ASSEMBLING COMMUNITY GRIDS FOR BAG-OF-TASKS APPLICATIONS

Nazareno Andrade
Departamento de Sistemas e Computação
Universidade Federal de Campina Grande
58.109-970, Campina Grande, PB, Brazil
http://dsc.ufcg.edu.br/ourgrid
nazareno@dsc.ufcg.edu.br

Abstract Besides having the necessary technology for running an application on a computational grid, a user must also have resources to form this grid. This very problem is still an open question, as there are no global infrastructures to which an ordinary user has access today. In fact, is very difficult for a user to obtain access to more than a handful of resources which owners she know, as this is done through personal negotiations with the resource owners. We argue that is possible for communities of users to have access to grid computing today using an exchange-based model of resource sharing, the *network of favors*. This simpler model does not depend on any additional infrastructure and can be deployed today, solving effectively the problem for a significant subset of the general grid assembling problem. We are experimenting these ideas building the OurGrid system of resource sharing [Andrade et al., 2003], that aims to solve the grid assembling problem for users of *bag-of-tasks* applications (those parallel applications whose tasks are independent).

Keywords: Grid Computing, Bag-of-Tasks applications, Community Grids, Peer-to-peer.

Introduction

An application needs resources to run. As obvious as it sounds, current grid computing solutions do not provide a mechanism for the user to obtain access to the resources her application needs. This problem is typically taken off scope. In Globus, for example, each resource's administrator has to manually map the user's grid-wide user-id to a local user-id [Czajkowski et al., 1998]. In MyGrid [Cirne et al., 2003]and APST [Casanova and Berman, 2003], as other examples, the user herself assembles her grid, using resources she has access to.

As such, a user must obtain access to each resource that she wishes to make part of her grid, needing to negotiate personally with each resource owner, which has no incentive for providing her resources. In many situations, however, to provide the non-trivial quality of service aimed by a user that is using grid computing, there is the need to gather as many resources as possible.

Automatic grid assembling is still an open question. There have been efforts in grid economy [Abramson et al., 2002, Buyya et al., 2000, Buyya and Vazhkudai, 2000]that aim to solve this problem. However, grid economy solutions demand non-trivial e-cash and e-banking infrastructure to be deployed. Moreover, it requires quality of service (QoS) guarantees (clients demand to receive what they "pay" for). As a result, it is currently very diffi cult to implement grid economy proposals in practice.

In this work, we present an alternative approach for automatically assembling on-demand grids. We are building OurGrid, a system that aims to do provide such an assembling it is based on the belief that it is possible to solve the grid assembling problem for a signifi cant subset of grid users relying on an approach much simpler than the grid economy approach. We argue that is possible for communities of users (e.g. the research community) to get access to a large set of resources in an exchanged-based way. This exchange is done through sharing accordingly to a *network of favors* model, which provides equity in the sharing and, as such, motivates the participation in the system and resource donation to the community.

Using such a simpler model, OurGrid does not give any guarantees about the QoS of the resources in the assembled grid. Not all types of applications are suitable for running in such a scenario. Specifi cally, the applications to which we aim to provide resources are *bag-of-tasks* (BoT) applications [Cirne et al., 2003], which are those applications whose tasks need no communication to proceed their computation, i.e., are independent. Despite their simplicity, BoT applications are used in a variety of scenarios, such as computational biology [Stiles et al., 1998], simulations, parameter sweep [Abramson et al., 2000]and computer imaging [Smallen et al., 2000].

1. OurGrid

OurGrid is a peer-to-peer resource sharing system targeted to providing resources to BoT applications. The central point of OurGrid is that the sharing is done using the network of favors model. In this model, each peer offers access to its idle resources to the community. In return, when there is work that exceeds local capacity, a peer expects to gain access to the idle resources of other participants. The system aims to allow users of BoT applications to easily obtain access and use the community's computational resources, dynamically forming an on-demand, large-scale, grid.

Since participation in an OurGrid community is voluntary, it is crucial to motivate the resource sharing. To do such, the network of favors (i) promotes equity (i.e., if the demand is greater than the offer of resources, the resources obtained from the grid should be proportional to resources donated to the grid), and (ii) makes the peer who have helped the community the most more prioritized when arbitrating conflicting requests to available resources.

In the network of favors, giving access to and allocating a resource to a requesting consumer is a favor. As such, the consumer becomes in debt with the owner of the consumed resources. The model is based on the reciprocation of favors to the consumers that a peer is in debt with, when solicited. If a participant does not act in this way, it is gradually less prioritized by the community, as its debt grows.

Every peer in the system keeps track of a local balance for each other known peer, based on their past interactions. This balance is used to prioritize peers with more credit when arbitrating conflicting requests. This balance is updated on each provided or consumed favor. The quantifi cation of each favor's value is done locally an independently. Hence, the system is completely autonomous. The emergent behavior of the system, however, is prioritizing the peers who have contributed more to the community, marginalizing the peers who do not reciprocate favors.

Using the network of favors model, the OurGrid system will assemble grids by inter-operating with existent middleware (e.g. Globus GRAM, Condor) to make the resources of the grid available to any client that uses the OurGrid resource sharing protocol (e.g. MyGrid, APST and Nimrod/G).

2. Preliminary evaluation

In this section we present some preliminary results we have from simulations and analytical evaluation of the proposed model. Due to its decentralized and autonomous nature, characterizing the behavior of an OurGrid community is both important and challenging. Therefore, initially, we based our analysis on a simplifi ed version of a simulation model meant to capture the key features of OurGrid, namely the system-wide behavior of the network of favors and the contention for fi nite resources.

The simplifi cation consists of grouping resource consumption into turns. In a turn, each peer is either a provider or a consumer. If a peer is a consumer, it tries to consume all available resources. If a peer is a provider, it allocates all resources it owns to the current turn consumers. In this model, every peer has an amount of identical resources that it owns and a static probability of being a provider in a given turn, denoted r and ρ, respectively.

In spite of its simplicity, this model already allowed us to draw some interesting conclusions about the network of favors. Figures 1 and 2 illustrate some of these results. A much more detailed discussion about this results is presented elsewhere [Andrade et al., 2003]. With this model, we were able to observe two properties: the equity and prioritization in the access to the resources. These properties are gauged by two metrics: the Favor Ratio (FR) and the Resource Gain (RG), respectively.

Favor Ratio is defined as the ratio of the accumulated amount of resources gained from the grid (note that this excludes the local resources consumed) by the accumulated amount of resources it has donated to the grid after a given turn. It represents the relation between the resources obtained and those donated to the community.

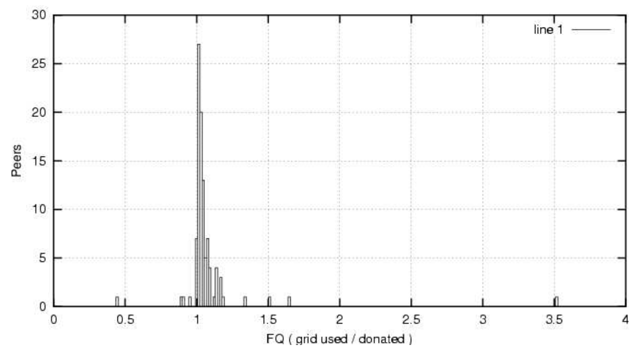

Figure 1. FR histogram in a 100-peer community with different allocation strategies on turn 3000

The histogram in Figure 1 shows the distribution of FR for a 100-peer community after 3000 turns. As can be seen, FR converged to 1. Due to the properties of this model, this denotes equity, as the peers were able to get back from the community as much resources as they donated. This convergence happened in all the simulated scenarios where there was enough competition to observe the contention for resources.

The RG of a peer after a given turn is obtained dividing the accumulated amount of resources used by it (both local and from the grid) by the accumulated amount of local resources it has used. RG represents the prioritization obtained by a peer when it requested resources, because whenever a peer asks for grid resources, it is also consuming its local resources.

Figure 2 shows RG for three peers with different values in a 10-peer community. It shows how a peer is prioritized as it donates more resources to the community. The peers have their RG proportional to the amount of resources they donated. Note that, the peer with the greatest ρ, RG explodes the scale in its first request, after it spent some turns providing. This gives us the almost vertical solid line in the graphic.

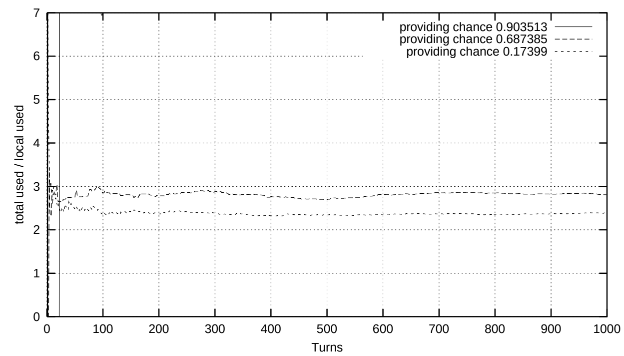

Figure 2. RG for three peers in a 10-peer community with different providing probabilities

3. Next Steps

The next steps in the OurGrid development are (i) studying the impact of malicious peers in the system; and (ii) the actual implementation and deployment of the OurGrid system. Now that we have evaluated the key characteristics of our network of favors, considering more realistic scenarios is needed to better understand the impact of the grid environment in the network of favors model.

Investigating peer's maliciousness is important mostly in two aspects in OurGrid: (i) to assure that a provider executed a task correctly and (ii) to make impossible for one to exploit the community using unfair accounting. To deal with the need of the consumer to assure correct task execution in unreliable providers, we plan to study how we can use replication in order to provide redundant results to gradually discover those providers a consumer can trust based as in the approach proposed in [Sarmenta, 2002]. We also want to study the use of a similar approach to cope with the objective of making the community tolerant to peers using unfair accounting.

We have started the OurGrid implementation as an extension of MyGrid[1] [Cirne et al., 2003]former work done at UFCG. OurGrid will be able to serve as a MyGrid resource in the user's grid, and will initially obtain access to resources through the already existent MyGrid's Grid Machine Interface (GM). The GM is an abstraction that provides access to different kinds of grid resources (Globus GRAM, MyGrid's UserAgent, Unix machines via ssh, etc.) and will allow OurGrid to interoperate with existing grid middleware.

4. Conclusions

We have presented our current work on assembling grids of resources owned by the users of a community through an exchange-based model. This approach aims to provide access to grid computing to a representative subset of the grid potential users in short term. Providing access to grid computing means not only providing these users with the technologies needed to run an application on the grid, but also the crucial part of gathering resources to them.

To experiment with this approach, we are developing the OurGrid resource sharing system. Through a network of favors based on a best-effort exchange-based model, we aim to build a system that provides equitable sharing of the idle resources from a community in a peer-to-peer network.

Our preliminary results point that our approach is promising. As such, we are currently pursuing more detailed analysis of the expected emergent behavior of the system and the implementation of a prototype. With OurGrid, we expect to give BoT users effective access to grid computing today, making possible that they use it even while the vision of the ubiquitous marketplace grid is not yet materialized.

Acknowledgements

This work would not have been done without the supervising of Prof. Walfredo Cirne and the insightful discussions with Prof. Francisco Brasileiro. We would also like to thank Hewlett Packard Brasil, CNPq and CAPES for the financial support.

Notes

1. MyGrid is an open-source grid broker for Bag-of-Tasks applications developed at the UFCG available at `http://dsc.ufcg.edu.br/mygrid/`

References

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

[Abramson et al., 2000] Abramson, D., Giddy, J., and Kotler, L. (2000). High performance parametric modeling with Nimrod/G: Killer application for the global grid? In *Proceedings of the IPDPS'2000*, pages 520–528. IEEE CS Press.

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

[Buyya et al., 2000] Buyya, R., Abramson, D., and Giddy, J. (2000). An economy driven resource management architecture for computational power grids. In *International Conference on Parallel and Distributed Processing Techniques and Applications*.

[Buyya and Vazhkudai, 2000] Buyya, R. and Vazhkudai, S. (2000). 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. IEEE Computer Society Press.

[Casanova and Berman, 2003] Casanova, H. and Berman, F. (2003). *Grid Computing: Making The Global Infrastructure a Reality*, chapter 33: Parameter Sweeps on the Grid with APST. Wiley.

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

[Czajkowski et al., 1998] Czajkowski, K., Foster, I., Karonis, N., Kesselman, C., Martin, S., Smith, W., and Tuecke, S. (1998). A resource management architecture for metacomputing systems. In *IPPS/SPDP'98 Workshop on Job Scheduling Strategies for Parallel Processing*, pages 62–82.

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

[Smallen et al., 2000] Smallen, S., Cirne, W., Frey, J., Berman, F., Wolski, R., Su, M.-H., Kesselman, C., Young, S., and Ellisman, M. (2000). Combining workstations and supercomputers to support grid applications: The parallel tomography experience. In *Proceedings of the HCW'2000 - Heterogeneous Computing Workshop*.

[Stiles et al., 1998] Stiles, J. R., Bartol, T. M., Salpeter, E. E., and Salpeter, M. M. (1998). Monte carlo simulation of neuromuscular transmitter release using MCell a general simulator of cellular physiological processes. *Computational Neuroscience*, pages 279–284.