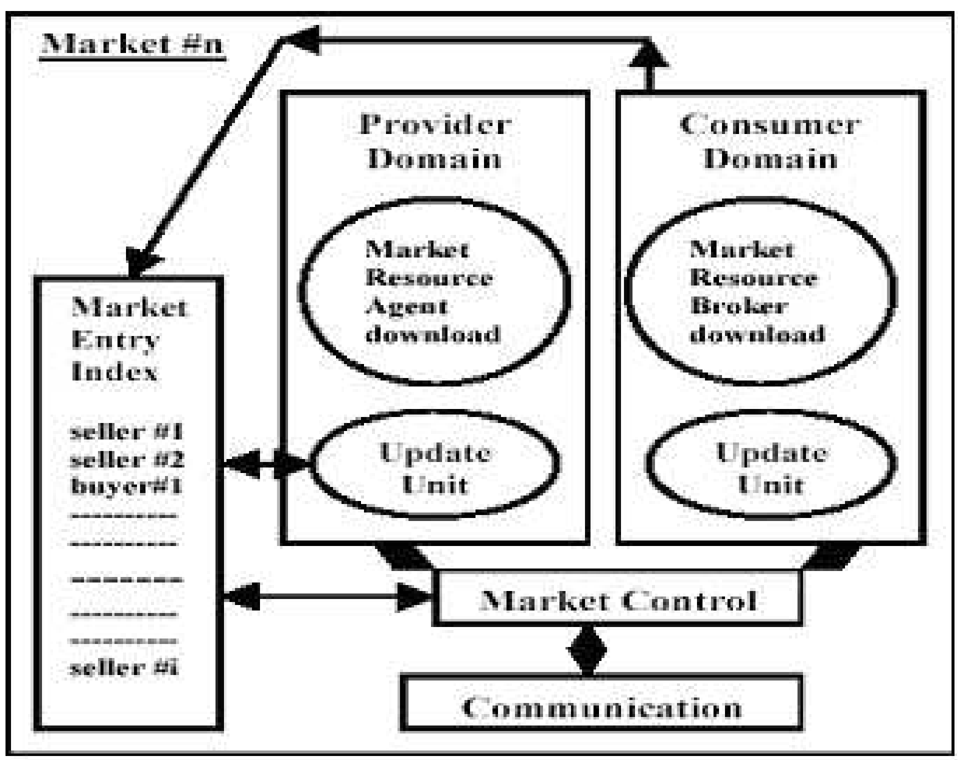

Fig. 4. Market architecture

domain is similar to provider domain but deals with resource consumers.

Resource providers download Market Resource Agent(MRA) from the market. MRA is responsible for updating the market with the latest information on providers' resources and accepting, deploying and launching the job. MRA works on a push-pull mechanism with the pull unit extracting information like available memory, number of processes etc. whereas the push unit pushes this data into the market through communication unit.

Resource Consumers download Market Resource Broker (MRB), which is responsible for locating the appropriate provider based on the information furnished by the market. It acts as a middleman between user applications and CPM resources. It includes the following components:

- Job Control Agent
- Market Explorer
- Resource Trader
- Scheduler
- Deployment Agent

Job Control agent is responsible for shepherding a job through the system by interacting with other components of the broker. Economy computations in the CPM grid is managed by schedule advisor, trade manger and trade server. The schedule advisor uses grid explorer for resource discovery, trade server for negotiating costs and scheduling algorithms for identifying mappings between jobs and resources that meet deadline and minimize the cost of computation.

E. Frontier: The Premier Internet Computing Platform

Parabon Computation is the first company to release a commercial distributed computing platform - Frontier [16]. Frontier delivers the spare processing power of millions of computer connected to Internet. It defines three categories - Clients (one who needs processing power), Providers (one who provides unutilized power of his/her computer) and Developers (one who writes application on Frontier).

Frontier is comprised of three components:

- *Pioneer Compute Engine*: It is a desktop application that manages the unused processing power of provider's computer. It is responsible for downloading tasks on the provider's computer from frontier Server and uploading the results of the tasks back to the Frontier server. Pioneer starts processing when the provider's computer becomes idle and uploads the results into the Frontier server when the provider connects to the Internet.
- *Frontier Server*: It is a centralized server responsible for scheduling, dispatching and executing jobs. It monitors all the tasks that are running and also takes care of switching jobs to another machine in case of system failure.
- *Frontier Software Development Kit*: Software tool that enables clients to write, monitor and control the execution of their jobs.

F. Miscellaneous Projects

Mojo Nation is a peer-to-peer file sharing system like Napster, but the users cannot simply share files free of cost instead users should pay the providers for whatever they download. It employs bartering model and builds a content sharing community within a network.

Entropia's DCGrid is aimed at organizations that have computational bottlenecks in critical path processing. DCGrid aims to utilize untapped CPU cycles of the Windows based Desktops in the existing enterprise network.

DCGrid is comprised of two key modules - DCGrid Scheduler and DCGrid Manager. DC Grid Scheduler schedules, deploys and manages execution of jobs on the network, whereas DCGrid Manager provides a web based, centralized management console for managing the resources. It takes care of splitting, submitting and gathering of job results.

G. Conclusion

We have surveyed various projects supporting market-oriented distributed computing. There has been significant work in various issues in market approaches. But, there is a lot of potential for more research. Finding optimal protocols for resource matching and benchmarking various protocols is still an open task. Research needs to be done for evolution protocols of matching networks. How much overhead do these protocols impose on networks? How many messages are distributed in the network? How to reduce the overhead? - all these questions are not fully answered by any of the projects.

REFERENCES

[1] E. Korpela, D. Werthimer, D. Anderson, J. Cobb, and M. Lebofsky, "Seti@home: Massively distributed computing for seti," *IEEE Computing in Sciecne Engineering(CiSE) Magazine*. [Online]. Available: http://www.computer.org/cise/articles/seti.htm

[2] I. Foster and C. Kesselman, Eds., *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann Publishers, 1999.

[3] I. Foster and C. Kesselman, "Globus: A metacomputing infrastructure toolkit," *The International Journal of Supercomputer Applications and High Performance Computing*, vol. 11, no. 2, pp. 115–128, 1997.

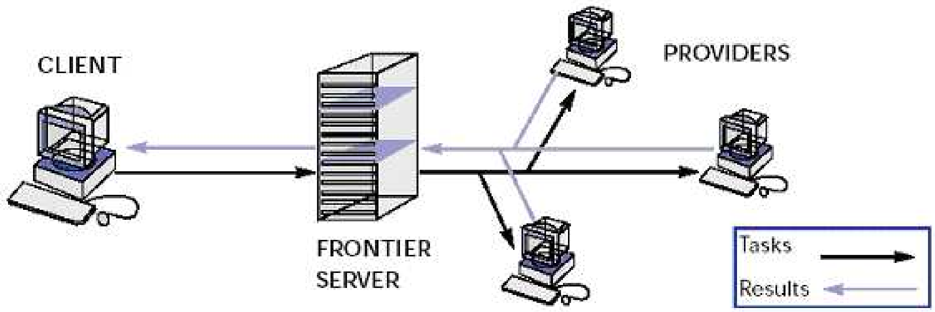

Fig. 5. Frontier

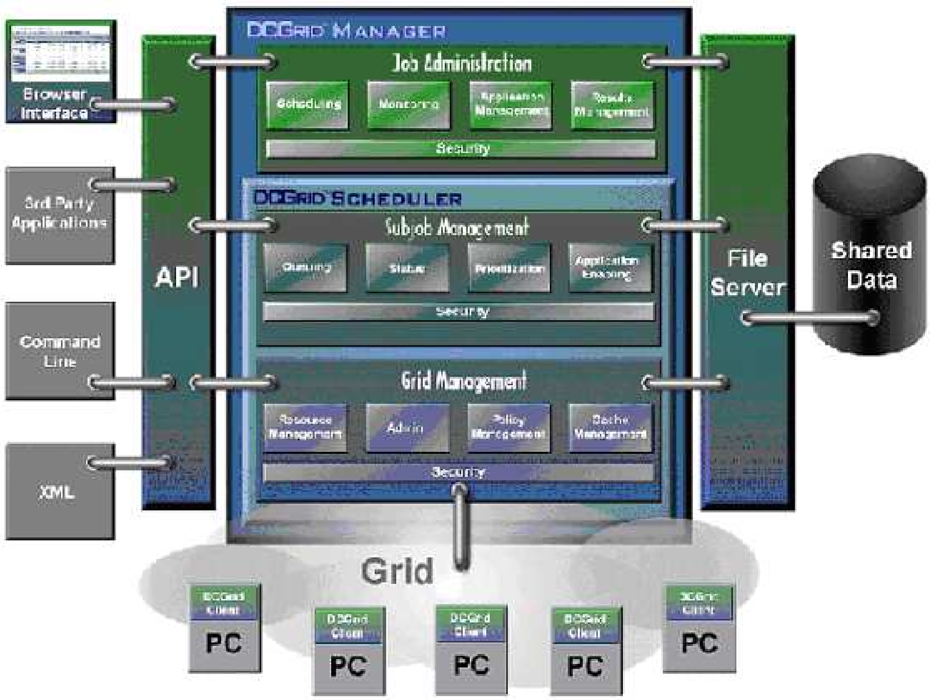

Fig. 6. DCGrid Architecture

[4] A. S. Grimshaw, W. A. Wulf, and the Legion team, "The legion vision of a worldwide virtual computer," *Communications of the ACM*, vol. 40, no. 1, pp. 39–45, Jan. 1997.

[5] V. Huber, "UNICORE: A Grid computing environment for distributed and parallel computing," *Lecture Notes in Computer Science*, vol. 2127, pp. 258–266, 2001.

[6] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic models for resource management and scheduling in grid computing," *Special Issue on Grid Computing Environments, The Journal of Concurrency and Computation: Practice and Experience(CCPE)*, vol. 14, no. 13, pp. 1507–1542, Nov. 2002.

[7] I. E. Sutherland, "A futures market in computer time," *Communications of the ACM*, vol. 11, no. 6, pp. 449–451, 1968.

[8] M. S. Miller and K. E. Drexler, "Markets and computation: Agoric open systems," in *The Ecology of Computation*, ser. Studies in Computer Science and Artificial Intelligence, B. A. Huberman, Ed. Elsevier Science Publishers, 1988, pp. 133–175.

[9] D. Tapscott, "Pc synergy: Private business considers ways to emulate seti-like cpu sharing," *Intelligent Enterprise Magazine*, June 2001. [Online]. Available: http://www.intelligententerprise.com/010613/change1_1.shtml/ent_dev/

[10] 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*. Springer-Verlag LNCS 1459, 1998, pp. 62–82.

[11] A. S. Grimshaw, M. J. Lewis, A. J. Ferrari, and J. F. Karpovich, "Architectural support for extensibility and autonomy in wide-area distributed object systems," Department of Computer Science, University of Virginia, Tech. Rep. CS-98-12, June 3 1998. [Online]. Available: http://www.cs.virginia.edu/~techrep/CS-98-12.ps.Z

[12] R. Buyya, D. Abramson, and J. Giddy, "Nimrod/G: An architecture of a resource management and scheduling system in a global computational grid," *HPC Asia 2000, IEEE Press*, Sept. 22 2000. [Online]. Available: http://arXiv.org/abs/cs/0009021

[13] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *Proceedings of the* 10^{th} *Symposium on High Performance Distributed Computing*, 2001. [Online]. Available:

http://citeseer.nj.nec.com/czajkowski01grid.html
[14] U. Devices, "Grid mp platform: Architecture overview." [Online]. Available: http://www.ud.com/solutions/rescenter/mp_architecture.pdf
[15] R. Buyya and S. Vazhkudai, "Compute power market: Towards a market-oriented grid," *The First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001)*, May 2001.
[16] "The frontier." [Online]. Available: http://www.parabon.com
[17] "Hivecache: Distributed enterprise online backups." [Online]. Available: http://www.mojonation.net/
[18] "The entropia approach to distributed computing." [Online]. Available: http://www.entropia.com/reg.asp?dl=eapproach