Gridlet Economics: Modelo e Políticas de Gestão de Recursos num Sistema para Partilha de Ciclos

Gridlet Economics: Resource Management Models and Policies for Cycle-Sharing Systems

Pedro Oliveira, Paulo Ferreira and Luís Veiga
INESC-ID / Instituto Superior Técnico - Technical University of Lisbon
Rua Alves Redol No 9, 1000 Lisboa, Portugal
Email: pedro.f.oliveira@ist.utl.pt, {paulo.ferreira, luis.veiga}@inesc-id.pt

ABSTRACT
In cycle-sharing peer-to-peer systems, the users contribute to a pool of resources which they can all use. The access to the resources can be made by many users simultaneously, so there is the need to define which resource each one will use. In this paper we propose an economic model for the management of resources in those systems, matching jobs to resources according to a flexible set of requirements. In order to use the resources of the system the user makes a transaction where he exchanges credits for the right to use them, those credits can only be received by previously contributing to the system. Thus the model encourages or forces the users to contribute, which is essential in a peer-to-peer system. To reduce the risk of the transactions a reputation system is used that penalizes misbehaving users.

Keywords
Cycle-sharing system, peer-to-peer network, economic model, resource management.

1. INTRODUCTION
In peer-to-peer cycle sharing systems the users make their home computers, and respective computational resources, available to be used by others. This way they create a global pool of resources with a huge computational power. The idea is that when they have a job that is very computationally intensive and that would take too long to be executed only on the machines that the users own, they can use the resources of the pool to make execution much faster. However, there is the need to regulate and manage the use of the resources in that pool. In this paper we propose a new model designed to do that management, called Gridlet Economics.

Revised version of the article published and presented in the GPC 2011, Oulu, Finland. Published in the Lecture Note in Computer Science (LNCS), Springer, May 2011

Although much research has been done already on peer-to-peer systems [1], there is the need for a new model because so far the main focus has been on file-sharing. The resource management model used in those systems only deals with a binary requirement, having the file or not. This cannot be applied to cycle sharing, because these environments have to deal with many and varied requirements, such as CPU speed, number of cores or OS installed. Nevertheless, other resource management models can be considered, such as the ones used in the BOINC system or in Grids. Despite not being intended as peer-to-peer, these systems also use several commodity computers to create a huge pool of resources. However, the strict client server architecture used in BOINC does not consider concurrency to the access of resources, since the only one that can use them is the central server hosting the project. This is unusable in a peer-to-peer system since one of the main characteristics of those systems is that everyone can act as a client and a server. On the other hand, in Grids, it is assumed that all computers are trustworthy, that the components are relatively static and that there is no need to encourage contributions, assumptions that cannot be made in a peer-to-peer system.

Therefore, in this paper we present Gridlet Economics, an economic model for the management of the resources in a peer-to-peer cycle sharing system which is able to match the jobs to the resources according to a rich set of requirements. The basis of this model are the transactions where a user exchanges credits for the right of using the resources of the system. Those credits can only be acquired by contributing to the system, thus the model drives the users to contribute, which is essential in a peer-to-peer system because it depends on the contribution of the participants to survive. Simultaneously, this model prevents users from controlling too many resources and/or their prices by virtue of their accumulated credit, keeping the system fair. Moreover it regulates in a fair manner the access to the resources, because users which contribute more have more credits and therefore can use more of the resources of the system. Also, this model uses a reputation system to penalize misbehaving users and prevent them from benefiting from their actions.

2. RELATED WORK

2.1 Economic Models

When there is demand and supply of resources there has to be some management of who uses what and when. In the real world, economic models have been used to do so for centuries and, with the appropriate regulation and regardless of ideology, have proved to be a successful and sustainable way of governing the exchange of resources, goods and services. Also, the use of an economic model provides a scalable option for the management of resources, especially when they are not all in the possession of the same entity because each user regulates the use of its own resources. Moreover, it provides a simple method for defining the priority order of the jobs, by establishing that the ones for which the users are willing to pay more have the highest priority. Another advantage of the use of an economic model is flexibility since it allows a uniform treatment of all kinds of resources, from CPU time to application version. In fact, there are already some systems that use economic models for resource management [2, 3]. However, none addresses the particular aspects of resource management in a peer-to-peer cycle-sharing system.

When using an economic model there are three main aspects that have to be considered: “the currency”, “how the price is defined”, by the supplier or seller, and “how it is selected”, by the consumer or buyer. The selection of what is used as currency is an important factor because it is what is exchanged in the transactions. The currency-based systems can be divided into two main categories: non-monetary and monetary. The non-monetary systems, such as bartering [4] and direct exchange are used in primitive economies because they are simple and easy to build, however they are very inflexible and limited. On the other hand, monetary systems are much more flexible, but also more complex due to the use of money. Virtual money is one form of money that only has value inside the system in which is used. This type of money, used in [16], is considered appropriate for systems without a controlling third party, such as the P2P system, because it has far fewer security concerns than real money.

One of the bases in an economic model is that the consumer has many options of services, or resources, provided by different suppliers, from which he can choose. Each alternative offers a different type of service for different cost and a decision of which is selected has to be made. In order to make that decision, a function that calculates the utility value of each option is used. The utility is a measure of the relative satisfaction of the consumer with the consumption or purchase of a determined good or service, so the right selection can be made by choosing the option that maximizes that value. On the other hand there is also the need to define the price that the suppliers will charge for the resource consumption, or use, and to do so there are several ways. The more traditional one is Commodity market where the resource owners specify the price and the consumers only have the choice of buying or not. The prices can be defined by the owner using a flat policy, where the price does not change for a certain amount of time, or variable one, where price changes very often based on the amount of supply and demand. Other options exist, such as bargaining or auctions [11], where the consumers not only have the choice of buying or not, but also influence the value asked. It is up to the resource owner to decide which method is used.

2.2 Reputation

Trading involves always a certain amount of asymmetric risk (that one of the parts will not fulfill its obligations once the other has committed the resources or currency) and, if it is not controlled, that risk can lead to the collapse of the economic model. The reputation is a good way of reducing the risk without the need of a supervising third party. Besides reducing the risk, reputation can also be used as a mechanism to induce good behavior in this type of markets. That concept is not new, in fact several economists have already published work analyzing its proprieties and one of the major auction sites in the world, eBay [6], relies almost exclusive on a system like this to reduce the risk and induce good behavior on the part of its members.

In order to calculate a user’s reputation the system collects the classification that the users give to him and then converts it into a value that can be quantified and compared using a reputation function [5]. In a centralized system, like eBay, that value is calculated and stored in a central server which is considered a trusted third party. However there are decentralized alternatives such as the Eigen Trust Algorithm [7], which can be used on a peer-to-peer system.

2.3 Resource discovery

As it has already been said, one of the basis of an economic model is the possibility of selecting the option that maximizes the utility of the resource consumer. In order to make the selection there is the need to know what options are available, for that there is the need to discover the resources available at a certain time. Regarding resource discovery, peer-to-peer systems are divided into three main categories: unstructured, structured and hybrid.

In unstructured peer-to-peer systems, such as Gnutella, the nodes are randomly distributed and therefore have low costs to enter and leave the system, which makes them appropriate for highly-transient populations. However, because there is no information about the organization of the peers, these systems have to use uninformed search techniques which are very inefficient. To compensate for this, Kazaa uses a hierarchic approach with two classes of nodes, super peers and ordinary nodes. Only the super peers participate in the search, this way the number of nodes searched is greatly reduced, making it more efficient.

In structured peer-to-peer system such as Chord [14], CAN (Content Addressable Network) [10], Pastry [12] and Kademlia [8], the peers are distributed using a DHT (Distributed Hash Table) in which peers and keys are mapped through the hash function. This allows the finding of the peer corresponding to a determined key to be very efficient. However, the high cost for entering and leaving this type of systems makes then less suitable for highly transient populations.

Many hybrid systems try to maintain the efficiency of the search of the structured systems while also employing a less strict organization, akin to unstructured system, thus making them more suitable for transient populations. This is usually achieved by having a set of super-peers belonging to a smaller structured overlay, while the majority of other peers discover content and resources only by mediation of a super-peer.

3. GRIDLET ECONOMICS

The model proposed is designed to be applied on top of a peer-to-peer cycle-sharing system. In this system the users make their computers, and respective computational resources, available to be used by the other participants. When a user wants, he can submit jobs that are executed on the resources of the other users of the system, making its execution much faster. In order to make it possible for the job to be distributed over the resources available in the system, it is split into smaller work units. These units, called gridlets, are the main input of the model and it is assumed that an existing layer is responsible for their creation and the aggregation of the result of their execution, such as that described in [15].

In practice, the objective of the model is to offer a decentralized mechanism that maps the gridlets submitted to the computers where they will be executed. The selection must take into consideration a set of requirements that correspond to the user utility and the contribution that the user has made to the system. These requirements will be represented and evaluated using the method described in Partial Utility Algebra [13].

3.1 Model Overview

In this model the computers present in the system, called the nodes, can be classified as producers, if they are contributing with resources to the system where gridlets can be executed, or consumers, if they have gridlets being executed in the system and therefore are consuming the resources of the system. Although a node can have both classifications at the same time, normally it will alternate between them, since when it is a consumer it will not contribute with his resources to the system, but use them to also execute some of its gridlets. The model also uses a hierarchical approach with two classes of nodes, brokers (similar to super peers) and ordinary nodes. The brokers maintain a global distributed index where all the nodes that are contributing to the system are registered and use it to select the executor of a gridlet, i.e. the producer that will execute it. Although the classification of the nodes is independent of the division in classes, it is considered that

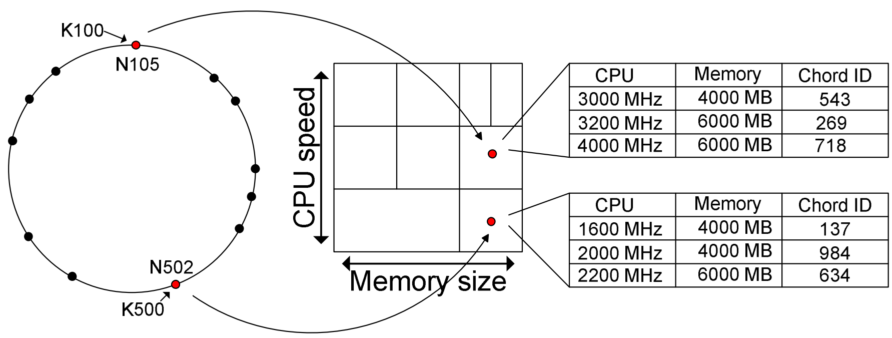

Figure 1. Model overview

the brokers will not be producers and therefore will not execute gridlets. This is done in this way because if a broker had to simultaneously execute a gridlet and make the selection of an executor, both activities would be slower, therefore, since the selection of executor is a crucial element of the model precedence is given to that task. Besides, when the brokers are selecting a producer to execute a gridlet, they are already contributing with resources to the system. Also, a possible conflict of interests could arise, similar to inside trading, because that node could be at the same time, the selector and the selected for the gridlet execution.

All the nodes in the system will be distributed over a Chord ring. The use of this look up service provides an efficient communication mechanism and a unique ID for each node that can be maintained between sessions. Also, the brokers are defined as the nodes that are responsible for a predefined set of keys (e.g. 1000, 2000, etc.) in the ring. This way when a node wants to communicate with a broker, he only has to calculate the closest predefined key and send the message to it. This allows a transparent change of the node that is acting as the broker and an even distribution of the load across all them. Nevertheless, all of brokers can be easily found, which means that if the user wants he can choose to use another one. Figure 1, shows an overview of the model, where the nodes N105 and N502 are responsible for the pre-defined keys K100 and K500 in the Chord ring and therefore are brokers, which maintain a distributed index of the nodes in the system.

The bases of this economic model are the transactions, where the consumer pays to have its gridlets executed on the resources available in the system. The payments will be made using credits, a currency of a virtual money system. The execution of each gridlet is cons idered a different transaction and it starts when the gridlet is received by the consumer (depicted next in Figure 2). First, it attaches to the gridlet the requirements that represent the user utility and then sends it to a broker (Step 1). Next, the broker that receives the gridlet, together with the other brokers, selects the node that according to the specification is better suited to execute it (Step 2). Then the gridlet is passed to the node selected and if that node is idle it will immediately execute the gridlet, otherwise it will put it in a local queue to be executed later (Step 3). During the execution, the producer executing the grilet will monitor its resources' consumption and, after the execution terminates, its result is sent back to the consumer along with an invoice (Step 4). In the end, the consumer pays the invoice, which includes the nodes that stored the result, and classifies the producer. After receiving the payment the producer classifies the consumer, thus ending the transaction (Step 5).

3.2 The Market Square

The market square is an index where the producers advertise their resources, so that they can be selected to execute the gridlets of the consumers. In practice, it is a table that contains an entry for all of the nodes that are contributing with their resources to the system. Each entry will have the Chord ID of the node and its characteristics (price, CPU speed, memory size, OS installed, etc.). This table will be indexed by a set of predefined characteristics, called the unit of cost. Since the search is done primarily through those characteristics, they should be things in which most users have an interest in, such as price, CPU speed, memory size or network bandwidth.

However, in a peer-to-peer system it is not possible for one node to store information about all the other nodes in the network, so the index will have to be distributed among all the brokers. In order to distribute the index among them without losing the ability to efficiently search through it, the brokers will organize themselves in a d-dimensional CAN, where each dimension corresponds to a characteristic of the

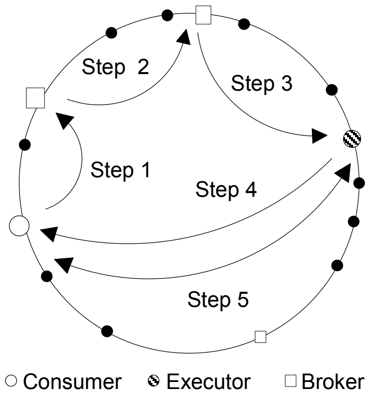

Figure 2. Steps in a transaction

unit of cost. Then, each broker will have the part of the index which corresponds to the characteristics that match the zone for which it is responsible. This means that the node responsible for the zone between 1950 and 2050 in the dimension of the CPU speed will have the index of the ordinary nodes that have the CPU speed in that range. In this way, it is still possible to travel efficiently through the index using the CAN routing mechanism. Though, normally the range of values and scale will differ from characteristic to characteristic and, if those exact values are used to define the CAN dimensions, this would hinder its routing system. So, instead of using the actual values of the dimensions a canonical representation is used. For example, from 0 to 100 and the conversion is made by calculating a percentage against the difference between the bottom and top value.

3.3 Selection of the Executor

The selection of the node that will execute the gridlet is done by evaluating the entries of the market square according to the requirements specified in the gridlet, which represent the user utility, and then choosing the node that is better suited to execute it. This means that all the nodes contributing to the system have to be published in the index of the market square, otherwise they will never be selected to execute a gridlet and therefore it is as if they were not contributing. However, as the number of entries of the index increases, it will become unfeasible to evaluate them all. Also, in real economy this sometimes happens, where consumers are unable to check the individual price of all producers or sellers. Furthermore, the index is split into zones that are stored in different nodes. So, instead of doing a full index search, the search starts at the point that is the most likely to maximize the user utility and stops when it estimates it that has reached the best alternative.

To find the point that is the most likely to maximize the user utility, the unit of cost is used. For that, it is assumed that those characteristics have a great interest to all of the users and also that they all agree in whether they should be maximized or minimized. In an economic cycle-sharing environment this can be easily achieved, if the characteristics considered are features such as price, processor speed or network bandwidth. Thus, the broker that first receives the gridlet uses the routing of CAN to send it to the point that maximizes the utility, which corresponds to the maximum / minimum limit of the dimensions of the characteristics of the unit of cost that should be maximized or minimized, respectively. When the gridlet reaches the best suitable point, a local search is done, which is made by evaluating the nodes in that part of the index and determining the best one. Then an estimation of the values present in the adjacent zones is evaluated. The adjacent zones correspond to the neighbors in CAN, so this is called the neighbor search. This is done to estimate the classification that could be achieved if a local search was done in one of those brokers. If the classification of the node selected locally is higher than the one of the best neighbor, the gridlet is sent to that node to be executed. Otherwise, the gridlet is sent to the broker responsible for best adjacent zone, which repeats the same search steps. The search stops when a node is selected, or neither the node selection nor the neighbor selection returns a possible result; in that case the search fails and the gridlet is returned to the node that submitted it to the system. Figure 3 shows the pseudo-code for the search in the index.

The search is directed in a contrary direction to the one indicated in the unit of cost, i.e. the search can only travel down in the characteristics that are supposed to be maximized and up in the ones that are minimized. This means that only the neighbors that respect this rule are considered in the neighbor search. That restriction is applied to avoid loops without requiring the brokers to maintain state of the ongoing searches, which is a great advantage when considering a great number of simultaneous searches, because it significantly improves the scalability and reliability.

3.4 Prices and payments

The consumers pay for the execution of their gridlets and this creates the need to define how much they will pay. However, the consumption of resources that occurs during an execution can vary much from gridlet to gridlet and it is difficult to estimate beforehand how much resource consumption there will be. This means that if the total price charged for the execution was determined beforehand, one of the parties would most likely lose with that transaction. So, instead of determining the price for the entire execution

```
// search in one zone of the index
searchTheIndex ( gridlet ) {

    node     = localSearch ( gridlet.specifications );
    neighbor = neighborSearch ( gridlet.specifications );

    // both searches failed
    if( node.classification < 0 &&  neighbor.classification < 0 )
        return fail;

    if(node.classification >= neighbor.classification)
        node.execute ( gridlet );
    else
        neighbor.searchTheIndex ( gridlet );
}
```

Figure 3. Pseudo-code for the search in the index

the producer defines a fee, which is the value that will be multiplied by the resources used (in essence, the unitary price charged). This fee can be the same for all the resources or be specified for each resource in particular. The definition of the value of the fee is determined as it is done in the Commodity market, which means that the producer is the only responsible for its definition and will then publish it in the index. This way, the fee is incorporated in the selection of the executor and selected according to the consumer utility. As an incentive to contribution in times of great demand, a variable price definition policy is used, where the price is altered based on the amount of supply and demand. The decision of raising or lowering the fee asked is made automatically based on the size of the queue of gridlets waiting to be executed on that node and on the nodes that have similar characteristics. That information can easily be supplied by the broker that stores the node index entry, since the other nodes will also have the entries in the same zone.

The brokers are also contributing to the system, however only the execution of gridlets is paid, which means that they are not rewarded for their contribution. So, in order to compensate them, they charge a tax for each transaction in which they participate. However, due to the high number of brokers that participate in a search, it is not feasible to pay a percentage to all them. Because for that to be done the taxes would have to be very high or the value paid to each one would not be relevant. So, only the first and the last broker involved in each search are paid. In this way, since the selection of the first broker done by the consumer is random, half of the taxes will be distributed uniformly among the all brokers. And the other half will go to the brokers that do the neighbor and local search, which are the operations that require more work.

3.5 Risk and reputation

In every transaction there is the risk that one of the parts might not behave properly and tries to take advantage of the other. In this model this can also happen, so a reputation system is used to minimize and control that risk by penalizing the users that act maliciously.

As a producer participating in a transaction, the user can take advantage of the consumer in two ways: fraud and overpricing. It is considered fraud when the producer instead of executing the gridlet returns a fake result and asks to be paid. When this happens, and is detected [9], the consumer gives a bad classification to the transaction and refuses to pay.

Overpricing is when a producer claims that the execution used more resources than it actually did and this way tries to charge a price higher than it should. Since it is difficult to know how many resources an execution will consume without executing it, this situation is hard to detect. In order to detect it, is assumed that the gridlets are passed by the consumer in groups with a relation among them, which is normal considering that a job will generate a set of similar gridlets [15]. Then, the price charged for the gridlets of the same job is compared in order to define an acceptance threshold and if one of the prices charged is higher than that threshold, it is considered overpriced. In this case the consumer also gives a bad classification and pays only the amount he considers fair. The bad classification is important because the reputation is inserted in the search. A producer with a lower reputation will have to lower the fee asked to compensate and eventually will stop being selected to execute the gridlets, which means that will not be able to gain credits to use the system.

On the other hand, the option of not paying gives the consumer the possibility of having some gridlets executed for free by falsely claiming that it was the victim of fraud. In extreme cases this would lead to the appearance of free-riders, nodes that never paid to use the system. In order to prevent this from happening, when the consumer submits a gridlet he has to pay a deposit, a value which is always paid to the producer that executed the gridlet. Moreover, after receiving the payment the producer also classifies the consumer. This is important because a consumer with a low reputation has to pay higher deposits. Also, a node is a consumer and producer, so a low reputation as a consumer will make him have a low reputation as a producer and vice versa.

4. EVALUATION

In this section, we describe the evaluation of the model in simulation in order to determine its capability to match jobs to the available resources, how well the users' utility is fulfilled, and the behavior of the variable price definition policy. First, we describe the simulation environment.

For simulation, the model was implemented on the PeerSim simulator, using an event driven simulation. The simulation used 10 000 nodes with a Chord ID of 90 bits randomly generated. The predefined keys for the broker position were the multiples of 1025, which creates the

Table 1. Characteristics of the unit of cost

Name	Unit	Minimum	Maximum
fee	credit	1	100
CPU speed	MHz	1000	5000
n. of cores	-	1	8
memory size	MB	500	8000
Upload	MB/s	1	100
Download	MB/s	1	100

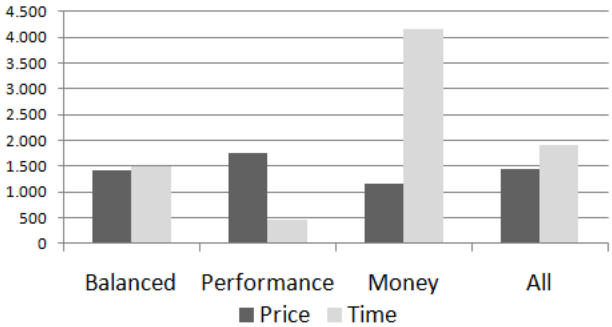

Figure 4. Results for the different type of users

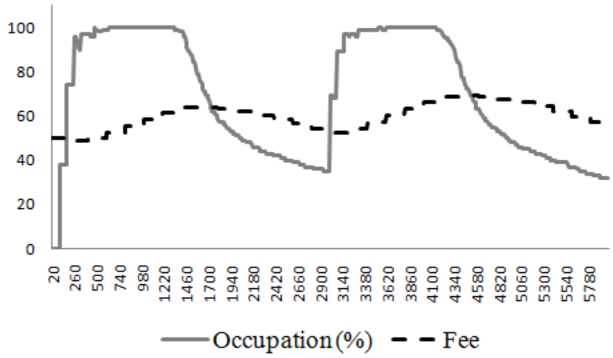

Figure 5. Variation of the fee with the occuation

presence of 123 brokers in the system. Table 1 presents the values of the characteristics used as the unit of cost. The characteristics of the nodes of the population were generated randomly. This generation had the following restriction: if a node had high values in one characteristic, it would have high values on all them. This is done because normally if a computer has a fast processor it also has a lot of memory space. In the beginning of all simulations, all the nodes start with 5 000 credits, so that they can pay for the jobs that they are assigned to submit into the system. This is done to avoid the long bootstrap time required if the credits were initial solely spread through transactions. The jobs consist of 10 gridlets all with the same size and the execution time is calculated as a function of the executor characteristics, which means that the higher the value of the characteristics the less time the execution will take. The time unit used is the tick, which is marked by PeerSim. The price is then calculated by applying the fee to the time the resources were busy.

4.1 Job distribution

First we tested the ability of the model to distribute the jobs over the available resources. For that, 40 000 gridlets are progressively inserted into the system, which is four times the system overall capability. The requirements of the gridlets have the preference for a node as less occupied as possible. As a result, when the system was full of gridlets it was achieved an node occupation of 100% and an average queue size of 3.9. Therefore we can conclude that the model is able to distribute the load evenly across all the nodes of the system.

4.2 Consumer utility

Next, in order to test the ability of the model to satisfy the users' utility, three types of users were created: balanced, performance and money. The balanced type of user wants to achieve a good balance between the execution time and the price paid. The performance type of user wants to get the execution done as fast as possible regardless of the cost. The money type wants to pay the least possible for the job execution. Figure 4 shows the average price paid and execution time for the three types of users. As it can be observed the type of users interested in having their jobs executed in the less amount of time possible, achieve an execution time three times faster than the time it takes to the balanced nodes. On the other hand the nodes which want to save the credits pay almost less twenty percent than the balanced nodes, however that comes with an execution time two and half times slower.

4.3 Variable price policy

The last aspect tested was how the variable price definition policy deals with variations of the demand and supply. Since we consider that the variations in the demand are cyclical, we present the results of the extent of two cycles (of 3000 ticks each) where the system was flooded with gridlets to be executed, during which the average fee asked was monitored. As it can be seen in Figure 5, the initial fee asked is of 50 and lowers a little because when simulation starts the system is empty, but after that the fee asked follows the trends of the occupation/utilization of the system. The average fee is raised while the system is heavily loaded and then lowered when it starts to have many nodes available, meaning that the offer is higher than the demand. This shows that resource management with our economic model is able to adapt to fluctuations in supply and demand. In greater demand, a premium is awarded to suppliers; with greater supply, an equivalent refund is awarded to consumers.

5. CONCLUSIONS

In conclusion the model presented provides a decentralized mechanism for the management of the resources in a cycle sharing peer-to-peer system which is able to distribute the jobs across the resources available in the system. In order to do that, it does not require any node to have global knowledge of the entire system. The resource management model allows each user to define its own requirements and is able to make the selection according to those requirements. This is an important aspect in a peer-to-peer system due to heterogeneity of users that can exist.

The model also offers incentives to resource contribution, since the only way of users gaining the credits they need to use the resources of the system, is by contributing themselves to the system. The incentives are improved by the variable price definition policy which creates a larger incentive for the contribution in times where the demand is bigger than the supply. Also, the use of the reputation system prevents misbehaving nodes of taking advantage of the system.

6. REFERENCES

[1] S. Androutsellis-Theotokis and D. Spinellis. A survey of peer-to-peer content distribution technologies. ACM Computing Surveys (CSUR), 2004.

[2] R. Buyya, H. Stockinger, J. Giddy, and D. Abramson. Economic models for management of resources in grid computing. In Technical Track on Commercial Applications for High-Performance Computing, SPIE International Symposium on The Convergence of Information Technologies and Communications (ITCom 2001), 2001.

[3] R. Buyya and S. Vazhkudai. Compute power market: Towards a market oriented grid. In The First IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid 2001), 2001.

[4] L.G.L.M. Carlo, E.T.O. Felipe, and MG França. The Use of Reciprocal Trade as a Model of Sharing Resources in P2P Networks. In Proceedings of the 2009 Fifth International Conference on Networking and Services - Volume 00, pages 91–96. IEEE Computer Society Washington, DC, USA, 2009.

[5] A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, page 132. ACM, 2005.

[6] C. Dellarocas. Analyzing the economic efficiency of eBay-like online reputation reporting mechanisms. In Proceedings of the 3rd ACM Conference on Electronic Commerce, pages 171–179. ACM New York, NY, USA, 2001.

[7] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In Proceedings of the 12th international conference on World Wide Web, pages 640–651. ACM New York, NY, USA, 2003.

[8] P. Mayamounkov and D. Mazieres. Kademlia: A peer-to-peer information system based on the xor metric. In Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS'02). MIT Faculty Club, Cambridge, MA. 2002

[9] J. Paulino, P. Ferreira and L. Veiga. Exploring Fault-tolerance and Reliability in a Peer-to-Peer Cycle-sharing Infrastructure. INFORUM 2010

[10] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker. A scalable content-addressable network. In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, page 172. ACM, 2001.

[11] D. Rolli, M. Conrad, D. Neumann, and C. Sorge. An asynchronous and secure ascending peer-to-peer auction. In Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, 2005.

[12] A. Rowstron and P. Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), volume 11, pages 329–350, 2001.

[13] J.N. Silva, P. Ferreira and L. Veiga. Service and resource discovery in cycle-sharing environments with a utility algebra. In Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium, 2010

[14] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, page 160. ACM, 2001.

[15] L. Veiga, R. Rodrigues, and P. Ferreira. GiGi: An Ocean of Gridlets on a "Grid-for-the-Masses". In Proceedings of the Seventh IEEE International Symposium on Cluster Computing and the Grid. IEEE Computer Society, 2007

[16] V. Vishnumurthy, S. Chandrakumar, and E.G. Sirer. Karma: A secure economic framework for peer-to-peer resource sharing. In Workshop on Economics of Peer-to-Peer Systems. Citeseer, 2003.