Discouraging Free Riding in a Peer-to-Peer CPU-Sharing Grid

Nazareno Andrade Francisco Brasileiro Walfredo Cirne
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
Campina Grande, Brazil
{nazareno,fubica,walfredo}@dsc.ufcg.edu.br

Miranda Mowbray
HP Labs Bristol
mjfm@hplb.hpl.hp.com

Abstract

Grid computing has excited many with the promise of access to huge amounts of resources distributed across the globe. However, there are no largely adopted solutions for automatically assembling grids, and this limits the scale of today's grids. Some argue that this is due to the overwhelming complexity of the proposed economy-based solutions. Peer-to-peer grids have emerged as a less complex alternative. We are currently deploying OurGrid, one such peer-to-peer grid. OurGrid is a CPU-sharing grid that targets Bag-of-Tasks applications (i.e. parallel applications whose tasks are independent). In order to ease system deployment, OurGrid is based on a very lightweight autonomous reputation scheme.

Free riding is an important issue for any peer-to-peer system. The aim of this paper is to show that OurGrid's reputation system successfully discourages free riding, making it in each peer's own interest to collaborate with the peer-to-peer community. We show this in two steps. First, we analyze the conditions under which a reputation scheme can discourage free riding in a CPU-sharing grid. Second, we show that OurGrid's reputation scheme satisfies these conditions, even in the presence of malicious peers. Unlike other distributed mechanisms for discouraging free riding, OurGrid's reputation scheme achieves this without requiring a shared cryptographic infrastructure or specialized storage.

1 Introduction

Grid computing emerged from the possibility of using a large federation of resources as an execution platform for parallel applications. However, building such a federation (i.e., assembling a grid) is not a simple task. There are no widely adopted solutions that make the negotiation between resource consumers and providers automatic.

This current lack of widely adopted solutions is probably due to the complexity of implementing most proposed solutions, which rely on currency-based economies for solving the general grid-assembly problem [5, 6, 13, 1]. In this scenario a consumer can negotiate an arbitrarily complex provision of resources and pay for it in some currency. To deploy a solution for this scenario, it is also necessary to deploy an infrastructure for secure e-cash, e-banking, and service auditing. We argue that this need is currently the greatest obstacle to building a Grid Economy.

Peer-to-peer grids [3, 4, 7, 18] have been proposed as alternatives to the currency-based solutions. They do not solve the general grid-assembly problem. However, they can take advantage of the ease of implementation of peer-to-peer systems to get resource-sharing networks in production today.

We are currently deploying a peer-to-peer grid called OurGrid [3]. OurGrid solves the grid-assembly problem for users of Bag-of-Tasks applications [8], ie. parallel applications whose tasks are independent, such as parameter sweep, massive search and Monte Carlo simulation.

As in most peer-to-peer resource sharing systems, it may be possible for users to *free-ride*, consuming resources donated by others but not donating any of their own. Experience with peer-to-peer systems shows that in the absence of incentives for donation, a large proportion of the peers only consume the resources of the system [2, 15, 17]. Free riding is a concern because it decreases the utility of the resource-sharing system, potentially to the point of system collapse.

In order to avoid this, OurGrid was designed to explicitly provide an incentive for peers to collaborate with the system. To provide this incentive, OurGrid uses a peer-to-peer autonomous reputation scheme, called the Network of Favors. In the Network of Favors, donating a resource is a favor, and each peer autonomously prioritizes peers who have reciprocated more favors in the past. The community does not have to rely on common knowledge to be effective. That is, it is not necessary to store global reputations. We will show that OurGrid's local reputations, which are based only on interactions directly involving the peer that stores them, are sufficient for an effective incentive scheme. Through the autonomous behavior of its components, the system prioritizes the peers who have higher reputations, motivating sharing.

Introducing the reputation scheme does not reduce the overall system utilization, because it only governs which peer receives a contested resource, not whether the resource is donated. Moreover, if peers decide to share more resources as a result of the incentive it creates, this will increase system utilization.

Our analysis relies on the fact that OurGrid peers are expected to be *eager consumers*. A consumer is eager when it gains positive benefit from whatever resources it obtains. We believe that it is reasonable to assume that Bag-of-Tasks application users are eager consumers, for a number of reasons. The desire to use a CPU-sharing grid suggests a large need for computational resources. Bag-of-Tasks applications usually involve a significant number of tasks, and extra resources can be used to increase their makespan using task replication [14]. Finally, CPU is the critical resource for Bag-of-Tasks applications. Most users can recompile their applications to most architectures usually available, if necessary.

In the next section, we analyze conditions under which a reputation scheme discourages free riding among eager consumers. In Section 3, we show that OurGrid's reputation scheme satisfies these conditions, even in the presence of malicious peers. Section 4 compares our approach with some related work, and Section 5 gives conclusions and future directions.

2 Resource Sharing among Eager Consumers

In this section we prove a general analytical result about resource sharing among eager consumers: if the community has some mechanism (not necessarily the mechanism used in OurGrid) by which it can identify collaborators with sufficient accuracy, and known collaborators get priority in access to the resources, then it pays to be a collaborator. As a consequence, if peers change their strategy to collaborating if it is in their interest to do so, then the community evolves to a state where there are no free riders.

Let n be the number of peers in the community (we do not necessarily assume that n is large). Since we allow peers to change their strategy, the number of free riders will vary over time. Let $f(t).n$ be the number of peers that are free riders at time t. The other $(1 - f(t)).n$ peers are collaborators at time t, donating all their spare resources to the community.

Since the peers are eager consumers, a peer that is donated a resource always gains some positive utility as a result, no matter how large the quantity of resources it is simultaneously consuming from other sources. It is therefore reasonable to assume that the utility lost by the donor as a result of donating the resource is a fixed multiple v of the utility gained by the recipient, with $0 < v < 1$.

If $f(t) = 0$, then all peers are collaborators. If $f(t) = 1$ the community contains only free riders, so no resources are donated and there is no incentive for any peer to remain in the community. Suppose now that $0 < f(t) < 1$ at some time t. Let $\epsilon(t)$ be the "error" probability at time t that if a collaborator has a spare resource, it will donate the resource to a free rider. This may happen either because the collaborator cannot distinguish another collaborator to whom it can donate its resources to or because there are only free riders requesting resources at time t. If there are no collaborators at time t, then we define $\epsilon(t) = 1$.

Now if the utility gained by the recipient of a particular resource donation is u, then the total expected utility gain to the set of collaborators as the result of the donation is $(1 - \epsilon(t)).u - v.u$, where t is the time that the donation takes place (the donor must be a collaborator, because free riders do not donate resources), and the total expected utility gain to the set of free riders arising from the donation is $\epsilon(t).u$. Since there are $(1 - f(t)).n$ collaborators, the expected utility gain to an average collaborator is $\frac{(1-\epsilon(t)-v).u}{(1-f(t)).n}$. Similarly, since there are $f(t).n$ free riders, the expected utility gain to an average free rider is $\frac{\epsilon(t).u}{f(t).n}$. Therefore, it is better to be a collaborator provided that $\frac{(1-\epsilon(t)-v).u}{(1-f(t)).n} > \frac{\epsilon(t).u}{f(t).n}$, which happens if and only if $\epsilon(t) < (1 - v).f(t)$. On the other hand, it is better to be a free rider when $\epsilon(t) > (1 - v).f(t)$. For the case that $\epsilon(t) = (1 - v).f(t)$, collaborators and free riders have the same expected utility.

We assume that peers will gradually change their strategies to or from free riding if it is in their interest to do so (i.e., if the expected utility with the new strategy is greater than the expected utility with the old strategy). Therefore, if there is some t' for which $\epsilon(t) < (1 - v).f(t)$ for all $t \geq t'$, then after time t' it will always be in the interest of free riding peers to become collaborators, and so free riding will eventually die out.

3 OurGrid and the Network of Favors

OurGrid, a system that we are currently deploying, is a solution to the problem of automatic grid assembly for users of Bag-of-Tasks applications [3]. Through OurGrid, users get access to the idle processors of the community in a peer-to-peer fashion. The assembled processing power from the community forms the grid.

In OurGrid, idle processors are not explicitly advertised, but requests are propagated through the system to as many peers as possible. Messages typically have several alternative routes to reach peers, so that it is difficult for a malicious peer to block others' requests. Peers with idle processors can allocate the use of these processors to a requesting peer, sending the result of the processing directly to the requesting peer.

OurGrid uses an autonomous reputation scheme called the *Network of Favors* to help peers with idle processors determine which requesting peer to donate to. A key motivation for the design of this scheme was to make it particularly lightweight and easy to implement in real systems. The central idea of the Network of Favors is that the users who are greater net contributors of processing power should get higher priority access to the spare processing power of the community. This principle acts as a guide to the apportioning of the available resources among the users currently requesting them and, thus, as an incentive for collaboration.

In the Network of Favors, allocating a processor to a peer that requests it is a favor, and the value of that favor is the value of the work done for the requesting peer. Each peer keeps a local record of the total value of favors it has given to and received from each known peer in the past. Every time it does a favor or receives one, it updates the appropriate number. The peer calculates a local reputation for each peer based on these numbers, such that a peer who has given many favors and received few will have a high reputation. The peer then uses the current reputations to decide to whom to offer a favor when it has to arbitrate between more than one requester. Thus, whenever there is resource contention, requesters with higher reputation get priority. We expect the scheme to scale gracefully because a peer only needs to keep track of peers that currently have nonzero local reputation, and only needs to store a small amount of information for each of these.

Since an autonomous reputation scheme uses no information on interactions that did not directly involve the peer assessing the reputation, this reduces the options that malicious peers have to distort the reputations. Malicious strategies based on lying about the behaviour of third parties cannot be applied. One of the remaining possibilities for a malicious peer to attack the reputation system is to change identity.

In peer-to-peer networks, it is usually easy for a peer to change its identity by leaving the community and coming back as a supposed newcomer. By this method a peer with a bad reputation can easily start afresh with a newcomer's reputation. Cryptographic or other guarantees of a peer's identity generally can do little to stop this. We do not want to address this by imposing stringent admission controls, as we would like to have as few barriers to the growth of OurGrid as possible.

3.1 Calculating the Local Reputation for a Peer

In the Network of Favors, a peer A calculates $r_A(B)$, the local reputation of peer B, using just two pieces of information: the value of favors A has received from B, and the value of favors B has received from A. Let $v(A, B)$ be the total value of the processing power donated from peer A to peer B over the past history of the system. We want $r_A(B)$ to be a function of $v(A, B)$ and $v(B, A)$, and we want the value of this function to increase when B does a favor for A, to decrease when A does a favor for B, and to be zero if A has never interacted with B.

The simplest function of $v(A, B)$ and $v(B, A)$ that satisfies these conditions is:

$$r_A(B) = v(B, A) - v(A, B) \tag{1}$$

In a previous work [3] we have shown that in simulations of this very simple autonomous reputation mechanism, the emergent behaviour of the community is that the peers who contribute more than they consume are prioritized. However, we did not consider the problem of malicious identity-changing. We will see in this paper that using equation 1 as reputation function makes the system vulnerable by identity-changing attacks.

A simple and effective solution to this problem, inspired by Yamagishi and Masuda's experiments with online auction markets [19], is to require the value of $r_A(B)$ to always be greater than or equal to zero, and zero for newcomers. This gives us the slightly more sophisticated function:

$$r_A(B) = max\{0, v(B, A) - v(A, B)\} \tag{2}$$

We shall see that the use of this function makes the OurGrid community robust to ID-changing.

Using a non-negative reputation function makes it possible to avoid prioritizing malicious ID-changing peers over collaborating peers who have consumed more resources than they have contributed. However, under this new reputation function a collaborator A cannot distinguish between a malicious ID-changing peer who never donates any resources and a collaborating peer B that has donated resources to A in the past but consumed at least the same amount of resources from A. To distinguish between these types of peers, we introduce another term in the reputation function $r_A(B)$, (we call it a history term), which reflects for peer A the history of its donations from peer B. To avoid creating a difference between the reputations of long-known peers and newcomers that is too high, and therefore too costly for a newcomer to overcome, we use a sublinear function of $v(B,A)$ as the history term in $r_A(B)$: for example

$$r_A(B) = max\{0, v(B,A) - v(A,B) + log(v(B,A))\} \quad (3)$$

or

$$r_A(B) = max\{0, v(B,A) - v(A,B) + \sqrt{v(B,A)}\} \quad (4)$$

For these functions, there is a relatively large difference in the history term between peers who have not donated to A at all and peers who have donated a little, but not much difference between two peers who both have long histories of reciprocating donations from A. This corresponds to intuition on how the relative values that people attach to favors varies with the amount of past interaction with the person granting these favors. Since the history terms take large positive values for large values of $v(B,A)$, they can make it possible to identify a collaborator even if the collaborator has consumed more resources than it has donated, provided that it has donated enough in the past.

In order to calculate $r_A(B)$, we assume that A has reliable information about $v(B,A)$ and $v(A,B)$, the value of favors received from and provided to B. Specifically, we assume that A can both (i) measure the value of a favor done by B for A; and (ii) verify that the work done was valid, i.e. that the data returned was not bogus. These assumptions are no stronger than the assumptions made for decentralized reputation schemes. To ensure the integrity of the information, A can use replication to both verify that the work was valid and that the value of the work was as reported by B. A detailed study of this approach applied to voluntary computing called *credibility-based fault tolerance* is presented by Sarmenta [16]. Using this scheme, a peer replicates each task on different service providers until at least a predetermined number of returned results is equal. Also, small probe tasks can be used periodically to verify a resource donator's correctness. By acting correctly, a donator gains credibility in a consumer's view, and the consumer gains confidence about its results. In OurGrid, we intend to implement this credibility-based mechanism both to check for sabotage and to verify other peers' informed accounting. Note that implementing sabotage tolerance for returned results is necessary in any resource sharing system.

Another possible attack is impersonation of peers with high reputation. This can be addressed through pairwise public key cryptography without requiring a shared cryptographic infrastructure (a certification hierarchy, system-wide revocation processes, etc). A peer about to interact with a peer that it has not interacted with before picks a new public key/private key pair from a large key space, and sends the public key to the new peer. Peers can check the identity of a requester by sending a challenge encoded in the appropriate key.

An important security issue that does not involve distortion of the reputation system is the potential use of the grid to launch denial-of-service attacks on donating peers or on some other service. In OurGrid this is addressed by ensuring that the work done on behalf of the requesting peer is carried out in a sandbox with restricted access to the underlying machine, and no network access.

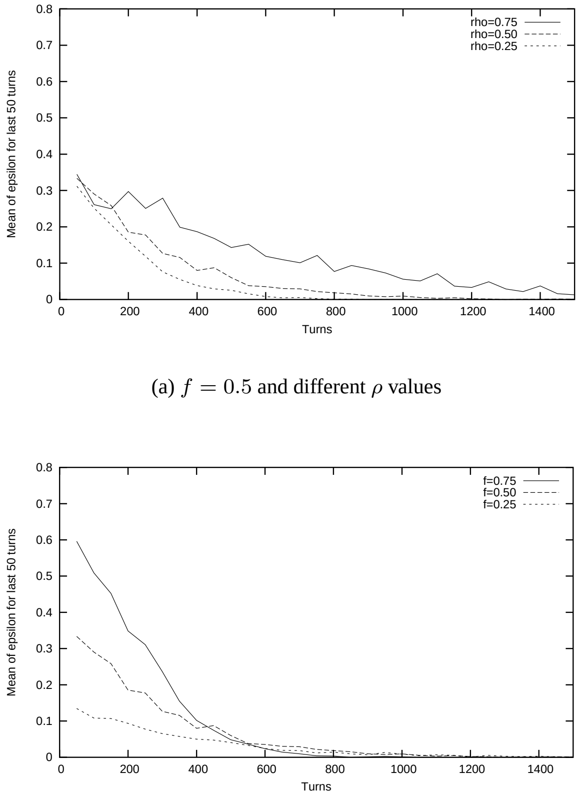
3.2 Evaluating the Network of Favors

This section describes the results of some simulations that show that the autonomous reputation scheme used in OurGrid is effective at distinguishing collaborators from free riders and promotes equitable resource sharing. We simulated the effects of all four of the reputation functions given in Subsection 3.1. The results for the two reputation functions with history terms (Equations 3 and 4) were very similar, so we will not report those for the function given in Equation 4.

We start by showing that even the very simplest reputation function — the one given by Equation 1 — makes the amount of resources donated to fixed-identity free riders very small indeed. After this, we introduce the case when a free rider changes identity by leaving the community and returning as a newcomer. In this scenario, the simplest reputation function cannot differentiate collaborators from free riders. We then show that the non-negative reputation schemes can successfully deal with this problem. Finally, we show that the reputation schemes with history terms have enhanced performance.

Our simulation scenario is a community of 100 peers that, in a time line divided in turns, share their resources. On each turn, each of the peers may be in consumer state with the same probability ρ. Of the hundred peers, $(1-f).100$ are collaborators and $f.100$ are free riders. When not in consumer state, each collaborator donates all its resources to one peer chosen among the consumers in the current turn according to their local reputation. The free riders, on the other hand, never donate. When not consuming, they go idle.

Recall that if a peer donates resources at time t, it will donate them to a free rider with probability $\epsilon(t)$. This probability can be estimated by measuring the proportion of the available resources that were consumed by the free riders in the simulation. Figure 1 shows this measurement for the simulation of a 100-peer community where $f = 0.5$ and $\rho = 0.5$. As there is a wide variation in the measured values from turn to turn, we have used the value averaged over



(a) $f = 0.5$ and different ρ values

(b) $\rho = 0.5$ and different f values

Figure 1. Measurement of $\epsilon(t)$ for different values of f and ρ, where all 100 peers use $r_A(B) = v(B, A) - v(A, B)$ as reputation function

the last 50 turns in the graph. All peers are using the simple balance of the favors exchanged with other peers (as in Equation 1) as their reputation functions. As time advances, the community identifies the free riders, and the probability of free riders getting resources becomes very small.

Note that the Network of Favors does require some time to identify the free riders, and so might not work well in a very dynamic resource-sharing network with many newcomers, such as the file distribution system BitTorrent [9]. However, incentives for cooperation can work even in BitTorrent, and we expect CPU-sharing grids to be much less volatile than this.

Figures 1(a) and 1(b) show the behaviour of the reputation system for varying values of ρ and f. These parameters affect the time the system takes to reach the steady state where the free riders are all identified and $\epsilon(t)$ is very small, and the early values of $\epsilon(t)$ before this state is reached, respectively. The time needed to reach the steady state is proportional to ρ. This happens because the community distinguishes a collaborator when it donates, and high values of ρ indicate that collaborators donate less frequently. In other words, the closer ρ is to 1, the more similar the behaviour of collaborators is to that of free riders, and the longer it takes for the community to determine that a peer is a collaborator.

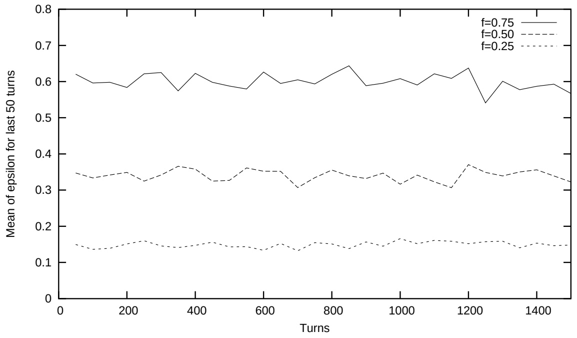


Figure 2. Measurement of $\epsilon(t)$ in a 100-peer community with $\rho = 0.5$ and different f values, but where all free riders are ID-changers, when all peers use the $r_A(B) = v(B, A) - v(A, B)$ as reputation function

On the other hand, for larger values of f, the early values of $\epsilon(t)$ are larger, because if a collaborator donates a resource to a peer with whom the collaborator has not previously had any interactions, the probability that this peer is a free rider is large for large f. However the value of f does not appear to significantly affect the time that the system takes to identify the free riders.

As the second step, we introduce another type of peer in the system, the *ID-changer*. This type of peer is a free rider that assumes a new identity on every turn, making it impossible for the community to keep track of its consumption. In Figure 2 we show how changing the 50 free riders with stable ID in the community of Figure 1(b) ($n = 100$, $\rho = 0.5$ and varying f) into ID-changers alters the emergent behaviour of the system.

As can be seen, the capacity to distinguish the free riders in the community greatly decreases, and $\epsilon(t)$ becomes close to f, which means that the probability that a donating peer selects a free rider as recipient is close to the probability that a peer selected at random is a free rider: the reputation information gives no significant help to the donating peer in distinguishing ID-changers from collaborators.

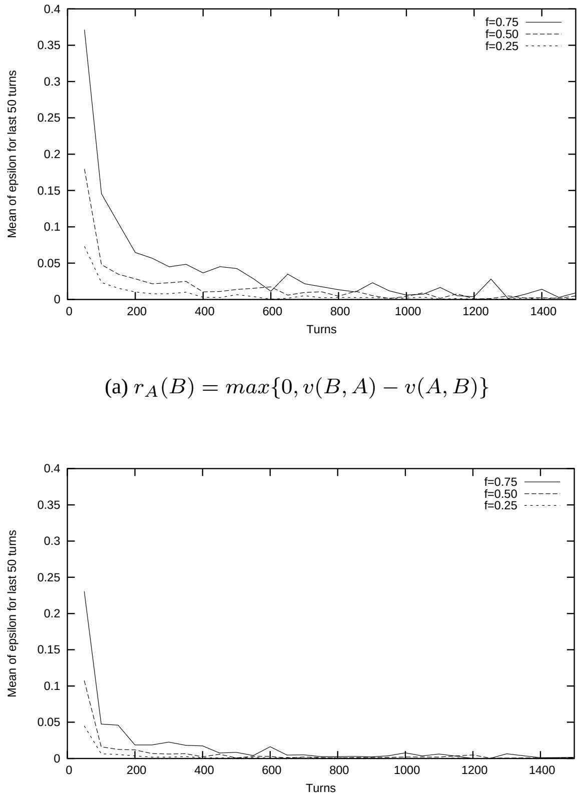
Figure 3 shows the same scenario as Figures 1 and 2 ($n = 100$, $f(t) = 0.5$ and $\rho = 0.5$), but in a community where the collaborators use a non-negative reputation function with and without a history term — to be precise, the reputation functions given by Equations 2 and 3. Figure 3(a) illustrates how the use of a non-negative reputation function improves the robustness of the community to the ID-changing behaviour. Adding the history term $log(v(B, A))$ further improves the ability of collaborators to identify each other, as can be seen in Figure 3(b).

Another interesting effect of using non-negative reputations is that ρ does not significantly affect the behaviour of ϵ in communities that use this kind of reputation. We believe that this happens because, in contrast to systems that use positive and negative reputations, free riders (and ID-changers) cannot have a reputation that is higher than that of a collaborator, and thus collaborators are more easily differentiated. Moreover, all it takes for a provider to not donate to free riders in a turn is that the provider identifies one collaborator among the consumers. This condition seems to be easily satisfied for any value of ρ.

In our simulations we have assumed that peers do not change their strategies from collaborator to free rider, or from free rider to collaborator. If they did change their strategies, the value of f would not be fixed, but would vary according to the number of free riders. Nevertheless, in all the scenarios we simulated with the three non-negative reputation functions, the measured value of $\epsilon(t)$ remained under $f/3$ after turn 100; so that for fixed f and $v < 2/3$, we have $\epsilon(t) < (1 - v).f$ for all $t > 100$. The analysis in Section 2 implies that in that case, free riders have lower expected utility than collaborators after turn 100. Since OurGrid is very lightweight and a peer can preempt a guest task at any moment, we expect v to be close to zero (and certainly smaller than 2/3). After higher numbers of turns there is an advantage to collaborators for values of v even greater than 2/3.

Now consider what happens using the same reputation functions if peers do change strategies to maximize to their expected utility. It might take slightly longer for the system to identify a peer as a collaborator if the peer had originally been a free rider. However, allowing for this, at some time after the first 100 turns in such a system we expect $\epsilon(t)$ to be less than $(1 - v).f(t)$ and to remain less than $(1 - v).f(t)$ for all subsequent turns, since this holds after 100 turns for all values of f in our simulations with fixed f. It follows from Section 2 that free riding dies out.

We have verified through simulations that the amount of resources that a collaborator receives divided by the amount it donates (denoted FR) is approximately 1. Figures 4 and 5 illustrate this for communities in which the amount a peer donates has a uniform distribution $U(1, 19)$. The cost of donating a resource is smaller than the utility gained by receiving it. It is therefore in the interest of peers to donate the largest amount of resources they can.



(a) $r_A(B) = max\{0, v(B, A) - v(A, B)\}$

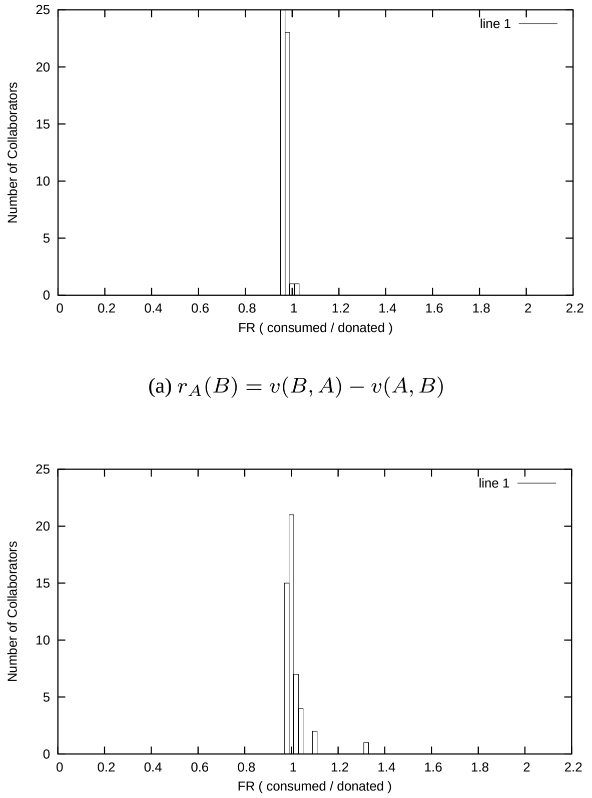
(b) $r_A(B) = max\{0, v(B, A) - v(A, B) + log(v(B, A))\}$

Figure 3. Measurement of $\epsilon(t)$ for two 100-peer communities with $\rho = 0.5$ and different f values. In the first community, peers use a simple non-negative reputation function. In the second they use a non-negative reputation function with history term $log(v(B, A)$, achieving a better performance.

4 Related Work

4.1 Peer-to-Peer Grids

As mentioned in the introduction, efforts are being made in the development of peer-to-peer grids as an alternative to the complexity of currency-based grids. The Condor [4] and Triana [18] projects have proposed peer-to-peer grids, but have not considered the problem of providing incentives for resource donation, relying solely on the altruism of the system's participants. Chun et al. propose an architecture for secure resource peering based on ticket exchange [7]. This architecture, however, assumes a shared cryptographic



(a) $r_A(B) = v(B, A) - v(A, B)$

(b) $r_A(B) = max\{0, v(B, A) - v(A, B)\}$

Figure 4. FR measured for all collaborators in two 100-peer communities where $f = 0.5$ and $\rho = 0.5$, after 3000 turns. In (a), peers use the positive and negative reputation function. In (b), they use a simple non-negative reputation function. In both FR approaches 1.

infrastructure and the establishment of relations of trust between peers to allow resource access.

We intend, with OurGrid [3], to provide a solution that is lighter and simpler to deploy in the context of Bag-of-Tasks applications. We believe that this simplicity will be the key property in enabling wide adoption.

4.2 Peer-to-Peer Reputation Schemes

A reputation scheme for a peer-to-peer system is a way of recording information about past behaviour of peers, for use as a guide to other peers. The information may be derived from objective facts, or the subjective impressions recorded by other peers, or a combination of these.

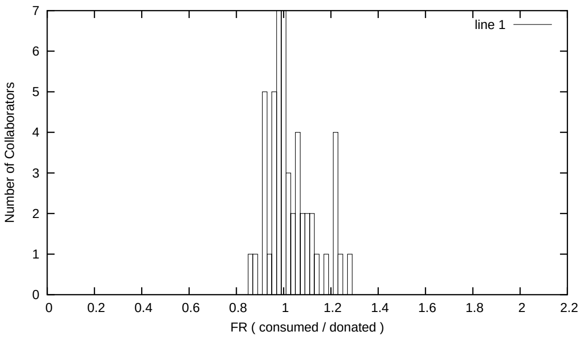


Figure 5. FR measured for all collaborators in a 100-peer community where $f = 0.5$ and $\rho = 0.5$, after 3000 turns. All peers use a non-negative reputation function with $log(v(B, A)$ as a history term.

For decentralized peer-to-peer systems, it makes sense to use distributed reputation schemes, in which the reputation information is distributed through different parts of the system. For example, in P2PRep [10] each peer stores information about their own interactions with other peers, and in EigenRep [12] each peer stores local reputation values and in addition random peers store global values derived from multiple local values. In a distributed reputation scheme, a peer can retrieve all the reputation information from the system concerning a given peer, using a retrieval protocol.

A challenging issue that a retrieval protocol must deal with is guaranteeing that the information gathered about peers is reliable, as malicious peers may tamper with the information they store. To assure the reliability of this information, P2PRep relies on voting for gathering opinions about a peer, heuristics to find clusters of potential malicious voters, and on a shared cryptographic infrastructure to verify the identities of the peers involved in a transaction. Alternatively, in EigenRep some replicated mother peers compute and store a global reputation value for a peer. The mother peers find the peers they must keep track of, and are found by peers who need information, through a distributed hash table.

In contrast, in OurGrid we circumvent the need to provide such guarantees by not aggregating a global reputation value for a peer. Instead, peers only use reputation information involving peer-to-peer interactions in which they themselves have participated. This information is stored locally by the peer, so is quick to retrieve. The reputation of a given peer will in general be different in the eyes of different peers, based on their own past interactions with the peer, and there is no attempt to create a global assessment. There is therefore no need for mechanisms to ensure the integrity of information received from other peers about their interactions with third parties, such as a shared cryptographic infrastructure or a specialized storage infrastructure. This allows the reputation scheme to be very lightweight. The authors of the Free Haven peer-to-peer publishing system considered the idea of not aggregating global reputations, but rejected it as leaving too much opportunity to malicious newcomers [11]. We have seen that in the OurGrid context, local reputations can give sufficient protection against these.

5 Conclusions and Future Work

We have shown that an autonomous reputation scheme can be sufficient to promote equitable sharing of resources in OurGrid, a peer-to-peer community of eager consumers. In particular, it can discourage free riding, and successfully deal with free riders who change identity to try to fool the system. Our scheme is very lightweight and does not require centralized storage or a shared cryptographic infrastructure. The only implementation issue that we have identified as potentially imposing difficulties for autonomous reputation schemes is sabotage tolerance, which is in fact an issue for any resource sharing system.

Our analysis assumed that the system exhibited eager consumption. However, we suspect that autonomous reputation schemes may also work under resource contention, which is a strictly weaker condition than eager consumption. If contention levels of peers offering the most popular services are high, this may be an incentive for peers to build up a good reputation, even if there is a limit to the amount of resources they can profitably consume. We intend to investigate further the circumstances under which our scheme remains effective.

We have also investigated how to calculate the local reputation in an autonomous reputation scheme. We have shown that using non-negative reputation functions makes the system robust to malicious identity-changing, and that adding a sublinear history term can improve further the system's ability to marginalize free riders. However, further research is needed on the role of the reputation function in the formation of long-term trust relationships between peers.

Another issue for investigation is the use of local subgroups of peers, where peers have priority access to resources owned within the same subgroup. Subgroups may also pool some reputation information.

Future work also includes finishing the deployment of OurGrid in a grid called Pauá, comprising over 220 machines in seven Brazilian cities, which is expected to be complete by the end of April 2004.

Acknowledgement

This work has been partially funded by CNPq/Brazil (grants 302317/2003-1 and 300646/1996-8).

References

[1] D. Abramson, R. Buyya, and J. Giddy. A computational economy for Grid computing and its implementation in the Nimrod-G resource broker. *Future Generation Computer Systems (FGCS) Journal*, 18:1061–1074, 2002.

[2] E. Adar and B. A. Huberman. Free riding on Gnutella. *First Monday*, 5(10), 2000. http://www.firstmonday.dk/.

[3] N. Andrade, W. Cirne, F. Brasileiro, and P. Roisenberg. OurGrid: An approach to easily assemble grids with equitable resource sharing. In *Proceedings of the 9th Workshop on Job Scheduling Strategies for Parallel Processing*, June 2003.

[4] A. R. Butt, R. Zhang, and Y. C. Hu. A self-organizing flock of condors. In *Proceedings of Supercomputing 2003*, November 2003.

[5] R. Buyya, D. Abramson, and J. Giddy. An economy driven resource management architecture for computational power grids. In *International Conference on Parallel and Distributed Processing Techniques and Applications*, 2000. http://citeseer.nj.nec.com/299926.html.

[6] 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)*, Beijing, China, 2000. IEEE Computer Society Press.

[7] B. N. Chun, Y. Fu, and A. Vahdat. Bootstrapping a distributed computational economy with peer-to-peer bartering. In *Proceedings of the 1st Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[8] W. Cirne, F. Brasileiro, J. Sauvé, N. Andrade, D. Paranhos, E. Santos-Neto, R. Medeiros, and F. Silva. Grid computing for Bag-of-Tasks applications. In *Proceedings of the I3E2003*, September 2003.

[9] B. Cohen. Incentives build robustness in BitTorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[10] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Managing and sharing servents' reputations in P2P systems. *IEEE Transactions on Data and Knowledge Engineering*, 15(4):840–854, July/August 2003.

[11] R. Dingledine, N. Mathewson, and P. Syverson. Reputation in p2p anonymity systems. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems*, June 2003.

[12] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. EigenRep: Reputation management in P2P networks. In *Twelth International World Wide Web Conference*, Budapest, Hungary, May 2003. Preprint at http://dbpubs.stanford.edu:8090/pub/2002-56.

[13] S. Newhouse, J. MacLaren, and K. Keahey. Grid Economic Services Architecture Working Group. http://www.doc.ic.ac.uk/˜sjn5/GGF/gesa-wg.html.

[14] D. Paranhos, W. Cirne, and F. Brasileiro. Trading cycles for information: Using replication to schedule bag-of-tasks applications on computational grids. In *Proceedings of the Euro-Par 2003: International Conference on Parallel and Distributed Computing*, 2003.

[15] M. Ripeanu and I. Foster. Mapping the Gnutella network: Macroscopic properties of large-scale peer-to-peer systems. In *First International Workshop on Peer-to-Peer Systems (IPTPS)*, 2002.

[16] L. F. G. Sarmenta. Sabotage-tolerance mechanisms for volunteer computing systems. *Future Generation Computer Systems*, 18(4):561–572, 2002.

[17] S. Saroiu, P. K. Gummadi, and S. D. Gribble. A measurement study of peer-to-peer file sharing systems. In *Proceedings of Multimedia Computing and Networking 2002 (MMCN ’02)*, San Jose, CA, USA, January 2002. http://citeseer.nj.nec.com/saroiu02measurement.html.

[18] I. Taylor, M. Shields, I. Wang, and R. Philp. Distributed P2P computing within Triana: A galaxy visualization test case. In *Proceedings of IPDPS’2003*, Abril 2003.

[19] T. Yamagishi and M. Matsuda. Improving the lemons market with a reputation system: An experimental study of internet auctioning, May 2002. http://joi.ito.com/archives/papers/Yamaghishi_ASQ1.pdf.