Economy-based computing with WebCom

Adarsh Patil, David A. Power and John P. Morrison
Department of Computer Science,
University College Cork,
Ireland.
{adarsh,d.power,j.morrison}@cs.ucc.ie

Abstract— Grid environments consist of the volatile integration of discrete heterogeneous resources. The notion of the Grid is to unite different users and organisations and pool their resources into one large computing platform where they can harness, inter-operate, collaborate and interact. If the Grid Community is to achieve this objective, then participants (Users and Organisations) need to be willing to donate or share their resources and permit other participants to use their resources. Resources do not have to be shared at all times, since it may result in users not having access to their own resource. The idea of reward-based computing was developed to address the sharing problem in a pragmatic manner. Participants are offered a reward to donate their resources to the Grid. A reward may include monetary recompense or a pro rata share of available resources when constrained. This latter point may imply a quality of service, which in turn may require some globally agreed reservation mechanism. This paper presents a platform for economy-based computing using the WebCom Grid middleware. Using this middleware, participants can configure their resources at times and priority levels to suit their local usage policy. The WebCom system accounts for processing done on individual participants' resources and rewards them accordingly.

Keywords: WebCom, Economy-based computing, WebCom Grid Bank Reward, Condensed Graph, Distributor, Accounting, Grid Point

I. Introduction

A Grid environment is built upon the numerous hardware and software resources and services available to the masses. The key role of any Grid middleware in this situation is to harvest a large number of geographically-dispersed heterogeneous resources, and present them to the application developer or an end user as a single system. Many different middlewares, such as Globus [14], Legion [15] and GridSolve [6] offer solutions to these

problems and today Grids using these technologies can be easily constructed. Complications can arise with Grid environments using these middlewares, where participants are expected to donate their resources and allow others to use them. This is not always practical, as some participants might not be willing to donate their resources to others and the resource provider does not encourage free use of their resources. Many reasons for this are possible:

- Participants may wish to recover some or all of their investment in a particular resource/set of resources. Therefore they may be unwilling to permit free access to these resources.
- Participants may not be guaranteed priority access to their own resources if they are shared. This might impact on a decision to engage in resource sharing, especially if the participant expects to utilize its own resources frequently.
- Participants may have concerns about the integrity of their resources if they allow others to use them.
- Credits might be lucrative however putting in place the essential software and hardware infrastructure might be complicated. Participants opt for a reward-based approach only when there is sufficient gain in doing so.

Reward-based computing was conceived to address these problems. Participants are offered economic rewards for donating their resources to the Grid. The word 'reward' signifies feasible contributions are paid for processing/sharing done on resources. Currently, rewards can be the exchange of currency, Grid points or tokens.

This paper presents a platform for reward-based computing using the WebCom Grid middleware. Using this software, participants can donate their resources to the Grid. Participants can configure their

resource availability and priority at times convenient to them.

The rest of this paper is organized as follows: Economic Models are discussed in Section II. In Section III, reasons for choosing a reward-based computing model are sketched. In Section IV, the use of WebCom as a reward-based computing platform and how WebCom can schedule the jobs on resources within their prescribed time limit is discussed. In Section V, some sample executions and outcomes are introduced. Finally, in Section VI, conclusions and some future work are discussed.

II. Economic Models on the Grid

The arrival of Grid computing exposed many unsolved problems in the domain of scheduling and job management systems. Recently, investigation has turned towards economy-based scheduling and job management. The main difficulty in constructing a computational Grid based on economic resource allocation is measuring the cost of resource usage. Obviously, valuable resources dispersed across different organisations are highly dynamic, with varying hardware and software. Finding a discrete way of charging for such resource usage may be difficult. Previous work that addresses some economy based models includes [8], [39] and [40]. These systems measure the price of utilising specific resources by transforming the value of using different type of resources into a common currency [17]. Some of these economic models are still under investigation and have not found widespread use in today's Grid Environments.

As mentioned already, the resources that make up a Grid can be dispersed across different administrative domains. Some of these domains share their resources with the Grid for free and others might only allocate their resources if they receive payment or reward. It can be argued that to make Grid Computing viable in the real world, incentive-based computing is very important: If everyone is a consumer, then the Grid cannot function. Grid participants therefore have to be enticed to share their resources. In the case of some Peer-to-Peer (P2P) networks, users have to maintain a 1:1 download-to-upload ratio in order to be allowed to participate in that network. There are other P2P networks that are not based on file sharing, but on the cooperation model of resource usage (for example, the Jabbar Instant messenger). All nodes in this network offer services for each other, be it searching, uploading or assisting message delivery.

A lot of research has gone into economy based computing with respect to P2P systems. Some research papers ([4], [41]) have categorized economy models into two groups based on their mechanisms and characteristics (weak and strong):

- **Weak Models:** In this type of model there is no actual representation of cost as such. Offering a service is not mandatory in this model. Some weak models include the classic model, the Higher Good/Charity model, and the common interest model:
  - **Classic model:** In this model participants are not required to offer or share the services. Examples: Napster [37], Gnutella [5], Freenet [10].
  - **Higher Good/Charity:** In this model participants only contribute services but do not use them. Examples: Seti@home [7], Folding@home [24], Aids@home [32]
  - **Common Interest:** This is similar to the previous model. In addition, the service provider may utilize the service. Example: Collaborative work on cryptographic key crack [33].
- **Strong Models:** In this type of model, service offering is mandatory for the participants' survival and existence within the system. Strong models typically offer Micro Currency, External Billing and Force Sharing:
  - **Micro Currency:** A virtual currency is established between all nodes, with services utilized between the nodes being paid for with this currency. A centralized authority is needed to issue, deploy and authenticate the currency to avoid inflation. Example: Mojo Nation [1]
  - **External Billing:** In this model, billing of the service is done by an authority outside the network. Example: Sun N1 Grid Engine 6 [3].
  - **Force Sharing:** This model implements a 'Give and Take policy': All the participants in this model have to share a service in order to consume a service. They should maintain an equal service-sharing and consuming ratio. Examples:
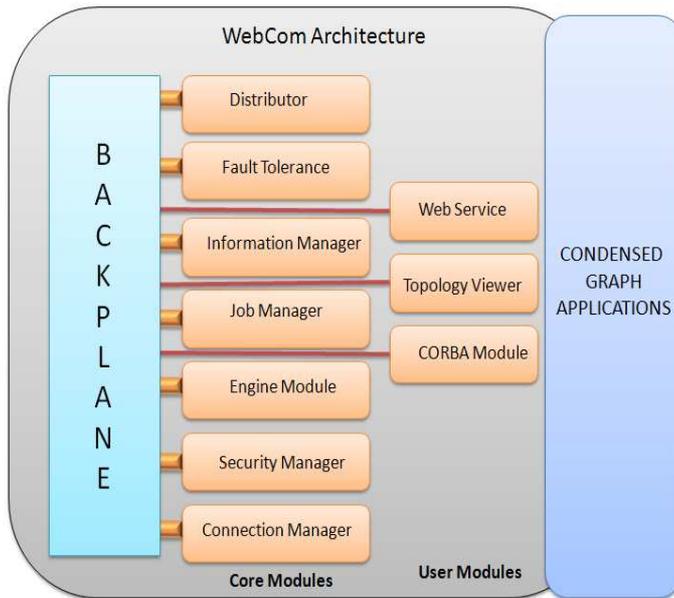
eDonkey [18], Overnet [23].

Fig. 1. WebCom architecture

III. WHY DO WE NEED ECONOMIC MODELS IN A GRID ENVIRONMENT?

The Grid is a heterogeneous parallel and distributed system spanning, and owned by, multiple administrative domains across the globe. Grid environments consist of collections of hardware resources ranging from desktops and laptops to supercomputers and mainframes, and a variety of software systems running on these. Typically such systems are not owned by a single domain or a single user: They have a distributed ownership. In addition, there may be users who do not own any of the software or hardware resources, but merely want to use the services offered by others. These services may include computational services, data management services, storage services and security services. Therefore, to provide a proper balance between supply offered by the providers and demand required by consumers, a suitable economic model needs to be implemented. Such a model must take account of each stakeholder's interests. Some reasons for providing an economic model are:

- To make a profit or achieve maximal return on investment in resources.
- To differentiate Providers, Consumers and Brokers. This classification gives a clear notion about the owners of the Grid resources, the end

users (who are utilising the resources) and the brokers (who perform matchmaking to choose which resource is utilised by the end user).
- In order to keep the Grid alive and growing, services offered by the Grid should be charged. These service may be open source software executing on dedicated hardware. The infrastructure has to be maintained, serviced and administered.

A. Opportunities for Profit

At present, water, gas and electricity services have become commodity requirements. Grid is a new candidate for the commodity list serving uniform access to computational resources, data storage, security and other services, as it involves constrained access to a distributed collection of shared resources. These resources may be either affordable or unaffordable. Therefore, to provide monetary benefits, proper tools must be developed. These tools must result in efficient use of the available resources. The following areas have been identified as potential methods of profit making:

- Development of application-dependent Grid middleware for sale to respective application dependent domains, enterprises and industries. Example: Platform Computing [2].
- Hosting of a Grid system/environment collating abundant sets of resources and invitation to consumers to use them and make money by applying business models. Examples: IBM on-Demand Computing [19], Sun N1 Computing [36].
- Development of a Grid middleware/client/agent and deployment within the academic resources. These can be used to test business models and analyze different results. Results from the analysis can be published in International Journals and Conferences. Example: Academic Research Institutes involved in Grid Computing Research through patenting their technology and models.

B. Economic Models from the Past to the Present

Traditionally, the availability of goods has been constrained by the locality of the supply of the goods. Certain goods may not be available locally, and hence have to be imported. The economic

models offered in the procurement of certain goods can be identified as:

1) **Monopoly** [35]: One seller and many consumers model.
2) **Monopsony** [25]: One consumer and many sellers model.
3) **Oligopoly** [38]: Small number of sellers and many consumers model.
4) **Commodity Market model** [27]: Open markets and online Internet shopping.
5) **Posted Price Model** [9]: Similar to Commodity Market Model, but includes announcement of discounts and special offers.
6) **Barter model** [16]: Exchange of goods of interest.
7) **Bid Model** [26]: eBay for example.
8) **Tender/Contract Model**
9) **Proportional Models**
10) **Auction Models** [26]
    - Dutch Auction. [21]
    - English Auction. [11]
    - Double Auction [34]
    - First Price Sealed Bid Auction [30] (Vickery method)
    - Second Price Sealed Bid Auction. [30]

C. Reasons to opt for Reward-Based Computing

The models illustrated in Section III-B and described in [8] are generally not suitable for reward-based computing because:

- These are traditional economic models that cannot be adapted to suit a Grid Computing environment.
- There are no Quality-of-Service guarantees in most of these models. They are prioritised for one-sided deals giving all the advantages to the Provider. Consumer preference and selection is not a priority.
- The number of messages exchanged in Auction models before the resources are mapped is an overhead that can use most of the available bandwidth before the service is initialized.
- For effective scheduling of the jobs, approximate application completion time should be known to the consumer before the application is run (budget constraints, deadline constraints, time constraints).

The provision of reward-based computing means the user does not need to specify any constraints that would be required by the models described previously. The only requirement is that the user offering sufficient renumeration for job exectution. It is therefore in the Providers best interests to facilitate the user requirements in turns of quality of service, guarantee of throughput and cost effectiveness.
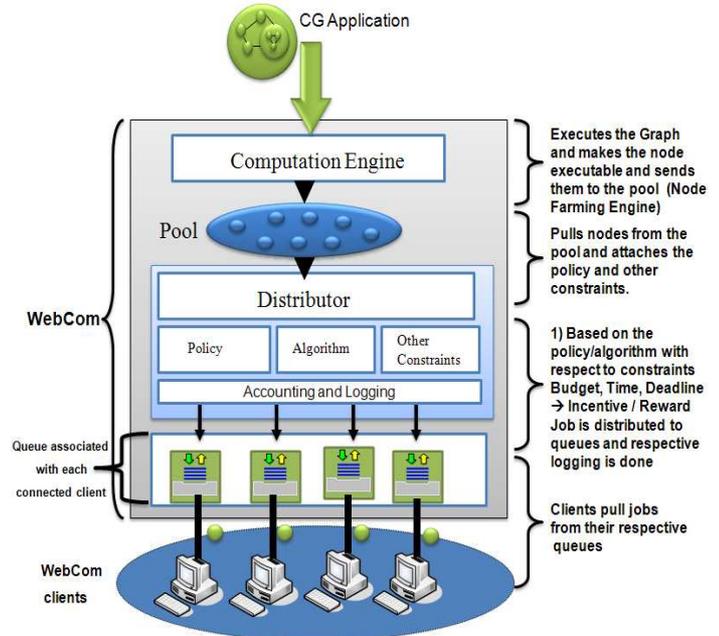

Fig. 2. Reward-Based Distributor in WebCom

IV. REWARD-BASED COMPUTING USING WEBCOM

WebCom [29] is a 'fledging Grid Operating System', designed to provide independent service access through interoperability with existing middlewares. It is based on the Condensed Graph (CG) [28] model of Computing, which is a graph based model that uses Directed Acyclic Graphs (DAGs). The core architecture of WebCom (see Fig. 1) consists of the following modules:

- The Condensed Graph Engine Module, which executes applications expressed as condensed graphs by uncovering fireable instructions and placing them in a *Pool*.
- The Distributor Module (see Fig. 2) performs the actions found in the traditional scheduler and load balancer. It receives instruction from the pool and selects a client to execute an instruction based on configured *Policies and*

Algorithms. Once a node is selected for distribution, it is placed in the clients server-side allocation queue.

- The Fault Tolerance Module monitors client resources and executing applications and reschedules work that was sent to machines that failed.
- The Security Module can be used to enforce different security policies on executing applications.
- The Communication/Connection Manager Module is responsible for transporting nodes to the selected WebCom instances.
- The Information Module is responsible for providing the status of the resources and module information of each WebCom instance.
- The Job Manager is responsible for tracking the execution of a job across the entire WebCom network.

A. Distributor Module

The Distributor (see Fig. 3) makes decisions on when and where to distribute instructions. Its operation is dictated by policies supplied both by site owners (statically) and users (at application runtime). Policies specify the behaviour of the Distributor module. These policies specify settings such as when to request work and what algorithms to use for load balancing. These algorithms include Round Robin and FIFO. Users can supply their own implementation of algorithms which can be used by their policies.

Policies provide rules and heuristics that allow the Distributor to make scheduling, load balancing and communication decisions. The behaviour of each WebCom instance is dictated by a hierarchy of policies. This hierarchy spans administration, system, graph and node policies. The site policy supersedes all others and is specified by the system owner. Next in the order of precedence is the administration policy, followed by graph and node policies. Graph policies travel between WebComs with their associated graphs. Likewise, node policies travel with associated nodes. Graph and node policies can be supplied by the user at run time.

Policies are specified as text files, and hence changing a policy requires little effort as no code re-writing is needed. Policy changes can be carried out dynamically. Policy specifications can include
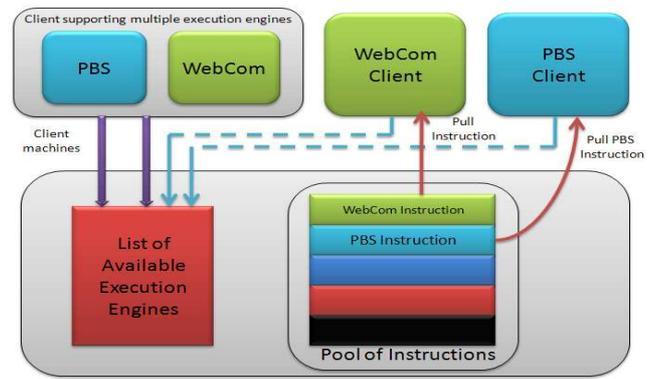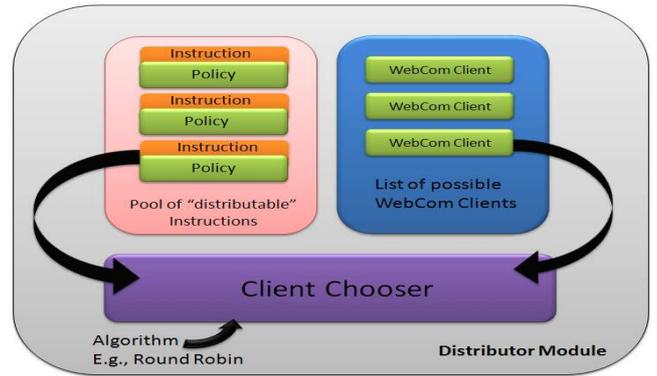

Fig. 3. The WebCom Distributor Module.

Fig. 4. Choosing appropriate clients within WebCom

heuristics such as pre-staging of data, node priorities and node groupings where, for example, inter-dependant nodes can be dynamically allocated to the same machine for execution (perhaps due to side-effects).

The Distributor Module has a pool of instructions handed to it by the backplane. It iterates through this pool, examining each instruction and its associated policies to determine if it can be executed locally or remotely (see Fig. 4). Client WebComs will then pull those instructions allocated to them.

B. Accounting and Payment Mechanism

Economic models can be easily included within the Distributor Framework. The economic model is implemented through the inclusion of a specific client chooser algorithm and associated policy. Hence, it is entirely possible that some node in a graph may be executed under one economic model, other nodes under a different economic model and still other nodes under no economic model.

The WebCom Grid Bank (see Fig. 5) maintains the financial accounts of both the Consumer and Provider. This bank is trusted by both participants. The payment mechanism used is based on the Grid points obtained for executing jobs. This effectively assigns a score to each machine in the Grid, based on the number of jobs executed. An accounting/event-logging system is maintained in the distributor. This keeps track of and logs, the number of instructions executed on each of the respective providers. As a job is executed, various aspects of its progress are recorded in log messages by every WebCom taking part in its execution. Information stored in the job log includes the time and location of the commencement and completion of each node's execution and the path taken during propagation of the node to its execution location. Because a WebCom network is not fully interconnected, nodes may have to be passed along a path in order to reach their intended execution location.

If a job completes, either normally or abnormally, its log file may be examined, allowing analysis of the execution profile. For a job that has failed, this analysis can provide insight into the ultimate cause of the failure. Also, log messages can be viewed in real-time so real-time decisions about payment to providers can be easily made. Grid points are credited to the Providers' accounts based on the number of executed instructions by a client. When the client wishes to cash in, they will contact the WebCom Grid Bank to claim credit commensurate to the number of Grid points they have earned.

C. How the model works

In a reward-based model, each WebCom server has a connected set of clients (providers). This defines the participants in the WebCom Grid. As far as this model is concerned, if there are no available clients, work will not be processed. The model has no concept of deadlines for application completion. The application has to be completed and client participating in executing the application have to be credited. Initially, both client and provider will have some credit in a WebCom Grid bank [22] [20]. Providers are credited Grid points upon execution of certain instructions. These Grid points are then exchanged for credit. Transaction security is provided by using secure SSL connections between the clients and Keynote credentials to
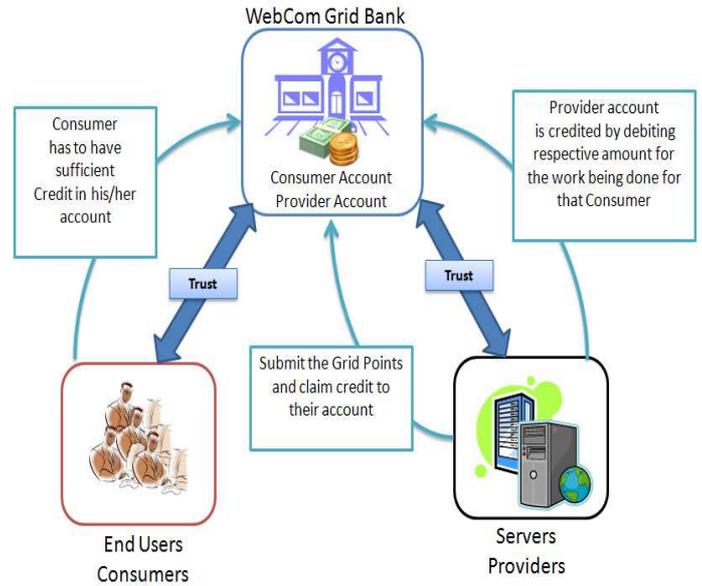

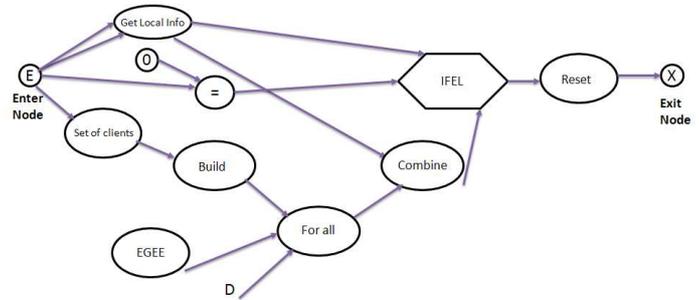Fig. 5.   Payment and Banking System

Fig. 6.   Mandelbrot Graph Application

authorise executions. These mechanisms, developed for WebCom security [12] [13] [31], provide added value to the economic models being developed. The procedure for executing an application using the economic model developed is:

1) Interested clients (Providers) are discovered dynamically and can be addressed by routing different message requests.
2) A Condensed Graph(CG) application is submitted to a WebCom Server.
3) The graph is expanded by the CG Engine to produce nodes. These nodes are passed on to the pool.
4) The pool containing the nodes is serviced by the Distributor thread using an algorithm and policies as specified by the economic model.
5) Nodes are pulled out by these clients for execution. Grid points will be awarded to clients based on the records maintained in

Throughput(instr/time)/Resources	R1(Grid pts)	R2(Grid pts)	R3(Grid pts)	R4(Grid pts)	R5(Grid pts)	R6(Grid pts)
<=5	1	0.5	1	2	0.2	1.5
6-10	2	3	5	8	9	11
11-20	8	5	7	10	4	9
21-30	10	11	12	14	13	10
32-40	14	14	18	15	16	15
41-50	18	19	23	25	28	28

TABLE I

Cost Matrix showing the cost in terms of Grid points relative to the throughput of each machine. Grid points indicate the cost per instruction for a given throughput

Resources	Instructions executed	Total Execution time (seconds)	Throughput (instr/time)	Cost(Grid Pts)
R1	144	9	16	1152
R2	47	29	1.62	0.81
R3	76	24	3.167	38.004
R4	78	26	3	54
R5	56	25	2.24	35.84
R6	56	40	1.4	32.2
			Total cost =	1312.854

TABLE II

EXECUTION PROFILE

the Accounting module on how many instructions (throughput) have been executed by the Clients.

6) Clients with these Grid points can visit the WebCom Grid Bank and cash out.

V. EXPERIMENTAL SETUP AND RESULTS

The experimental setup configuration is shown in Fig. 7. For these experiments the economic model employed considered only the providers and consumers executing on the WebCom network. In total, six providers and one consumer wishing to run the application were used. The application executed is represented by the graph in Fig. 6. The cost matrix for instruction throughput for each provider is shown in Table I. In a production environment, this matrix would be based on the running costs of the machines involved, such as power, cooling, administration and maintenance. The event and logging system keeps track of the jobs executed on each provider. Table II lists the number of instructions executed by each provider. It also shows the total time taken to execute these instructions. By cross referencing the values in Tables I and II, it can be seen that the cost of executing the graph is 1312.854 Grid points. This sample execution indicates how work can be appropriated using this

economic model. It is possible to alter the execution profile based on the cost matrix. For example, if a cost matrix is known *a priori*, the economic model could be configured to schedule work based on any number of factors such as minimum time (by selecting the clients that provided maximum throughput) or minimum cost (by selecting the clients offering the maximum throughput for a minimum cost).

Consider the profile shown for machines *R5* and *R6*. Here, both machines execute the same number of instructions, with *R5* executing its allocation 15 seconds faster than *R6*, albeit at an additional cost of 3.64 Grid points. We can apply the following reasoning to the decision making process: Is it feasible to pay a additional 3.64 Grid points to ensure a 15 seconds speedup in the execution of these instructions?

However, this reasoning has to be balanced with the delay incurred by scheduling the second batch of instructions, i.e., should the decision be made to execute the instructions on *R5*, it would take 50 seconds in total to execute all 112 instructions, whereas machines *R5* and *R6* would take 25 and 40 seconds respectively. Assuming work has been scheduled on both machines at the same time, the worst-case scenario here would be that all 112 instructions are executed in 40 seconds. These results

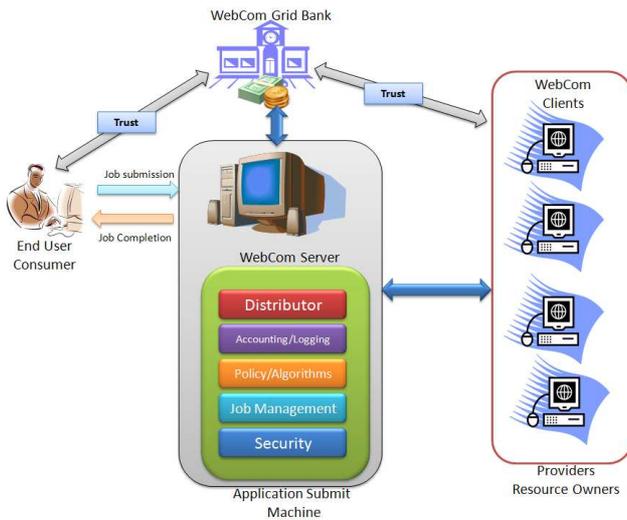illustrate the possible benefits in applying economic models to the scheduling of instructions to clients.

Fig. 7.   High Level View of the Economic Setup

VI. CONCLUSION AND FUTURE DIRECTIONS

In this paper, a number of economic paradigms were presented. A reward-based model used as a base for an economy based computing using WebCom was also presented. The modular architecture of WebCom allows rapid development of economy-based computing within a Grid Environment. We have presented a simplified producer consumer model based on the exchange of Grid points. However several directions for the future enhancement of economy-based computing using WebCom are identified. These include the provision of:

- Incentive-basedcomputing to act as a driving force to keep the Grid alive.
- Reward-based computing to support quality of service and on-demand completion of jobs (completion of a job with respect to a deadline).
- Use of Keynote certificates to enhance the decision-making process of deciding where work is sent. This would reduce the volume of messages sent between WebComs. Instead of messaging computational resources for their interests in offering services, Keynote credentials that describe these interests will be deployed on

these resources. When new service providers are detected, an initial discovery process will be launched to identify the resources provided via the security credentials.

- A Penalty system to penalize the providers who fail to adhere to the agreements made.
- Automated Banking for crediting/debiting participants' accounts.

REFERENCES

[1] Mojo nation. link: http://sourceforge.net/projects/mojonation/.
[2] Platform Computing: Symphony. http://www.platform.com/products/Symphony/.
[3] Sun n1 grid engine 6, link: http://www.sun.com/software/gridware/.
[4] T. Ackemann, R. Gold, C. Mascolo, and W. Emmerich. Incentives in peer-to-peer and grid networking. Torsten Ackemann, Richard Gold, Cecilia Mascolo, and Wolfgang Emmerich. Incentives in peer-to-peer and grid networking. UCL Research Note RN/02/24, 2002., 2002.
[5] E. Adar and B. Huberman. Free riding on gnutella. Eytan Adar and Bernardo A. Huberman. Free riding on gnutella. Technical report, Xerox PARC, 10 Aug. 2000., 2000.
[6] S. Agrawal, J. Dongarra, K. Seymour, and S. Vadhiyar. NetSolve: Past, Present, and Future - a Look at a Grid Enabled Server.
[7] D. P. Anderson, J. Cobb, E. Korpela, M. Lebofsky, and D. Werthimer. Seti@home: an experiment in public-resource computing. *Communications of the ACM*, 45(11):56–61, November 2002.
[8] R. Buyya. Economic-based distributed resource management and scheduling for grid computing. Ph.D Dissertation.
[9] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger. Economic models for resource management and scheduling in grid computing, 2002.
[10] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science*, 2009:46+, 2001.
[11] E. David, R. Azoulay-Schwartz, and S. Kraus. An english auction protocol for multi-attribute items. In *AAMAS '02: Revised Papers from the Workshop on Agent Mediated Electronic Commerce on Agent-Mediated Electronic Commerce IV, Designing Mechanisms and Systems*, pages 52–68, London, UK, 2002. Springer-Verlag.
[12] S. N. Foley, B. P. Mulcahy, T. B. Quillinan, M. O'Connor, and J. P. Morrison. Supporting heterogeneous middleware security policies in webcom. *Journal of High Speed Networks*, 15(3):301–313, 2006.
[13] S. N. Foley, T. B. Quillinan, M. O'Connor, B. P. Mulcahy, and J. P. Morrison. A framework for heterogeneous middleware security.
[14] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing*, 11(2):115–128, Summer 1997.
[15] A. Grimshaw, A. Ferrari, F. Knabe, and M. Humphrey. Legion: An operating system for wide-area computing. *IEEE Computer*, 32((5)):(??), 1999.
[16] P. Haddawy, N. Rujikeadkumjorn, K. Dhananaiyapergse, and C. Cheng. Balanced matching of buyers and sellers in e-marketplaces: the barter trade exchange model. In *ICEC '04: Proceedings of the 6th international conference on Electronic*

---

[0] Incentive can be described as a special reward awarded to complete the work on demand, keeping quality of service to the maximum.

commerce, pages 85–94, New York, NY, USA, 2004. ACM Press.

[17] L. He and T. R. Ioerger. Task-oriented computational economic-based distributed resource allocation mechanisms for computational grids. In *IC-AI*, pages 462–468, 2004.

[18] O. Heckmann, A. Bock, A. Mauthe, and R. Steinmetz. The eDonkey File-Sharing Network. In *Workshop on Algorithms and Protocols for Efficient Peer-to-Peer Applications, Informatik 2004*, Sept. 2004.

[19] J. Hoskins. *IBM On Demand Technology Made Simple: Understanding the IBM On Demand Business Strategy and Underlying Products (MaxFacts Guidebook series)*. Maximum Press, Gulf Breeze, FL, USA, 2005.

[20] A. Kothari, S. Suri, and Y. Zhou. Bandwidthconstrained allocation in grid computing, 2003.

[21] R. Kothari, M. K. Mohania, and Y. Kambayashi. Increasing realized revenue in a web based dutch auction. In *EC-WEB '02: Proceedings of the Third International Conference on E-Commerce and Web Technologies*, pages 7–16, London, UK, 2002. Springer-Verlag.

[22] S. Kungpisdan, B. Srinivasan, and P. D. Le. A secure wireless prepaid micropayment protocol with extension to postpaid micropayment. In *iiWAS*, 2004.

[23] K. Kutzner and T. Fuhrmann. Measuring large overlay networks - the overnet example. In *Konferenzband der 14. Fachtagung Kommunikation in Verteilten Systemen (KiVS 2005)*, Kaiserslautern, Germany, 2005.

[24] S. M. Larson, C. D. Snow, M. R. Shirts, and V. S. Pande. Folding@home and genome@home: Using distributed computing to tackle previously intractable problems in computational biology. *Computational Genomics*, 2002.

[25] A. Manning. A generalised model of monopsony. *The Economic Journal*, 116(508):84–100, January 2006.

[26] P. P. Mcafee and J. Mcmillan. Auctions and bidding. *Journal of Economic Literature*, 25(2):699–738, 1987.

[27] M. J. Miranda. Numerical strategies for solving the nonlinear rational expectations commodity market model. *Comput. Econ.*, 11(1-2):71–87, 1998.

[28] J. P. Morrison. *Condensed Graphs: Unifying Availability-Driven, Coercion-Driven and Control-Driven Computing*. PhD thesis, Technische Universiteit Eindhoven, 1996.

[29] J. P. Morrison, B. Clayton, D. A. Power, and A. Patil. Webcom-g: Grid enabled metacomputing. *The Journal of Neural, Parallel and Scientific Computation. Special Issue on Grid Computing.*, 2004(12)(2):419–438, April 2004.

[30] K. Peng, C. Boyd, E. Dawson, and K. Viswanathan. Five sealed-bid auction models. In *ACSW Frontiers '03: Proceedings of the Australasian information security workshop conference on ACSW frontiers 2003*, pages 77–86, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

[31] T. B. Quillinan, B. C. Clayton, and S. N. Foley. GridAdmin: Decentralising grid administration using trust management.

[32] A. L. Rosenberg. Accountable web-computing. *IEEE Trans. Parallel Distrib. Syst.*, 14(2):97–106, 2003.

[33] L. F. G. Sarmenta. Volunteer computing.

[34] M. Shubik. A double auction market: Teaching, experiment, and theory. *Simul. Gaming*, 36(2):166–182, 2005.

[35] M. A. Spence. Monopoly, quality, and regulation. *The Bell Journal of Economics*, 6(2):417–429, 1975.

[36] P. Strong. Enterprise grid computing. *Queue*, 3(6):50–59, 2005.

[37] The napster website.

[38] X. Vives. *Oligopoly Pricing: Old Ideas and New Tools*. The MIT Press, March 2000.

[39] R. Wolski, J. Brevik, J. S. Plank, and T. Bryan. Grid resource allocation and control using computational economies. In F. Berman, G. Fox, and A. Hey, editors, *Grid Computing: Making The Global Infrastructure a Reality*. John Wiley & Sons, 2003.

[40] R. Wolski, J. S. Plank, J. Brevik, and T. Bryan. Analyzing market-based resource allocation strategies for the computational grid. *International Journal of High Performance Computing Applications*, 15(3):258–281, Fall 2001.

[41] Y. Zhu, L. Xiao, L. M. Ni, and Z. Xu. Incentive-based p2p scheduling in grid computing. In *GCC*, pages 209–216, 2004.