Automatic Grid Assembly by Promoting Collaboration in Peer-to-Peer Grids[♦]

Nazareno Andrade[1], Francisco Brasileiro[1], Walfredo Cirne[1]
Miranda Mowbray
Enterprise Systems and Storage Laboratory
HP Laboratories Bristol
HPL-2005-22(R.1)
July 25, 2007*

distributed systems,
peer-to-peer,
grid computing,
incentive
mechanisms

Currently, most computational grids (systems allowing transparent sharing of computing resources across organizational boundaries) are assembled using human negotiation. This procedure does not scale well, and is too inflexible to allow for large open grids. Peer-to-peer grids present an alternative way to build grids with many sites. However, to actually assemble a large grid, peers must have an incentive to provide resources to the system. In this paper we present an incentive mechanism called the *Network of Favors*, which makes it in the interest of each participating peer to contribute its spare resources. We show through simulations with up to 10,000 peers and experiments with software implementing the mechanism in a deployed system that the Network of Favors promotes collaboration in a simple, robust and scalable fashion. We also discuss experiences of using OurGrid, a grid based on this mechanism.

* Internal Accession Date Only
[1] Universidade Federal de Campina Grande, Brazil
[♦] Journal of Parallel and Distributed Computing, Vol. 68, Issue 8, 2007

Approved for External Publication

Automatic Grid Assembly by Promoting Collaboration in Peer–to–Peer Grids

Nazareno Andrade   Francisco Brasileiro   Walfredo Cirne

Universidade Federal de Campina Grande
Av. Aprígio Veloso, 882 - 58.109-970, Campina Grande, Brazil
Tel:+55 83 3310 1365 Ext. 30 Fax:+55 83 3310 1498

Miranda Mowbray

Hewlett-Packard Laboratories Bristol
Filton Rd, Stoke Gifford, Bristol BS34 8QZ, UK

NOTICE: this is the authors' version of a work that was accepted for publication in Journal of Parallel and Distributed Computing. Changes resulting from the publishing processes, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version was subsequently published in JPDC [Volume 67, Issue 8, (August 2007)], DOI 10.1016/j.jpdc.2007.04.011

Preprint submitted to Elsevier Science *13 July 2007*

Abstract

Currently, most computational grids (systems allowing transparent sharing of computing resources across organizational boundaries) are assembled using human negotiation. This procedure does not scale well, and is too inflexible to allow for large open grids. Peer-to-peer grids present an alternative way to build grids with many sites. However, to actually assemble a large grid, peers must have an incentive to provide resources to the system. In this paper we present an incentive mechanism called the *Network of Favors*, which makes it in the interest of each participating peer to contribute its spare resources. We show through simulations with up to 10,000 peers and experiments with software implementing the mechanism in a deployed system that the Network of Favors promotes collaboration in a simple, robust and scalable fashion. We also discuss experiences of using OurGrid, a grid based on this mechanism.

Key words: Distributed Systems, Peer-to-Peer, Grid Computing, Incentive Mechanisms

---

1 Introduction

Computational grids—systems enabling the transparent sharing of computing resources across organizational boundaries—promise to make unprecedented amounts of resources available for parallel applications at lower cost than traditional alternatives based on parallel supercomputers. Much research has been done in recent years to realize this vision, and the first grids are now in use (see eg. [1], [2], [3], [4], [5]).

Although these grids are useful for many users and applications, they are somewhat limited in scale. Typically, the grid is created and managed off-line by human negotiation between the owners of the resources which form the grid. For a new site to join the grid, it is necessary to negotiate terms of participation with those who manage it. This limits how far and how fast the grid can grow. For example, in 2005 CERN's LCG [1] was the world's largest grid initiative with more than 10,000 CPUs; yet it had only around 100 sites.

Peer-to-peer (P2P) grids [6], [7], [8], [9], [10], [11], [12] present an alternative way to build grid infrastructures with a larger number of sites, or peers—in this paper we assume that each peer in the P2P grid represents a site in

---

Email addresses: `nazareno@dsc.ufcg.edu.br` (Nazareno Andrade), `fubica@dsc.ufcg.edu.br` (Francisco Brasileiro), `walfredo@dsc.ufcg.edu.br` (Walfredo Cirne), `miranda.mowbray@hp.com` (Miranda Mowbray).

a different administrative domain. These grids sacrifice support for complex sharing policies in exchange for ease of deployment, management, use, and growth [13], using P2P technology. This ease is based on three characteristics: i) they are independent from any centralized infrastructure or trusted authority; ii) they are essentially free to join; and, iii) they support only a limited class of applications.

However, setting up a P2P grid infrastructure has its own complications. Although there are many proposals to solve the scalability challenges for P2P systems (discovery, routing, etc.), these alone are not enough to guarantee growth. In general, not all participants in a P2P grid know each other. It cannot be assumed that these participants are unselfish or trustworthy. If sharing is to to occur, there must be incentives for peers to contribute resources to the grid. Otherwise, peers have an economic incentive to become *free riders*, ie. only consuming resources and not donating back to the P2P grid, reducing the resources available for donation in the system and diminishing the grid's utility. In fact, in most P2P file-sharing networks the majority of users free ride [14], [15], [16]. In this paper we consider how to address this problem, promoting cooperation in P2P computational grids.

Our approach to providing an incentive for donation is to use pair-wise reciprocation between peers [6], [17], [18], [19]. A peer decides to whom to donate its spare resources based on *local history*, that is, information gathered from its past direct interactions with other peers. Using only local history obviates the need to ensure the integrity of second-hand information. It is well known that incentive mechanisms using local history are not effective in all settings [20], [21]. There need to be frequent interactions between pairs of peers for local history to be useful when deciding which peers to reward. However, as we will show, P2P computational grids have characteristics which promote frequent interactions, even when the number of peers is large.

In this paper we show that, given these characteristics, it is possible to provide incentives for resource provision in fairly large P2P grids (at least in the order of 10,000 peers) using pair-wise reciprocation based on local history. Under the incentive mechanism that we present, the emergent behavior of the system is that peers which contribute to the grid are prioritized when requesting resources. Also, in the long run, the more value a peer donates to the system (resulting from the donation of its spare resources), the more value it receives back. The mechanism, named the *Network of Favors*, performs nearly as efficiently as an oracular mechanism which never prioritizes free riders over *collaborator* peers (ie. peers that are not free riders). Moreover, the mechanism is particularly lightweight and robust against malicious peers, making it very suitable for P2P grids.

Our initial motivation for this work was the desire to build a grid that could pool the computational resources of the thousands of small and medium-sized scientific research labs around the world that have unserved computational demands [8]. The researchers in these labs typically do not use their computers all the time. When carrying out research they tend to alternate between executing computing intensive jobs, and analyzing the results. While they are analyzing the results their computing resources are mostly idle. (Users of Enterprise Desktop Grids have behaved in a similar fashion [22].) In the open, free-to-join, cooperative grid that we envisaged, labs would donate the use of their idle computational resources in exchange for accessing other labs' idle resources when needed, thus gaining faster turnaround times for their computing jobs.

The rest of this paper is structured as follows. In Section 2 we discuss in more detail various approaches to the problem of providing incentives for resource sharing in grids. In Sections 3 and 4 we present the Network of Favors and evaluate its effectiveness at encouraging the contribution of resources to the grid. In Section 5 we present an experiment with the software developed for a P2P grid based on the Network of Favors, and discuss lessons learned in our implementation, deployment and use of the grid. The discussion is based on our ongoing experience with OurGrid (available at http://www.ourgrid.org/), a free-to-join P2P grid which has been in production since December 2004. Section 6 concludes the paper.

The main contributions of this paper are the examination in Subsection 3.1 of how the Network of Favors ensures repeated interactions, which are necessary for any pair-wise reciprocation mechanism to work in a P2P system; the results of simulations for 1,000 and 10,000 peers in Subsection 4.3, which are key to justifying the utility and scope of the Network of Favors; and the experiment with OurGrid software in Subsection 5.2, which shows how well the implemented system provides incentives for collaboration.

2 Related Work

Several proposals for P2P grids exist. Butt et al. [7] have proposed organizing Condor pools in a P2P network with no central coordination, and XtremWeb [9] envisages the creation of a P2P network of collaborators to use a shared volunteer computing platform. Triana [12], Cluster Computing on the Fly [10], and P3 [11] are other examples. However, none of these address the issue of providing incentives for the donation of resources.

The incentive mechanisms that have been proposed to deal with this problem in grids and P2P systems in general can be broadly classified into market-based and reciprocation mechanisms.

2.1 Market-based mechanisms

Markets are a well-known mechanism for regulating access to resources by selfish agents. Abramson et al. [23] proposed a framework for a grid market in which brokers negotiate with resource providers using a choice of market mechanisms. Buyya and Vazhkudai [24] proposed a P2P network in which peers interact with each other to obtain computing power and pay real currency for it. The Grid Economic Services Architecture (GESA) Working Group [25] aims at defining open standards for grid market services. Tycoon [26] time-shares grid resources according to the result of auctions among consumers.

All these proposals presume the existence of a currency distribution system and banking services, which inherently require trusted institutions. Therefore these proposals do not suit the scenario we consider for P2P grids. Furthermore, market-based systems can be complex to use [27]. In a market, one has to estimate the application resource consumption, plan how to spend a constrained budget, and deal with prices that change over time.

2.2 Reciprocation mechanisms

An alternative approach is to make the system reward its participants based on how much they have contributed to the system in the past. Naturally, for the system to reciprocate the past contributions of peers, it needs a way to store information about peers' past behavior.

Under reputation-based incentive mechanisms, peers decide how to allocate their resources according to the reputations of the requesting peers. A peer's reputation is an aggregation of the opinion of other peers in the system about the peer. P2PRep is a reputation mechanism for Gnutella [28] which aggregates opinions based on a polling protocol. EigenTrust [29] uses eigenvectors of normalized trust scores. Karma [30] tracks the resources consumed and contributed by each peer. P2PRep defends against malicious peers who try to defraud the voting by contacting some voting peers directly to check their votes have been correctly reported, and by looking for suspicious clusters of votes from similar IP addresses. EigenTrust and Karma rely on specialized secure score management systems.

Our mechanism, the Network of Favors, differs from these as it does not use any aggregation, relying instead only on the local information available to each peer. Thus, it can be particularly lightweight, and operate securely in the absence of advanced infrastructure. It does not need certified identities, trusted third parties, or score management systems.

The idea of using pair-wise reciprocation has been explored in other P2P systems. GNUnet and BitTorrent were independently developed at the same time as the Network of Favors. The former uses an economic model to allocate bandwidth resources in an anonymous file-sharing network [19], while the latter uses pair-wise reciprocation to allocate the bandwidth of peers taking part in a swarm download [17]. The eMule file sharing client [18] uses the same principle to decrease cooperative peers' wait time for file downloads. These mechanisms are similar to the Network of Favors, but the resource shared is bandwidth or data, rather than computing cycles. To the best of our knowledge, our work is the first to extensively analyze the behavior and performance of such mechanisms for sharing computing cycles.

Another related proposal is that of Chun et al. [31]. They propose a P2P bartering framework based on Sharp [32] that enables peers to exchange claims on resources. In [31] resources are bartered in within specified time intervals. A peer P only gives its resource to a peer Q, if there is an agreement that Q will also give the resources P requires during some specified future time interval. In the absence of such an agreement, P does not give resources to Q. In the Network of Favors, in contrast, resources are donated as a favor that it is likely to be eventually reciprocated. We attribute this difference to the distinct original targets of the two approaches. The Network of Favors was designed to cater for resource-intensive applications, for which there is no point in giving away resources when they are needed locally. Chun et al.'s proposal targets applications which benefit more from having multiple network vantage points. These applications will typically want to trade some of their local resources for remote ones, even when they are actively using local resources.

Finally, Ma *et al.* [33], [34], [35] propose a resource distribution mechanism to encourage peers to share. Their mechanism has some very nice properties, including guaranteeing Pareto-optimal allocation of resources. However it relies on a distributed auditing authority whose cryptographic signatures peers can check, to ensure that the total amount that each peer contributes to the system is correctly known by all the peers.

It is worth noting that reciprocation mechanisms are less flexible than market-based ones. For instance, in the Network of Favors, the "price" of the work performed by a resource is independent of how quickly or slowly the work was performed, and of the system conditions (eg. whether there was contention for the resource used). This is because to have varying prices, it is necessary to have a currency. If having varying prices is crucial for a system, then a market-based mechanism would be a better option. However, in the setting we target, simplicity of the deployment outweighs the flexibility gained by market-based mechanisms.

3 The Network of Favors

In this section we define the Network of Favors and evaluate its effectiveness at encouraging resource donation.

Let us assume that any peer P can autonomously compute the value of the useful computing power that it consumes from another peer Q and the value of the useful computing power that it donates to Q. We will call such a donation a *favor*.

The local score of peer Q in the eyes of peer P is denoted by $s_P(Q)$. If P and Q have never interacted, $s_P(Q)$ and $s_Q(P)$ are zero. If resources of value v are donated from P to Q, and before the donation $s_P(Q) = x$ and $s_Q(P) = y$, then after the donation the new local scores satisfy

$$s_P(Q) = \max(0, x - v), \;\; s_Q(P) = y + v. \tag{1}$$

The only way that Q can increase its score in the eyes of another peer P is by donating resources to P. The fact that the Network of Favors uses only local information prevents it from taking wrong decisions driven by information provided by malicious peers. The value of $s_P(Q)$ is an upper bound on the amount of favors that P owes to Q.

Resource allocation is performed taking into account the local scores maintained by each peer. Whenever P has resources available for donation, it allocates these resources to the requesting peers, giving highest priority to satisfying the requests of those with the highest local scores. If all requesters have a score equal to zero, P chooses recipients at random.

By donating their spare resources, collaborator peers increase their priority when requesting resources in the future. Free riders (which never donate resources), on the other hand, can only get a fraction of the resources that are not contended by collaborators with positive scores.

Note that scores are always greater than or equal to zero. This renders innocuous whitewashing attacks in which a free rider keeps presenting itself to the system as a new peer with a brand new identity. Since zero is the worst score any peer can have, a peer cannot gain by leaving the system and re-entering with a new identity. This use of non-negative scores was inspired by Yamagishi and Matsuda's reputation experiments [36].

No special bootstrap scheme is required for the Network of Favors to work. Newcomers start with a zero score. A given collaborator peer P, when choosing to whom donate its spare resources, treats newcomers in the same way as any other peer with a score equal to zero. These can be either the free riders or other collaborators which have consumed from P at least as much as they have donated to P in the past. Provided that no peer with a positive score is contending for P's spare resources, a newcomer has the same chance to get the donation as any other requesting peer.

3.1 Repeated Interactions and the Network of Favors

For any pair-wise reciprocation mechanism to work, peers that have interacted once must have a high probability of interacting again. Feldman et al. [20] and Lai et al. [21] have shown that when interactions between peers are infrequent (for example, because of asymmetry of interest or rapid population turnover), then incentive mechanisms for sharing based on local history do not scale well beyond 1,000 peers. Asymmetry of interest happens when peer P is interested in the resources of a peer Q, but Q is not interested in P's resources. A large peer population with rapid turnover makes repeated interactions between pairs of peers less likely, and therefore makes it harder for collaborators to be rewarded.

However, the Network of Favors is effective in P2P grids because in these systems, unlike with the file-sharing systems considered by Feldman, Lai et al., i) there is much more symmetry of interest; ii) interactions among peers follow a many-to-many pattern; iii) there is a relatively slow-changing population of collaborators; and iv) the calculation of local scores encourages repeated interactions.

Interest is much more symmetric in P2P grids than in file-sharing systems. A file-sharing system may share many thousands of files, of which only a few will be interesting to a peer, but there are not many different combinations of operating systems and processor architectures. Furthermore, virtualization by resource providers and the compilation of portable code or applications for different platforms by users make most computational resources interchangeable.

The many-to-many interactions in P2P grids make it more likely that two peers interact frequently than is the case in systems with one-to-one interactions, such as most file-sharing systems. A peer in a P2P grid typically interacts with a large number of other peers each time it requests resources, due to the large demand for computing power that characterizes high performance computational grid applications. Moreover, a peer providing resources does not need to allocate all its resources to a single requester, and can use the fact that it has several resources to signal to several peers at once that it is a collaborator.

We believe it is reasonable to assume a relatively slow-changing population of collaborators. This is because each peer of the grid is a service which will be managed by system administrators for one or more users in an organization. The frequency of new collaborators joining the grid will be slow compared to the frequency of peer interactions. As we show in this paper, a stable population of collaborators using the Network of Favors can manage to prioritize collaborating peers, even if a majority of the peers in the system at any time are free riders and the free riders have a very rapid turnover.

Finally, the way the local scores are calculated has the effect of encouraging repeated interactions between collaborators. If collaborators P and Q have ever interacted, it is guaranteed that $s_P(Q) + s_Q(P) > 0$. This implies that at least one of them will prioritize the other over free riders and peers with which it has not interacted.

4 Evaluation

In this section we model the Network of Favors, and examine the conditions under which it succeeds in disadvantaging free riders, and therefore, promoting collaboration. We first present an idealized model of a P2P grid and report an analytical result for this model on the situations in which a perfectly-informed pair-wise reciprocation mechanism can promote collaboration. We then use simulations with large numbers of peers to compare the performance of the Network of Favors with this ideal mechanism, under representative scenarios drawn from the analysis.

The system model and analysis for an ideal mechanism appeared in two conference papers [37], [38]. Simulations of the scenarios were reported in [38], but only for 100 peers. We here simulate the Network of Favors with 1,000 and 10,000 peers and discuss its scalability. Although these simulation results are similar to the results previously presented for 100 peers, they cannot be inferred from those. In fact, the effect of increasing the population is that the reciprocation mechanism becomes more effective. This is the opposite of what happens under other reciprocation mechanisms [20], [21]. The extension of the results to 10,000 peers is important as it assures that the Network of Favors can be used in a P2P grid for research labs, which is a scenario of great practical relevance.

We consider a P2P grid comprised of a set of collaborators and free riders. At a given time t, a peer can be either in consuming or in non-consuming state. When in non-consuming state, collaborators donate their resources, while free riders go idle. The protocol for donation of resources is that collaborators that are not in consuming state donate all the resources that they have available as long as there are peers in consuming state prepared to consume them. Peers with high scores in the eyes of a donator are given priority in its donations. Any resources left over, after all peers in consuming state have been donated the maximum amount of resources that they are prepared to accept, are not donated. The design parameters that we consider for the P2P system are:

- **Eagerness**. We assume that, for each peer, C is the maximum value of the utility that can be obtained by consuming resources from other peers in a unit time interval, when the peer is in consuming state.
- **Frequency of consumption**. We assume for simplicity that at a given time each peer has an independent probability ρ of being in consuming state. (We model differences in peers' workloads by differences in C.) Reciprocation mechanisms perform better under strong contention, and thus our assumption of independence is conservative, modeling the least favorable condition in the absence of data on demand correlation.
- **Value of donation**. When a collaborator is not in consuming state, it has resources of value D available to donate to the system.
- **Cost of donation**. The utility lost to the donator as a result of donation is a fraction v of the utility gained by the recipient as a result of the donation, with $0 < v < 1$. If resources are available for donation but are not donated, no utility cost associated with these resources is incurred by the resource owner.
- **Prevalence of free riding**. We write f for the proportion of all peers that are free riders. The other peers are collaborators.

The values of v and C/D are the same for all peers, but D (and hence C) can vary among different peers. The value of ρ is either the same for all peers, or is chosen independently for each peer from the same Uniform distribtion. We denote the average values of C, D, ρ by $\overline{C}, \overline{D}, \overline{\rho}$. We assume for our simulations that all parameter values are fixed over time. In our analysis however we allow f to vary over time, because we allow for the possibility that peers change their strategies over time in order to increase their expected utility. We also assume that the granularity of resources is low enough that a donating peer with at least as many spare resources as a consuming peer requests is able to give exactly the amount of resources requested.

We define the *advantage to collaborators* at time t as the expected long-term utility gain to a collaborator as a result of being in the system minus the expected utility gain to a free rider. This is a measure of how much free riding is discouraged at time t.

Consider an ideal pair-wise reciprocation mechanism with perfect information, which never prioritizes a free rider over a collaborator. Analysis of the system model shows that, under the assumption that the amount of resources requested and donated at a given time can be accurately approximated by their mean values, the advantage to collaborators is always positive under this mechanism if there is at least one free rider and $\overline{\rho} \cdot \overline{C} \geq (1 - \overline{\rho}) \cdot \overline{D}$. It follows that if this inequality holds, and peers choose whether or not to free ride in order to maximize their long-term expected utility, then eventually all peers will collaborate. On the other hand, analysis shows that if the inequality does not hold, then the ideal mechanism will not succeed in eliminating free riding from the system (and hence we cannot expect the Network of Favors to do so in this case). The estimate of the advantage to collaborators given by approximating the amounts requested and donated by their means is least accurate for parameter sets close to the borderline $\overline{\rho} \cdot \overline{C} = (1 - \overline{\rho}) \cdot \overline{D}$ between the two regions of parameter space. Further detail and proofs can be found in [38].

4.2 Simulation Method

In order to evaluate the effectiveness of the Network of Favors in deterring free riding, we picked some representative scenarios for the system parameters, and simulated the Network of Favors and the ideal mechanism for these scenarios. We used a home-brewed simulator that implements the model described in Subsection 4.1. The simulator includes some aspects of a real implementation (specifically, the fluctuations over time of amounts donated and requested) that are ignored by the analysis.

We initially chose the 54 scenarios in which the parameter values satisfy $D = 10$; $C \in \{D/10, D, 9D\}$; $\rho \in \{0.1, 0.5, 0.9\}$; $f \in \{0.25, 0.5, 0.75\}$; and $v \in \{0.1, 0.4\}$. In these initial scenarios the values of D (and hence C) and ρ did not vary between peers. We chose these parameter values to include both low and high realistic values, and to include some scenarios for which the parameter sets are on the borderline found by the analysis. We simulated these scenarios for populations of 10,000 peers.

As a second step, we introduced some new scenarios where D (and hence C) or ρ are different for different peers. We investigated the cases where either D or ρ are given by the uniform distributions $U(1, 19)$ or $U(0.1, 0.9)$, respectively. As simulations with 10,000 peers are very demanding on computing power, we used 1,000-peer populations for these scenarios.

In our simulations, the timeline is in turns, and at each turn each peer has an independent probability ρ of being in consuming state.

For the sake of simplicity, we assume in the simulations that a donating peer donates as much as possible to the requester with the largest local score, then to the requester with the second largest local score, and so on. If there are still resources to be donated and no more peers whose local scores are greater than zero, the peer randomly chooses requesters with scores equal to zero to donate its resources.

We calculated the advantage to collaborators as the mean utility of collaborators minus the mean utility of free riders divided by the number of turns we have simulated.

Our simulations are for fixed values of f. If peers change strategy in order to maximize their utility, f may vary over time. However, if we can show in the simulations for different fixed values of f that the two mechanisms give similar advantages to collaborators, we argue that it is reasonable to suppose that the effect over time of the Network of Favors on peers' willingness to free ride will be similar to that predicted in 4.1 for the ideal mechanism, with the prediction being least accurate when the parameter set lies on the borderline.

4.3 Simulation Results

In all scenarios simulated, the value calculated for the advantage to collaborators converged with a negligible margin of error after 2,000 turns.

For both incentive mechanisms, the advantage to collaborators in the simulations was positive for all 36 of the initial scenarios for which the analysis of an ideal mechanism had predicted that it would be positive. Moreover, for most of these scenarios, there was little difference between the values of the two incentive mechanisms; the difference was significant only in the 12 scenarios for which $C = D$ and $\rho = 0.5$ or $C = 9D$ and $\rho = 0.1$. These are all on the borderline found by the analysis. We detail the results for these scenarios in Table 1. The Network of Favors performs on average only 22% worse than the ideal mechanism on these 12 scenarios.

It is worth noting that, for both incentive mechanisms, the advantage to collaborators is positive for all the scenarios presented in Table 1. In the absence of an incentive mechanism, free riders would be better off than collaborators. We should also note that we are comparing an ideal incentive mechanism, which is not implementable, with the practical Network of Favors.

Table 1
Advantage to collaborators for the scenarios where $C = D$ and $\rho = 0.5$ or $C = 9D$ and $\rho = 0.1$

C	ρ	v	f	NoF	Ideal	Difference
D	0.5	0.1	0.25	3.52756	4.32169	12.25%
D	0.5	0.1	0.50	3.77000	4.38480	16.31%
D	0.5	0.1	0.75	3.77830	4.39351	16.28%
D	0.5	0.4	0.25	1.74116	2.81585	61.72%
D	0.5	0.4	0.50	2.27121	2.88190	26.89%
D	0.5	0.4	0.75	2.27799	2.89583	27.12%
$9D$	0.1	0.1	0.25	6.67030	7.56041	13.34%
$9D$	0.1	0.1	0.50	6.97965	7.75235	11.07%
$9D$	0.1	0.1	0.75	6.88675	7.77904	12.96%
$9D$	0.1	0.4	0.25	3.99047	4.87592	22.19%
$9D$	0.1	0.4	0.50	4.27503	5.05738	18.30%
$9D$	0.1	0.4	0.75	4.19039	5.07599	21.13%

Changing the distribution of D to $U(1, 19)$ had little effect. Changing the distribution of ρ to $U(0.1, 0.9)$ resulted (via the increased statistical fluctuations in the amount of resources donated) in an increased difference in the performance of the two incentive mechanisms in the scenarios where $C = D$, and made the advantage to collaborators negative under the Network of Favors in the scenario where $C = D$, $v = 0.4$, $\rho = U(0.1, 0.9)$ and $f = 0.25$. All these scenarios are on the borderline. In addition, the scenario in which the advantage to collaborators was negative has high donation cost and a relatively small proportion of free riders which share the resources not donated to collaborators.

Overall, the Network of Favors gave a positive advantage to collaborators in all but one of the scenarios for which the analysis predicted there was a chance to eliminate free riding, and for most of the scenarios there was very little difference between the measured advantage for collaborators in a system using the Network of Favors and in a system using the ideal incentive mechanism. In particular, the Network of Favors gave similar results to the ideal mechanism in all the scenarios for which $\overline{\rho} \cdot \overline{C} > (1 - \overline{\rho}) \cdot \overline{D}$.

The population size of 10,000 is two orders of magnitude greater than the current largest grids and was chosen based on the characteristics of the kind of grids we envisage will make best use of the Network of Favors [8]. By carrying out further simulations of the initial 54 scenarios with 1,000 peers, and comparing with our previously-published results for simulations of all the scenarios with 100 peers, we found that the Network of Favors behaves differently from other reciprocation mechanisms which do not scale well beyond 1,000 peers [20], [21]. We found that within the limits of scalability, the larger the population size, the better both the Network of Favors and the ideal mechanism performed. This is because under the ideal mechanism free riders are only able to consume resources which are not contended by any collaborator, while under the Network of Favors only the resources not contended by collaborators with scores greater than zero. The more peers there are, the more collaborators, and the more likely it is that a resource is contended by a collaborator.

A system using the Network of Favors requires some time to reach a steady state in which free riders are marginalized. In simulations of 1,000-peer systems with 750 free riders, we measured the percentage of resources which were donated to the free riders in the last 50 turns. We found that this percentage dropped steadily after the 50^{th} turn until stabilizing at very low levels after at most 1,000 turns.

5 The OurGrid System

We have implemented and evaluated the Network of Favors in OurGrid, a P2P grid that provides resources for Bag-of-Tasks (BoT) applications, ie. those parallel applications whose tasks are independent [39], [40]. BoT applications are especially well suited for execution on today's grids due to their large computation-to-communication ratio. Moreover, despite the simplicity of their structure, BoT applications are useful in a very wide variety of application areas, including data mining, brute-force search, parameter sweeps, meteorology, environmental engineering, simulations, computational biology, and computer imaging.

BoT applications have two characteristics which make them suitable for a system using the Network of Favors. The first is that they can benefit from the use of volatile resources, because the failure of one task does not affect another. Volatile resources are ones that may become unavailable at any moment. As the Network of Favors does not provide guarantees about how long a resource should be allocated to a given peer, the resources exchanged using it are volatile.

The second characteristic is that their users are often *eager consumers*, meaning that if they are in consuming mode and are offered additional computing power, they are likely to find profitable ways to use it. (In our system model, this corresponds to having very high values of C. As shown in 4, high values of C ensure that the Network of Favors is effective in promoting collaboration.) It is frequently the case that BoT applications are comprised of a great number of tasks, and that the amount of work done in a given time is therefore approximately proportional to the amount of resources available. John Reynders of Celera Corporation has stated that Celera could "certainly use any resources we can get our hands on" to perform their computations for drug discovery [41]. In rendering, the folk wisdom known as Blinn's law [42] states that if there is an increase in rendering speed, then the artist will increase the complexity of the frames to be rendered, to the point that the speed improvement is negated. Moreover, Paranhos et al. [43] and Santos-Neto et al. [44] have shown that the replication of tasks among the available resources improves the makespan of BoT applications running on grids. These arguments suggest that contention for resources is very likely in a grid used to run BoT applications. Resource contention increases the effectiveness of the Network of Favors.

In the rest of this section we describe our experience implementing, deploying and using the Network of Favors in OurGrid. In Subsection 5.1 we describe how we implemented the Network of Favors in OurGrid. We then present results from controlled experiments with the OurGrid software in Subsection 5.2 and our experience with the deployment of OurGrid in Subsection 5.3.

5.1 The Network of Favors in OurGrid

Each peer in OurGrid is a site that has grid users and a set of grid machines, which are the machines from the site made available to the grid. Whenever the grid machines owned by a peer are not required by its local users, the peer tries to use them to satisfy requests made by the other peers in the system. Local requests always have priority over remote ones. If a local resource which is allocated to another peer is requested by a local user, it is immediately pre-empted and reallocated to the local user.

As in any P2P network, a participant must discover which other peers can satisfy his demand, and propagate his request to them. Currently, peers use a simple point-to-point broadcast mechanism to propagate queries, but we intend to evolve the software to use a more scalable grid discovery mechanism such as NodeWiz [45, 46].

When collaborator peers receive remote requests and have idle resources, they allocate these resources using the Network of Favors. A collaborator identifies all requesters, gets the current local score for them, and divides the grid machines among the peers with positive scores proportionally to their scores. Any idle resources that are not allocated are shared among requesters with scores equal to zero. Similarly, if all requesters have scores equal to zero, they receive an equal share of the resources. Both consumer and provider update the appropriate local scores as soon as a resource is returned to the provider.

To update the local score of peer Q, peer P autonomously performs an accounting of the favors it has given to and received from Q, so as to avoid problems of forged accounting. A favor from Q to P is the allocation of some of Q's resources to P for some time. During this time, P will run some of the tasks which compose a given *job* (ie. a BoT application) on the resources obtained from Q. Accounting for the value of such tasks is not trivial. Note that valuing the favor based solely on the time for which P used Q's resources does not gauge how useful Q's resources were for P to run its jobs. Resources may vary in CPU speed, memory size, current load and available storage. These attributes may affect the performance of tasks on these resources in different ways. Moreover, since we do not assume that peers are trustworthy, P cannot trust accounting results that are based on information provided by Q. In fact, there is an incentive for Q to tamper with measurements of attributes to make its resources appear more valuable than they really are. To deal with these problems, we developed an autonomous relative accounting mechanism to determine how useful a resource was in running a task [47].

To do the relative accounting for a favor obtained from Q, P executes some of the tasks of the job on its own resources and some others in the resources donated by Q. Then P calculates the relative power of Q in relation to itself ($rpower_P(Q)$) as the ratio between the average execution time of the tasks ran locally and the average execution time of those run in Q. The favor Q has provided is accounted by P as the product of the duration of the favor by $rpower_P(Q)$. On the other hand, since $rpower_Q(Q) = 1$, Q accounts the favor it has provided to P as simply the duration of the favor. This way no information needs to be exchanged between consumer and provider, and it is not necessary to determine which resource characteristics affect which tasks. This relative accounting mechanism has been shown to be efficient even when the size of tasks varies widely, with an exponential distribution [47].

Note that the autonomous estimates of favor values are the only information that the peers need to allocate resources according to the Network of Favors.

5.2 Experimental Results

We now provide experimental data from the use of the Network of Favors in the OurGrid software. This data shows how well the implemented system provides incentives for collaboration. The implementation of the Network of Favors in OurGrid has one distinguishing aspect that we do not consider in our analysis or simulations, which is the pre-emption of tasks. Tasks are preempted in two situations. If a peer receives a request for machines from a local user, it tries to satisfy the request, even if in order to do so it has to abort work from remote peers that is being done in the local machines. This is meant to ensure that participating in the system can never be worse than being out of the system. Also, if a peer receives a request from another peer and decides to change the current allocation of its resources, it aborts any work being done in the machines which will be reallocated. As a performance metric, we measure the *makespan* of the jobs submitted to the peers—that is, the time between the submission of a given job and the end of the execution of its last task.

Our experiment consists of observing the behavior of a 4-peer system running OurGrid software under contention for resources. Each peer owns 4 grid machines and alternates between periods of idleness and demand for the resources in the network. All grid machines are identical. After the start of the experiment, each peer receives submissions of 60 jobs, each comprised of 40 1-minute long tasks. At each peer, there is a scheduler that manages the execution of jobs that have been submitted to that peer. This scheduler only starts the execution of a job if the previous job execution (if any) has finished. Otherwise, the submitted job waits in a queue until the previous job is completed. The times in minutes between the submission of any two jobs are given by the uniform distribution $U(1, 20)$. Note that if a peer uses only its local resources, it will take 10 minutes to complete a job (if we ignore the queueing, scheduling and communication overheads). The experiments were conducted in a controlled environment, and one run was performed for each of them.

We first executed the experiment with peers acting in isolation from each other, ie. without forming a P2P grid. In this experiment the average makespan over all jobs executed was 26.18 minutes. The increased average makespan was mostly due to queuing delays caused by jobs that ran concurrently. Then, we replayed the execution of the same jobs, this time with the peers forming a grid, ie. the same jobs were again submitted to each peer at the same time they were submitted in the previous experiment, but this time peers could use both local resources and remote resources from the other peers in the system. For each peer we computed the average makespan using the results collected for the execution of the first 55 jobs, when no peer had yet finished all 60 jobs. In this case, the 4 peers had an average makespan of 7.27, 7.55, 7.12 and 7.70 minutes, respectively, and the average makespan over all peers dropped to 7.41 minutes, which is 3.5 times faster than in the isolated setting, and indicates that some tasks were carried out using other peers' machines. Thus, there was a clear advantage for the peers to join the grid.

Finally, we measured the impact of a free rider joining the grid. Again, we replayed the previous execution adding a fifth peer to the grid with the same configuration of the other peers, but behaving as a free rider. In this case the average makespan for the 4 collaborators was 6.32, 7.20, 7.15 and 8.18, respectively, for an average makespan of 7.21 minutes, nearly the same as before. This indicates that collaborators are not significantly affected by the free rider peer. On the other hand, the average makespan of the free rider was 12.15 minutes. This indicates that although it was able to get some of the machines that were not contended and also decrease its average makespan, it would have been better off as a collaborator. Figure 1 plots the results collected for the execution of this last scenario and shows how the free rider is marginalized over time, from its 6^{th} job onwards.

5.3 Deployment Experience

OurGrid is currently in production as a free-to-join grid. A fresh snapshot of the system can be seen at http://status.ourgrid.org/. Our experience deploying this system and the feedback received from users were useful for verifying assumptions made when designing the Network of Favors and to show us new requirements for OurGrid.

Figure 2 presents a little over three months of data extracted from the logs

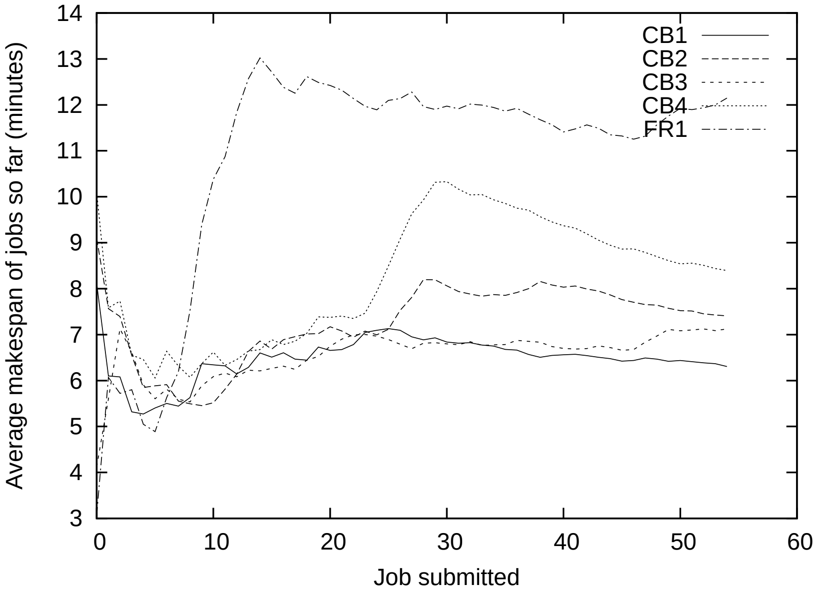

Fig. 1. Accumulated makespan average for the first 55 jobs submitted to the 5-peer grid

of all peers in the system and shows, at intervals of 2 hours, the evolution of the number of peers, machines available in the system (including those being used locally by the peers) and machines being donated. Since there are machines being donated most of the time, we can infer that most of the time: i) there is demand that is not satisfied by local resources (indicating consumer eagerness); and, ii) there are resources in the system which are not being used locally. Therefore, the resource sharing enabled by OurGrid is useful for its users.

An interesting lesson from our initial experience with OurGrid is that although it was designed not to depend on human cooperation, local policies to improve cooperation have been requested by users. For example, laboratories that have close collaborations offline would like to prioritize each other as recipients of resource donations, in addition to sharing resources with the rest of the system. This happens because besides being part of OurGrid, peers can also be part of offline communities. We intend to reflect this in the creation of sub-communities inside OurGrid. Members of the same sub-community will prioritize other members according to local policies. To identify such members securely, these sub-communities will need certificates issued by recognized Certification Authorities. However, peers only need to have a certification authority that is recognized by the sub-community. OurGrid does not require globally recognized certification authorities, and will still work with no certification authority at all.

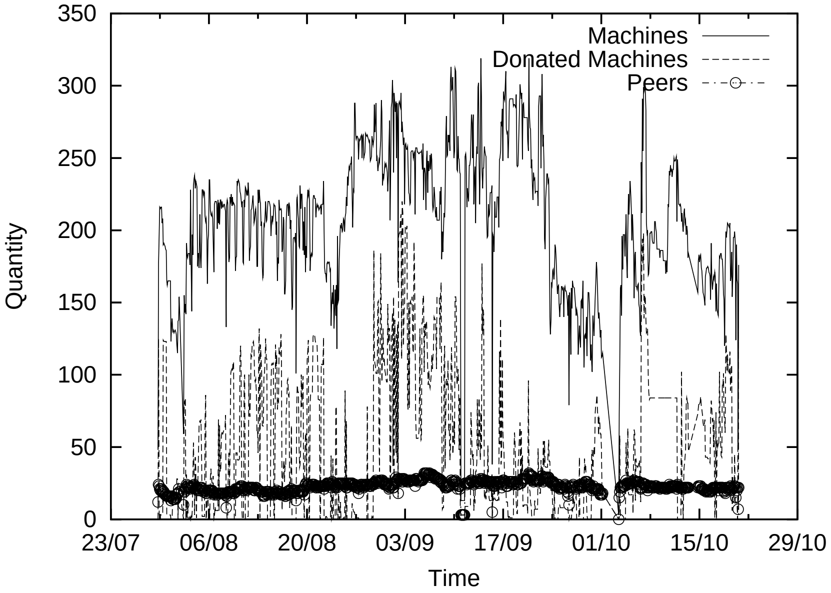

Fig. 2. Evolution of the production OurGrid system between July 29^{th} and October 20^{th} 2005

6 Conclusion

An important but frequently overlooked aspect of grid computing is how to create and manage the growth of grids. Currently, this process involves human negotiation. This is not flexible or scalable enough for grids with thousands of peers. In this paper we have presented the Network of Favors, which provides incentives for assembling grids without depending on human negotiation. We have investigated its behavior and discussed its implementation and use in OurGrid.

Compared to alternative solutions for the problem considered, the Network of Favors is much simpler and less costly, allowing systems that use it to depend on much less infrastructure than would be necessary for more sophisticated mechanisms. The drawback is that by choosing this simple reciprocation-based mechanism, one loses the greater flexibility provided by market-based incentive mechanisms.

The simplicity and effectiveness of the Network of Favors have allowed us to successfully build and deploy OurGrid (see www.ourgrid.org), which is in production since December 2004. OurGrid is currently being used to run applications in several different domains, including molecular dynamics, discrete-event simulations, climate forecast, image rendering, hydrological management, and data mining.

Acknowledgments

We are indebted to the whole OurGrid team, but particularly to William Voorsluys who ran the controlled experiments with the OurGrid software. Francisco Brasileiro and Walfredo Cirne thank the financial support from CNPq/Brazil (grants 307.954/2004, 141.655/2002, respectively). This work was partially developed in collaboration with HP Brazil R&D.

References

[1] CERN, LHC computing grid, http://lcg.web.cern.ch/LCG/.

[2] Grid3, An Application Grid Laboratory for Science, http://www.ivdgl.org/grid2003/.

[3] H. Jin, ChinaGrid: Making Grid Computing a Reality, in: Digital Libraries: International collaboration and cross-fiertilization, Vol. 3334 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 13–24.

[4] TeraGrid project site, http://www.teragrid.org/.

[5] UK e-Science Grid Support Centre, http://www.grid-support.ac.uk (2002–2005).

[6] N. Andrade, W. Cirne, F. V. Brasileiro, P. Roisenberg, OurGrid: An approach to easily assemble grids with equitable resource sharing, in: Proc. 9th Workshop on Job Scheduling Strategies for Parallel Processing, Vol. 2862 of Lecture Notes in Computer Science, Springer-Verlag Heidelberg, 2003, pp. 68–86.

[7] A. R. Butt, R. Zhang, Y. C. Hu, A self-organizing flock of condors, J. Parallel Distrib. Comput. 66 (1) (2006) 145–161.

[8] W. Cirne, F. Brasileiro, N. Andrade, L. Costa, A. Andrade, R. Novaes, M. Mowbray, Labs of the world, unite!!!, Journal of Grid Computing 4 (3) (2006) 225–246.

[9] G. Fedak, C. Germain, V. Neri, F. Cappello, XtremWeb: A generic global computing system, in: IEEE Int'l Symp. Cluster Computing and the Grid, Brisbane, Australia, IEEE Press, 2001, pp. 582–587.

[10] V. M. Lo, D. Zappala, D. Zhou, Y. Liu, S. Zhao, Cluster computing on the fly: P2P scheduling of idle cycles in the internet., in: IPTPS' 04: Proc. of the 3^{rd} International Workshop on Peer-to-Peer Systems, 2004, pp. 227–236.

[11] L. Oliveira, L. Lopes, F. Silva, P3 (Parallel Peer to Peer): An Internet Parallel Programming Environment, in: Web Engineering and Peer-to-Peer Computing Workshop, Vol. 2790 of Lecture Notes in Computer Science, Springer, Pisa, Italy, 2002, pp. 274–288.

[12] I. Taylor, M. Shields, I. Wang, R. Philp, Distributed P2P computing within triana: A galaxy visualization test case, in: IPDPS '03: Proceedings of the 17th International Symposium on Parallel and Distributed Processing, IEEE Computer Society, Washington, DC, USA, 2003, p. 16.1.

[13] I. T. Foster, A. Iamnitchi, On death, taxes, and the convergence of peer-to-peer and grid computing., in: IPTPS' 03: Proc. of the 2^{nd} International Workshop on Peer-to-Peer Systems, 2003, pp. 118–128.

[14] E. Adar, B. A. Huberman, Free Riding on Gnutella, First Monday 5 (10).

[15] D. Hughes, G. Coulson, J. Walkerdine, Freeriding on Gnutella Revisited: the Bell Tolls, IEEE Distributed Systems Online 6 (6).

[16] S. Saroiu, P. K. Gummadi, S. D. Gribble, A Measurement Study of Peer-to-Peer File Sharing Systems, in: Proc. Multimedia Computing and Networking 2002, San Jose CA, USA, 2002.

[17] B. Cohen, Incentives Build Robustness in BitTorrent, in: Proc. Workshop on Economics of Peer-to-Peer Systems, Berkeley CA, USA, 2003.

[18] eMule-Project.net, eMule site, http://www.emule-project.net/ (2002-2003).

[19] C. Grothoff, An Excess-Based Economic Model for Resource Allocation in Peer-to-Peer Networks, Wirtschaftsinformatik 3-2003.

[20] M. Feldman, K. Lai, I. Stoica, J. Chuang, Robust incentive techniques for peer-to-peer networks, in: EC '04: Proceedings of the 5th ACM conference on Electronic commerce, ACM Press, New York, NY, USA, 2004, pp. 102–111.

[21] K. Lai, M. Feldman, I. Stoica, J. Chuang, Incentives for Cooperation in Peer-to-Peer Networks, in: Proc. Workshop on Economics of Peer-to-Peer Systems, Berkeley, California, USA, 2003.

[22] D. Kondo, A. Chien, H. Casanova, Resource management for short-lived applications on enterprise desktop grids, in: Proceedings of Supercomputing 2004, Pittsburgh, PA, USA, 2004, p. 17.

[23] D. Abramson, R. Buyya, J. Giddy, A Computational Economy for Grid Computing and its Implementation in the Nimrod-G Resource Broker, Future Generation Computer Systems J. 18 (8) (2002) 1061–1074.

[24] R. Buyya, S. Vazhkudai, Compute Power Market: Towards a market-oriented grid, in: First IEEE/ACM Int'l Symp. Cluster Computing and the Grid, Brisbane, Australia, IEEE Computer Society Press, 2001, pp. 574–581.

[25] Grid Economic Services Architecture Working Group, Home page, http://www.doc.ic.ac.uk/~sjn5/GGF/gesa-wg.html (2002–2005).

[26] K. Lai, B. A. Huberman, L. Fine, Tycoon: A distributed market-based resource allocation system, Technical Report cs.DC/0404013, HP Labs, http://arxiv.org/abs/cs.DC/0404013 (Apr. 2004).

[27] J. Schneidman, C. Ng, D. Parker, A. AuYoung, A. C. Snoeren, A. Vahda, Why markets could (but don't currently) solve resource allocation problems in systems, in: 10th USENIX Workshop on Hot Topics in Operating System, 2005.

[28] E. Damiani, S. D. C. di Vimercati, S. Parabosci, P. Samaranti, Managing and Sharing Servents' Reputations in P2P Systems, IEEE Trans. Data and Knowledge Engineering 15 (4) (2003) 840–854.

[29] S. D. Kamvar, M. T. Schlosser, H. Garcia-Molina, The EigenTrust algorithm for reputation management in P2P networks, in: WWW '03: Proceedings of the 12th international conference on World Wide Web, ACM Press, New York, NY, USA, 2003, pp. 640–651.

[30] V. Vishnumurthy, S. Chandrakumar, E. G. Sirer, KARMA: A secure economic framework for peer-to-peer resource sharing, in: Proc. Workshop on Economics of Peer-to-Peer Systems, Berkeley, California, USA, 2003.

[31] B. N. Chun, Y. Fu, A. Vahdat, Bootstrapping a Distributed Computational Economy with Peer-to-Peer Bartering, in: Proc. 1st Workshop on Economics of Peer-to-Peer Systems, Berkeley CA, USA, 2003.

[32] Y. Fu, J. Chase, B. Chun, S. Schwab, A. Vahdat, Sharp: an architecture for secure resource peering, in: SOSP '03: Proceedings of the nineteenth ACM Symposium on Operating Systems Principles, ACM Press, New York, NY, USA, 2003, pp. 133–148.

[33] R. T. Ma, S. C. M. Lee, J. C. S. Lui, D. K. Y. Yau, An incentive mechanism for P2P networks, in: The 24^{th} International Conference on Distributed Computing Systems (ICDCS'04), IEEE CS Press, Tokyo, Japan, 2004, pp. 516–523.

[34] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, D. K. Y. Yau, A game theoretic approach to provide incentive and service differentiation in P2P networks, in: ACM SIGMETRICS/PERFORMANCE, ACM Press, New York, NY, USA, 2004, pp. 189–198.

[35] R. T. B. Ma, S. C. M. Lee, J. C. S. Lui, D. K. Y. Yau, Incentive and service differentiation in P2P networks: a game theoretic approach, IEEE/ACM Trans. Netw. 14 (5) (2006) 978–991.

[36] T. Yamagishi, M. Matsuda, Improving the lemons market with a reputation system: An experimental study of internet auctioning, Tech. rep., University of Hokkaido, http://joi.ito.com/archives/papers/Yamaghishi_ASQ1.pdf (May 2002).

[37] N. Andrade, F. Brasileiro, W. Cirne, M. Mowbray, Discouraging free riding in a peer-to-peer cpu-sharing grid, in: HPDC '04: Proceedings of the 13th IEEE International Symposium on High Performance Distributed Computing (HPDC'04), IEEE Computer Society, Washington, DC, USA, 2004, pp. 129–137.

[38] N. Andrade, M. Mowbray, W. Cirne, F. Brasileiro, When can an autonomous reputation scheme discourage free-riding in a peer-to-peer system?, in: CCGRID '04: Proceedings of the 2004 IEEE International Symposium on Cluster Computing and the Grid, IEEE Computer Society, Washington, DC, USA, 2004, pp. 440–448.

[39] W. Cirne, F. V. Brasileiro, J. Sauvé, N. Andrade, D. Paranhos, E. L. Santos-Neto, R. Medeiros, F. Silva, Grid computing for Bag-of-Tasks applications, in: Proc. 3rd IFIP Conference on E-Commerce, E-Business and E-Covernment, 2003.

[40] J. A. Smith, S. K. Shrivastava, A System for Fault-Tolerant Execution of Data and Compute Intensive Programs over a Network of Workstations, in: Lecture Notes in Computer Science, Vol. 1123, IEEE Press, 1996, pp. 487–495.

[41] J. Reynders, Informatics and Data Integration in Drug Discovery, Keynote presentation, 13th IEEE Symp. High Performance Distributed Computing (June 2004).

[42] Blinn's Law, http://slashdot.org/comments.pl?sid=20260&cid=2120694, anonymous Slashdot comment (Aug. 2001).

[43] D. Paranhos, W. Cirne, F. V. Brasileiro, Trading Cycles for Information: Using Replication to Schedule Bag-of-Tasks Applications on Computational Grids, in: Euro-Par 2003: Int'l Conf. Parallel and Distributed Computing, Vol. 2790 of Lecture Notes in Computer Science, Springer, 2003, pp. 169–180.

[44] E. L. Santos-Neto, W. Cirne, F. V. Brasileiro, A. Lima, Exploiting replication and data reuse to efficiently schedule data-intensive applications on grids, in: Proc. 10th Workshop on Job Scheduling Strategies for Parallel Processing, New York, USA, 2004, pp. 123–135.

[45] S. Basu, S. Banerjee, P. Sharma, S.-J. Lee, Nodewiz: peer-to-peer resource discovery for grids, in: CCGRID '05: Proceedings of the Fifth IEEE International Symposium on Cluster Computing and the Grid, IEEE Computer Society, Washington, DC, USA, 2005, pp. 213–220.

[46] F. Brasileiro, L. B. Costa, A. Andrade, W. Cirne, S. Basu, S. Banerjee, A large scale fault-tolerant grid information service, in: MCG '06: Proceedings of the 4th international workshop on Middleware for grid computing, ACM Press, New York, NY, USA, 2006, p. 14.

[47] R. Santos, A. Andrade, W. Cirne, F. Brasileiro, N. Andrade, Accurate autonomous accounting in peer-to-peer grids, in: MGC '05: Proceedings of the 3rd international workshop on Middleware for grid computing, ACM Press, New York, NY, USA, 2005, pp. 1–6.