A FUNCTIONAL SKETCH FOR RESOURCES MANAGEMENT IN COLLABORATIVE SYSTEMS FOR BUSINESS

Moca Mircea

Silaghi Gheorghe Cosmin

Universitatea Babeş-Bolyai Cluj-Napoca, Facultatea de Ştiinţe Economice şi Gestiunea Afacerilor, Str. Theodor Mihali nr. 58-60, {mircea.moca, gheorghe.silaghi}@econ.ubbcluj.ro, 0264-412569

Abstract: This paper presents a functional design sketch for the resource management module of a highly scalable collaborative system. Small and medium enterprises require such tools in order to benefit from and develop innovative business ideas and technologies. As computing power is a modern increasing demand and no easy and cheap solutions are defined, especially small companies or emerging business projects abide a more accessible alternative. Our work targets to settle a model for how P2P architecture can be used as infrastructure for a collaborative system that delivers resource access services. We are focused on finding a workable collaborative strategy between peers so that the system offers a cheap, trustable and quality service. Thus, in this phase we are not concerned about solutions for a specific type of task to be executed by peers, but only considering CPU power as resource. This work concerns the resource management module as a part of a larger project in which we aim to build a collaborative system for businesses with important resource demands.

Keywords: resource management, p2p, open-systems, service oriented computing, collaborative systems

1. Introduction

Nowadays businesses are becoming more and more dependent on technology. Applying a technology in a business can often increase returns. For example, a company can use the Internet in order to widen its market segment, employ a collaborative system to increase control over the production activity or to make the management process more efficient. Any of these situations may require innovation as a premise for a particular business progress. A new product, service or business idea requires building lots of scenarios and running simulations to conclude over the feasibility of the idea, which is the main pall for any investment.

Small and medium companies generally don't afford to buy access to computing resources for running their simulations. Computing services access is then a backset for small and medium companies.

Our work is related to the resource management module for a collaborative system that aims to meet small and medium companies' needs for computing resources. The service of access to computing resources is based on aggregating resources from all participating units. These entities are thought as collaborating nodes overlaying Internet.

In sections 2 and 3 we investigate technologies for achieving an economically efficient resource management for the mentioned collaborative system. Section 4 presents our work on establishing a functional model for the resource management module. Therefore, we discuss the general aspects of the module and assess alternative paths for the module behavior. In section 5 we conclude over the resource management module and the overall collaborative system, also presenting our future intentions concerning this work.

2. Technologies

2.1. Grids

Grid systems try to solve the following problem: the coordinated resource sharing and problems solving in dynamic, multi-institutional virtual organizations [5]. The CoreGrid network of excellence defines the grid as being "a fully distributed, dynamically reconfigurable, scalable, autonomous infrastructure to provide location independent, pervasive, reliable, secure and efficient access to a coordinated set of services encapsulating and virtualizing resources in order to generate *knowledge* [4].

The nowadays classical grid systems poorly fulfill the autonomy, scalability and dynamicity attributes of the above/listed definition. There are systems performing resource aggregation in a centralized manner building computing structures like TeraGrid [16] in USA, National Grid Service [17] in UK, Grid5000 [18] in France or the EGEE Grid [23] the project of the European Commission toward a global grid in Europe and world wide. Therefore, universities, research institutes, other selected organizations contribute with resources in this grid systems. These systems are used for running various scientific experiments that require huge resources, in various domains (particle physics, chemistry, biochemistry, medicine, Earth sciences, Life sciences, astronomy etc.). For participating in such a grid, an institution should follow a long and difficult process to get on with the technologies required to deploy the grid. To run experiments on the grid, a user should obtain security certificates, proving that the user already belongs to an agreed organization and it has a motivated research program for using the computing resources. Private companies are only very few involved in these grid systems. They prefer to build in-house closed grids to aggregate resources. Classical grid systems are based on middleware software platforms like Globus [19], Unicore [20] or gLite [21]. All the above mentioned platforms have problems to realize the scalability, autonomy and dynamicity attributes of the grid, because they are designed on a centralized coordination.

Classical Grids are relative closed systems, regarding the possibility of someone to enter the grid and contribute with resources. To outrun the organizational borders of the classical grid, the "grid economics" proposes the usage of methods from Economics for assessing and pricing the resources, for obtaining economic efficiency out of resource management [2]. Therefore they create the premises that each resource would have an associated value, users are able to pay the resources in correspondence with the utility they perceive from consuming the grid service. Therefore, negotiation between producers and consumers holds when allocating resources or when creating the so-called virtual organizations. The efficient exploitation of the grid is due to be realized through the usage of economic models. Economic-based resource management will be performed in a standardized framework, where the negotiation results will be described in the Service Level Agreements (SLAs), resources will be virtualized as web services. OGF [22] intends to standardize the WS-Agreement [15] language for SLAs. Part of the CoreGrid network of excellence, there are activities in this direction.

These research results are very promising, but, the classical grid systems still move to slow in this direction, also due to their centralized structure. Classical grid systems are mainly used for other scientific research, the money to support the grid are provided mainly from research councils or from the government; therefore, they do not push toward making the grid economically-efficient.

Up-to-date, the trust assumption holds in classical grids: the grid partners are trusted for delivering the quality of services they conceded to. This assumption does not hold on open systems that collect resources from various unsafe environments, or for systems that allow logging-in anonymous users. To tackle this problem, reputation models such as [13] are required for the grid to minimize cheating-users effects. This reputation model is based on the satisfaction perceived by the user and also intensively uses the huge amount of data supplied by the existing monitoring infrastructure.

2.2. Peer-to-peer

At the opposite side exists the P2P systems, which are applications that benefit from the resources (computing power, storage, network bandwidth) collected from Internet users [9]. These systems proved to be scalable because of the lack of centralization, auto-adaptable – they succeed to solve the absence of structure using internal mechanisms, and they are also very dynamic. In a remarkable vision paper, Foster and Iamnitchi [6] argue for the unification of grid and P2P, considering that both of these perform resource aggregation and sharing and each one has to solve problems that are identified in its counterpart system.

P2P is very popular for data distribution and sharing (Gnutella, Kademlia, BitTorrent), but are almost inexistent for sharing computing cycles. These systems are based on reciprocity. If one user wants to download a file, she would have to deliver other files to the system or to help the file distribution to other users. Technically, P2P systems are based on structured or un-structured networks of peers. Structured networks are organized on so-called DHT (dynamic hash tables), like Chord [14], CAN [10] or Pastry [11]. These systems provide with a structure and an algorithm for message routing and information retrieval. Unstructured P2P networks allow a free collaboration between peers. Usual, they have some fairness rules [3], which directs the collaboration to the general goal of the P2P network. An important advantage in P2P networks is that malicious behavior usually affects only small parts of them. There are developments

showing out possible attacks on the structure of a P2P network and possible solutions to fight against these attacks [12].

2.3 Desktop grids

After presenting in sections 2.1 and 2.2 grid and P2P networks, in what follows we will describe Desktop Grids (DG), which, in general, supplies with the same functionalities like a classical grid but, apart from classical grids, they collect the computing resources of the (desktop) PCs spread over the Internet or over some computer network. In particular, if the computing resources are collected from volunteer users in the Internet, we say that we built an Internet Desktop Grid (IDG) and we deal with volunteer computing. Considering that most of the DGs are also IDGs, in what follows, we will simply use the expression "Desktop Grids" for mentioning IDGs.

Actual DGs are technically built on a master-worker computational model [8]: the master delivers tasks to workers and collects and verifies the results. Although the resources are collected from distributed Internet users, a master exists – being a centralization issue. The master distributes tasks and data related with those tasks and coordinates the whole activity of a DG project. The success of DGs resides on the fact that is cheaper and more flexible to manage a "master" in a DG, rather than to manage a supercomputer. Therefore, the computing power becomes available "on demand", any time the user request it and at a fair price.

However, there are papers that try to design a desktop grid as a P2P system. We can mention the system proposed by researchers from Maryland University USA [7]. They analyze the situation that each peer can inject tasks in the DG, rather than only to contribute with resources. For the people to trust and use such DGs, these must be "trustable" and secure. In a P2P system, a peer joins in if it finds a benefit from joining the network. A peer will not participate in the system if the system will not offer him enough security warranties. In [7] the authors do not mention the mechanism that would attract volunteers to join the system and how to protect the system against sabotage, as in classical DGs. They omit the problem of protecting the volunteers against the risk that a peer to inject malicious code (viruses) or corrupted data into the system.

3. Resource management module

3.1. Overall system

We will shortly present now the main issues that determine our collaborative system. As a consequence of the previously presented technologies, we chose to design our system by employing P2P architecture. This ensures scalability of our system and openness to the aggregated resources. As a general view, the system gathers (buys) resources and then distributes (sell) them to different types of customers considering a SOA (Service Oriented Architecture).

The behavior of the system is wanted to be fully generated by economic rules (such as offer and demand rules). An important target which is not discussed in this paper but proposed for future work is the government of economics over our collaborative system. We are focused on finding a workable collaborative strategy between peers so that the system offers a trustable and quality computing service. Thus, in this phase we are not concerned about solutions for a specific type of task to be executed by peers, but only considering the computing capacity as resource.

3.2. System infrastructure

As Internet overlays, P2P systems can globally access resources. This is a key-advantage since by rule any resource is wanted in greater and greater quantity. Another reason is that resources in these networks are fast and cheap available. We will employ structured P2P architectures, because of their advantages over the unstructured ones. Such strong points are scalability, fault-tolerance resistance and guaranteed resource location. The structured P2P architectures evolved since their start, still being improved and this highly supports our target for global resource aggregation. We are not focused yet on a particular architecture, so we will make this decision based on results from simulations of our aimed resource management module on several P2P systems such as: Chord, CAN, Pastry, Kademlia, comparing the different specific results.

3.3. Resources aggregation

An overview on the resource management issue raises certain demands on how the system should act in terms of resource gathering and delivering. Therefore, each participating node holds a specific resource quantity available for the system. The way a peer announces its capacity could be made in a web-service manner by publishing this information (resource offer), this being a common habit in SOAs. In our system, the web interface will be replaced with other types of connection-based communication.

After the announcement of individual available computing capacity, there comes the aspect of aggregating these resources. In this context, we will establish a method for maintaining resource-offer aggregated information. This means a place or a method for accounting the aggregated resource. This could be done in two ways: either using leader-peers to centralize information about resources or defining an algorithm for searching through the network to satisfy ad-hoc resource demand. The algorithm searches and gathers computing capacity until the needed quantity is reached.

The approach of creating leader-peers brings into discussion hybrid-p2p scheme, which premises weaknesses points to the system. Therefore, certain nodes must be endowed with server capacities in order to centralize and deliver information about the aggregated resources. The challenge is, here, to determine which nodes and under what circumstances should become leaders (reputation criterion). The disadvantage of this presumed solution is that the centralized information must be updated as the network evolves and vulnerability of the system concerning the leader-peers. As P2P systems are highly dynamic networks we intuit an overcharge on the leader-nodes, but we are not sure about certain consequences unless testing the solution.

The second possible solution to gathering the resources employs pure-p2p architecture. In this situation, peers are functionally equivalent. When a service-customer wants to access the aggregated resource, it only contacts at random a node in the system. To ease the reference to this node, we will call it *initiator*. In response to the request, the initiator will run a *gather* procedure in order to build a list with those nodes in the system that will "work" for this same request. This hypothetical method has no threats about a single point of failure, since no information or functionality is centralized. However, an obvious disadvantage would be a certain delay from the moment the service-request was placed in the system until enough nodes are co-opted so that the requested capacity is reached. The question that arises here is about the size of the system segment from which the leader-node should centralize information. As presented in [14], managing information about all other nodes in the system is not scalable in P2P architecture, so an alternative such as keeping capacity and availability information about neighbors should be taken into consideration. Manifestly, the results of this approach will be highly influenced by the kind of P2P system architecture, since each scheme tackles the neighbors issue differently.

When the initiator searches for nodes, it traverses the network and asks nodes of their current available capacity. Based on this answer, the initiator infers whether a specific node is available at that moment at all. This aspect resembles with the behavior of nodes in BitTorrent network (reference), where a node can temporarily refuse under certain circumstances requests from other nodes. A key issue here is that the searching mechanism is very tight to the employed P2P architecture since routing mechanism and topology differs from one to another.

3.4. Resource owner

In this section we will discuss the aspect of resource ownership and how this influences the negotiation for resource. From a real situation depicture, as in figure 1, a resource can be available from an individual, or from a group of nodes. In this particular case, the group is controlled for example by a company that invested in computing resource for own use. Since there are moments when the resources are not used (or used at partial capacity) for own interest, the company is motivated for making them available to our system in turn for money. This trade can for example lead to a shorter amortization period. For a greater efficiency of our system, we are motivated to consider that a node from the group will represent the entire group when negotiating resources. This node will keep track of the availability and capacity of any other member in its group, being the negotiator of the company in our proposed system. The other possibility of resource existence is an individual computer at a certain physical location.

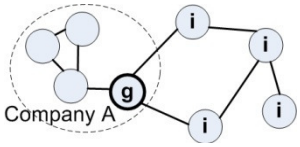

Figure 1: group-role node and individual-role nodes

From the discussed resource ownership aspects we draw the conclusion that a participating node can play one of the following two roles: individual-role, when the node acts for itself in the system and group-role for interfacing a group. The latter role makes the node to act in the system as if the resources of the whole group were its own. In figure 1 nodes with individual role are marked with "i" and the one with group-role with "g". The g-tagged node represents resources for company "A" in the system.

3.5. System trust

Since our proposed system globally aggregates resources, their heterogeneity must carefully be taken into consideration. This aspect does not concern as much the capacity and type of the resource as its availability and correctness. By resource correctness we mean all implied characteristics of a resource and its delivery so it is according to the negotiated parameters values. Therefore, our system should be endowed with a reputation mechanism, to reflect the quality parameters of a resource identity. Such a mechanism would contain pointing rules that maximizes the total score if the resource is stable (stays up for long time) and correct (doesn't cheat).

The cheating aspect has the most important influence on the system trust. If there are nodes that receive a resource request and they agree but finally deliver an intentionally wrong result the system will be suspected and abandoned by customers. Consequently, the system failed from its target.

3.6. Peer specialization

In P2P systems, nodes are free to come and go as they whish. Therefore, our system will consider a mechanism for avoiding possible losses determined by a sudden node failure. A suitable approach for our system would be replication, as a common technique presented in structured P2P papers (struct.P2Particles). In our system situation, replication would mean periodically copies of a "working" node state to one or several of its neighbors. Therefore, the node that holds the copy would periodically check the presence of the node from which it received the copy. The moment no answer is received from it, a recovery action would take place.

Based on the presented resource management functionality of our system, we intend to build an upper layer on it, in order to differentiate resource identities by functionality semantic. While the resource management discussed in previous sections only concerns raw resource availability and delivery (hardware resources), this layer groups nodes into service-type classes (software resources). That is, the resource at a node can be used to perform a specific action, such as: physics-specific computations, accounting operations, graphic analyses, and so on. This layer plays an obvious role, since computing resource can only be used by running specific algorithms on it.

Having the second layer in mind, one can imagine our system as a large group of nodes with available resources, each of them being capable of performing a strict set of operations. When a customer needs complex information processing, it places a request on the system, which is further analyzed, decomposed and processed by certain resource entities in the system.

The upper layer acts for the system just like division of labor in human society, where each individual is a potential resource, but which becomes a full-resource when acting in a specific way. The specific behavior of an individual means specialization which is a brick in a particular workflow wall. If we consider the workflow of a complex activity, each peace of it would be carried out by adequate specialized nodes in our system.

3.7. Resource atomicity

There are two reasons for which we consider that the system should comprise different levels of resource aggregation. The first is that an individual-role peer represents a too small resource unit as compared to the

total resource demand for a resource request placed by a customer. That is, when searching for peers, the initiator must negotiate with many nodes in order to reach the demanded resource quantity. This problem resembles with the need of measuring or representing a long distance with a very small length measure unit, which is an arduous task. The second aspect is when payment for the service is done. If the customer gets to the moment of payment, it is not concerned at all about how many resource units needed its request and what is their identity. Thus, the payment must be centralized, implied resource units being transparent for the user. These two reasons lead us to creating leader peers, which represent, thus negotiate and receive payment in the name of a certain number of individual peers.

Individuals from a group are considered to form an alliance, since by this way, they can obtain a better price for their individual resources compared to the situation of acting alone. This is because a customer with a large resource demand would deal with a group much easier than with many individuals.

Figure shows our system as groups of resource individuals, represented by leader peers. Leader peers are tagged with "g", since they play a group-role and the individuals from a group with "i".

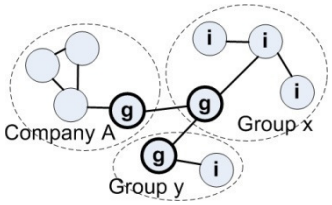

Figure 2: System viewed as groups of resource individuals

Two of the challenges here are to settle a rule to maintain the leader peer as the network evolves and to establish the size of a group.

3.8. Resource brokerage

When a user places a resource request in the system, a certain peer should handle it. This node is the access point of the customer in the system. Consequently, all actions that lead to delivering satisfactory results to the customer start from here. Once the conditions for the resource request are established by the user, the system must gather resources in order to accomplish the request according to those conditions. An example of such restriction would be the price that the customer is willing to pay for the resource. The finding of the available resources corresponding to the user conditions could be made through a negotiating process. Thus, we propose the broker peer role, as functionality responsible with finding user-restrictions-compliant resources in the system. In an implementation, the broker role could overlay the leader role since they both represent centralized, apart functionality.

4. Conclusions and future work

As mentioned in section 3.1, we are not aware of system behavior when employing a particular P2P architecture, so a future target is to test several architectures, compare the results and choose the most appropriate solution according to our functional design.

An important issue that concerns our model is the service currency. We have discussed functionality of the resource management module but haven't specified the motivation of different entities for making available their resources. Thus, we are concerned to find a realistic, workable and economically-driven scheme for determining participants to share and access resources. This target will consist of studying the implications of economics in the resource management module behavior.

5. References

1. I. Anderson, G. Fedak. "The computational and storage potential of Volunteer Computing". In Proceedings of 6th IEEE Intl Symposium on Cluster Computing and the Grid (CCGRID06) 2006
2. R. Buyya, D. Abramson, J. Giddy, H. Stockinger. "Economic models for resource management and scheduling in Grid computing", in Concurrency and Computation: Practice and Experience; 14(13–15), 2002
3. B. Cohen, "Incentives Build Robustness in BitTorrent", in Proceedings of the 1st Workshop on the Economics of Peer-to-Peer Systems, Berkeley, CA, June 2003

4. CoreGrid vision about the Grid, http://www.coregrid.net/mambo/content/view/2/25/
5. I. Foster, C. Kesselman, S. Tuecke. “The Anatomy of the Grid: Enabling scalable virtual organizations”. In International Journal of Supercomputer, Applications, 15(3), 2001
6. I. Foster, A. Iamnitchi, “On Death, Taxes and Convergence of Peer-to-peer and Grid Computing ”. In Lecture Notes in Computer Science, 2735, Springer Verlag, 2003
7. J. Kim, B. Nam, P. Keleher, M. Marsh, B. Bhattacharjee, A. Sussman. “Creating a Robust Desktop Grid using Peer-to-Peer Services”, in 2007 NSF Next Generation Software Workshop (NSFNGS 2007)
8. L.F.G. Sarmenta, “Sabotage Tolerance Mechanisms for Volunteer Computing Systems”, in Future Generation Computer Systems, 18(4), 2002
9. C. Shirky, . “What is P2P and what Isn’t”, www.openp2p.com/pub/a/p2p/2000/11/24/shirky1-whatisp2p.html , 2000
10. Sylvia Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker. “A Scalable Content Addressable Network”, in Proceedings of 2001 ACM SIGCOMM Conference
11. A. Rowstron, P. Druschel. “Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems”. In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, 2001
12. A. Singh, T.W. Ngan, P. Druschel, D.S. Wallach, “Eclipse Attacks on Overlay Networks: Threats and Defenses”, in INFOCOM06: 25th IEEE Intl Conference on Computer Communications
13. G.C. Silaghi, A.E. Arenas, L.M. Silva, “A Utility-based Reputation Model for Service oriented Computing”, in 1st CoreGrid Symposium, Rennes, France, 2007
14. Ion Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan. “CHORD: A Scalable Peer-to-Peer Lookup Service for Intenet Applications”, in Proceedins of 2001 ACM SIGCOMM Conference
15. OFG GRAAP working group, “Web Services Agreement specification”, 2005, http://www.ogf.org/Public_Comment_Docs/Documents/Oct-2005/WS-AgreementSpecificationDraft050920.pdf
16. http://www.teragrid.org/
17. http://www.grid-support.ac.uk/
18. https://www.grid5000.fr/
19. http://www.globus.org/
20. http://www.unicore.org/
21. http://glite.web.cern.ch/glite/
22. http://www.ogf.org/
23. http://www.eu-egee.org/