Organization of Virtual Communities

Asif Akram, Rob J Allan
CCLRC Daresbury Laboratory, Daresbury, UK
{a.akram, r.j.allan}@dl.ac.uk

Abstract

Locating suitable resources within a Grid system is a computationally intensive process, with no guarantee of quality and suitability of the discovered resources. An alternative approach is to categorize resources based on the services they provide – leading to the interaction of peers with common goals to form societies/communities. Organization of resources in different communities is suggested to be useful for efficient resource discovery. We have implemented JXTA prototype to illustrate the concepts of community formation in which Peers offering different services can be grouped together based on different criteria.

1. Introduction

Emerging distributed computing paradigms, such as a Grid Computing, comprise of dynamic and distributed resources/peers; which can be organized as a "Virtual Organizations". Resource discovery is a time-consuming process and imposes an overhead on network access in the Grid. The number of interactions is likely to increase exponentially as the number of peers grows. Restricting interactions between the set of peers is a key factor to scale the resource discovery problem. Peers are categorized based on criteria i.e. type and quality of service, etc. Any initial cost in categorizing peers results in discovering "preferable" resource with minimum discovery cost subsequently – thereby leading to the development of "communities".

The concept of communities is similar to interactions between different departments at a university. For instance, a lecturer can be a member of different faculties e.g. a mathematics lecturer teaching calculus to computer science students. This analogy helps us to define two terms, Expertise and Interest [12], [13]. Expertise of a peer is the basic service provided by that peer and Interest of a peer is the service/services provided by other peers which are supportive to its main service. A similar problem in the Grid Computing is what Davis and Smith refer to as the "connection problem" [1], where peers need to find other suitable peers to co-operate with, assist, or interact with. "Focused Addressing" [2] is one solution to the connection problem where requests are sent to a particular subset of peers, believed to assist the requesting peer.

Individual peers, although selfish, are expected to interact with each other in some way. Co-operation of one form or another therefore becomes essential. Each peer prefers to be in environment where it may be easily discovered by a suitable user, and can locate other peers with minimum efforts. Peers providing different services may be grouped together based on attributes such as type of services, resources owned and domains of operation. Each community has one Service Peer with dual responsibility of not only managing the member peers but also keeping track of other communities with which it interact on behalf of member peers.

In Community environment, members propagate and share information in a decentralized, self-organizing and open manner. Data is not owned by a particular member or a server; and is passed around, flowing freely towards the end subscribers without centralized control or management. The community as a whole ensures the protection and persistency of data through its unique ability to adapt, resist and protect data by scattering the multiple copies within the community boundary. Reliability and resilience is developed by making each member as interchangeable as possible.

2. Community Formation

When a new peer joins the Grid, it tries to discover the Service Peer which may have interest in its capabilities/services. If the interests of a Service Peer are different, the new peer is either referred to any suitable Service Peer/s, or the new peer tries to locate alternative Service Peer/s with compatible interests. A Service Peer and all peers registered with it constitute a community. A Service Peer manages all peers within the community and communicates with neighboring Service Peers from other communities on the behalf of member peers. A Service Peer is essential for the bootstrapping of a new peer, as it supports a new peer to discover enough network resources to sustain itself. We therefore also foresee the existence of common infrastructure services (such as monitoring, directory, security/certificate authority, etc) within each community.

3. Types of Communities

Individual autonomous peers have expertise and interests in specific resource/s. Based on these expertise and interests, peers are grouped together, but expertise and interests are not the only criteria for categorizing peers. Communities/societies can be of different types as mentioned below:

3.1 Competing Community. In a Competing Community each peer has approximately the same expertise – although some service attributes may vary. Similarity in services may develop competition amongst member peers; member peers compete against each other to get selected by a client by adapting their attributes, e.g. cost or QoS. These are sometimes referred to as “service pools”.

3.2 Co-Operative Community. In Co-Operative communities all peers provide different services, which must be used alongside services of other member peers. In such communities, each peer is dependent on at least one other member peer. Hence, when one peer is selected, then the possibility of selecting another member peer providing utility service/s is increased. This mutual co-operation is suitable for peers which provide simple services. This community is the basis for service aggregation or choreography.

3.3 Goal Oriented Community. This is a collection of peers working together to achieve a particular goal. Membership in such a community is only to accomplish the assigned task. Goal oriented communities are important in self-organizing systems, where interactions between member peers are not pre-defined, but the services required are. In such instances, member peers may interact with each other in arbitrary ways to achieve a given end result, probably via a workflow prescribed from a particular business or research process.

3.4 Ad Hoc Community. Peers can be in a co-operative or competing community, but need to work together as a team. In ad hoc communities peers interact directly with each other without interference and involvement of the CC. They need different mechanisms to query each other’s capabilities. Peers belonging to different communities providing different but supporting services form the basis of an ad hoc community, as long as both concerned communities have agreed to use each other’s services. Communication between the coordinators of different communities is in itself a process of ad hoc initiation of a larger community.

3.5 Domain-Oriented Community. Such a community is formed by linking together similar-minded organizations and institutions, instead of the services they provide, such as academic communities, research communities, and open-source communities. Hence these communities are domain-oriented rather than service-oriented. The current VRE projects tend to be of this type [13].

3.6 Virtual Community. The Virtual Community is a community of communities. In the Virtual Community, Coordinators from different Communities can directly interact with member Peers of other Communities; as they are in their own Community without further involvement of corresponding Service Peer. This effect is achieved by leasing out the member Peer to other community for certain time period, before that lease period either Service Peer requests to renew the lease of corresponding Peer or it can’t use the service of the Peer directly.

3.7 Sharing Community. In this type of Community different Communities share their resources with each other; this sharing of resources is not restricted only to member Peers. Community A may have QoS monitoring module, which it shares with Community B assuming either Community B doesn’t have such module or Community A may have more advanced monitoring module or may be QoS monitoring module in Community B is overloaded. In return Community B may make few of its own resources available to Community A with certain limitations and restrictions according to its own policy, which must be negotiable for maximum flexibility.

4. Architecture of Toolkit

Applications which involve collaborations of one form or another are ill suited for to the classical client-server model. Ubiquitous Computing is the field where Grid model has significant importance. Ubiquitous Computing can be defined as “making many computers available throughout the physical environment all offering and consuming each others services, while making them effectively invisible to the user”. Ubiquitous computing is held by some to be the Third Wave of computing. The First Wave was many people per computer; the Second Wave was one person per computer; the Third Wave is many computers per person. A dynamic collaborative network, where peers

providing and consuming services come and go, is better suited for Grid.
A prerequisite for ubiquitous computing to work is that Peers are able to search for each others services in way that is uniform, flexible and powerful, yet simple enough for the required protocols to fit in resource-constrained embedded devices such as hand-held computers or mobile phones. Service/Resource discovery in Grid computing is very crucial for self sustaining system and efficient resource discovery is one of the key factors.

4.1 Objectives

The goal of this work is to develop an efficient service discovery system by organizing peers into communities for Grid networks. Main objective of the project is to automate continuous search and membership to Communities by Peers based on membership policies of Communities and selection criteria of Peers. We will study the stability and dynamics of the system, when Peers change communities to achieve their personal goals and effect of new Peers joining the system.

4.2 Tool Selection

JXTA (jxta.org) [10] is a open source P2P framework initiated by Sun Microsystems. The JXTA protocols are independent of any programming language, and multiple implementations exist for different environments which make it best choice for prototype. The JXTA network consists of a series of interconnected nodes, or Peers. A JXTA Peers is "any entity capable of performing some useful work and communicating the results of that work to another entity over a network...". Peers can self-organize into Peers Groups, which provide a common set of services. JXTA has the concept of Peers and Peer Groups which match to our vision of Peers and Communities, which makes JXTA as best choice for implementing our prototype. Secondly as JXTA is platform independent and Peers can be different hardware nodes connected to network sharing different services and resources with other Peers it gives us flexibility of improving our prototype for different devices. Currently available P2P systems tend to use protocols which are proprietary and independent of other networks, incapable of leveraging their services. Each system creates its own P2P community, duplicating efforts in creating software primitives required by P2P systems, such as managing underlying physical network. This problem was solved by the project JXTA which provided a common P2P platform that is platform and language agnostics and spares the difficulty of designing protocols to handle the communication.

4.3 Design View

Our main goal is to organize Peers in Communities based on the services or resources they are offering to collaborative Grid network for efficient resource discovery. To achieve our goal we have defined few protocols:

4.3.1 Description: JXTA has its own notation of describing services offered by Peers and Peer Groups in the form of Module, Specification and Implementation Advertisement in XML format. One possibility is to use these default Module Advertisement for the services or we can use XML based Peer Advertisements for each Peer. Modifying XML based Peer Advertisement gives more flexibility as it is collection of all services provided by given Peer but in the early stages of prototype our Peers are not providing any specific concrete services and don't have corresponding different Module Advertisements. Description of every service can be divided in two categories:
a) Service Independent Parameters
Each service maintains a list of general parameters which are independent of the nature and type of service i.e. CPU usage, reliability, Quality of Service (QoS) or remote storage facility etc. These general parameters are common for most of the services offered by Peer or Peer Group and can be used to describe the policy for Peer membership or discovery [8] or monitoring Quality of Service information [9]. Service independent parameters are described in XML format and are part of its advertisement.
b) Service Dependent Parameters
These parameters are based on the type and nature of the service and tightly bound to that specific service. Each service will have its own set of these dependent parameters and which vary from service to service.

4.3.2 Reasoning: Peers provide different services/resources and based on these services they can be organized in different communities. Peers apply for the membership and Community can accept or refuse membership request based on the service/s provided by the Peers. In order to decide about membership request, community parses the description of services provided by Peers and compares it with its own Membership policy [8]. Community selects Peers as members which are most compatible with their policies.

4.3.3 Rating: Different Peers can provide similar service/s and one Peer can provide many services, based on quality of services and number of services there should be mechanism to rate service/s provided by Peers and extending that mechanism to rate Peers themselves. This rating mechanism [6] can be one criteria in the Reasoning discussed above.

4.3.4 Membership: Services offered by Peers can be arranged in many different ways to have different type of communities and each community will have its own membership policy. Membership to any Community is based on the Description of Service/s provided by Peer, Reasoning of the service description and Rating of Service/s and Peer which is applying for the membership. Peers can also have different membership policy for different communities. Peer offering many services may not expose all of its services in every joined community or it may use different access policy for any service in different communities i.e. Priority Based, First Come First Served (FCFS), Shortest Job First (SJF).

4.4 Graphical User Interface (GUI)

In our prototype we have Java GUI which can be used to access different core goals of the system. The prototype mainly offers Management of Peers and Groups and Management of System. Management of Peers and Groups means “Creating”, “Removing”, “Searching” and “Modifying” Group and Peers. Management of System is to manage the membership of Peers in Peer Groups and different options to control how Peer/s join or leave Group/s. The current status of any Peer or Group can be checked from the management category. These individual features are discussed below with key points and limitations. Few of the options are valid both for Peers and Groups.

Fig. 1. Main Menu

Create: This option helps to create new Group. User selects this option and in a new GUI gives the name of Peer Group, its description and selects category of community from drop down menu. Description of Peer Group is one of the criteria in the membership policy and Peer Group description is matched with the description of the Peer at the time of membership. Each newly created Peer Group is randomly assigned ranking within the range of 1 – 100 using Math.random() method.

Remove: Any Peer Group can be removed from the system at any time by giving its name and ID. If Peer Group with given name or ID is found then it is removed and member Peers are notified for appropriate changes.

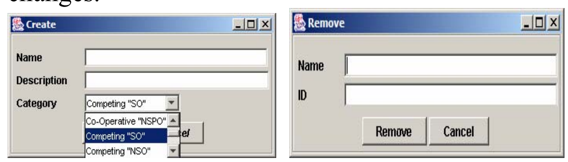

Fig. 2. Menu to Create and Remove Peer Group

Search: Peer Groups can be searched based on its name, id, description or category. Peer Groups which fulfill criterion partially or fully are displayed in the JTextArea. Double clicking any Peer Group in the text area displays its properties in non-editable GUI.

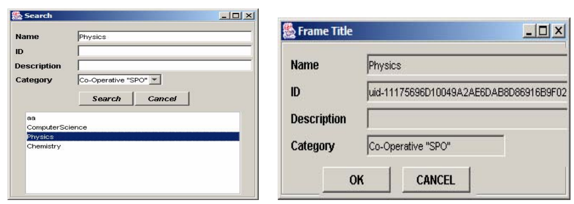

Fig. 3. Menu to search Pear Group and non-editable display

Change Properties: It is possible to change the properties of Peer Group after searching the Peer Group. Change properties functionality uses the same search feature as described above but this time editable GUI is presented where name or description of the Peer Group can be changed.

Management of system controls any Peer joining different Peer Groups, any Peer resigning from the Peer Group, any Peer Group terminating membership of any Peer, displaying all Peers and Peer Groups and displaying members of any Peer Group. Different sub-options in the management category have same GUI but based on the sub-option selected available features in the GUI may vary.

Join Group: This option can be used to add manually any Peer in any Peer Group, by selecting any Peer Group from the left hand tree of all created Peer Groups and any Peer from right hand side tree displaying all created Peers and clicking the JOIN button. Selected Group will check its own membership policy and its current members and will compare the ranking of lowest rated member with the rating of Peer applying for the membership. If new Peer has higher ranking than ranking of lowest rated existing member then Peer Group will terminate the membership of existing lowest rated member and give membership to the new Peer, otherwise it will decline the membership request of new Peer.

Leave Group: This option is exactly opposite to the option Join Group, through this option any Peer can be forced to leave any Peer Group. When a Peer leaves member Group then resign method of Peer will be called and list of members for both Peer Group and Peer will be updated. If the Peer is not a member of selected Group then error will be generated.

All Groups and All Peers: These options display all Groups or all Peers and selecting either any Peer or Group its current members can be checked by clicking the option Properties. Selected Peer or Peer Group along with its SortedList of members is displayed in the tree form**.**

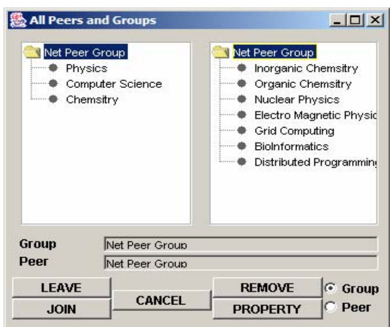

Fig. 4. Management view of the prototype

5. Limitations

Our prototype is more about organizing different Peers in Communities and we do not intend to implement our system in real life devices during the early stages of our development. This prototype was tested by running all Peers on workstations/PC on local area network. The main goal is to discover services offered by different devices including portable devices in quick and reliable way.

This work is not about creating different services offered by different Peers and for time being Peers are providing simple dummy service descriptions which are used by membership policy. No attempt is made to rate services based on any solid criteria offered by Peers but Peers themselves are rated randomly at the time when they are created.

GUI created for simulation purposes is also very simple and supporting basic functionality. There is lot of work required to make this GUI all-purpose simulation for all ubiquitous computing which will be addressed in the latter updates.

6. Results

Prototype was evaluated in control environment with following assumptions:

- Only Peer discovers Groups.
- Peer randomly selects one of discovered Group and match Group description with its own description.
- Peers can join limited number of Groups.
- Groups can have limited number of Peers.

Prototype was evaluated with different set of parameters i.e. number of member Peers in Group, number of joined Groups by single Peer etc. Evaluation results were quite encouraging, and similar pattern was observed even with different set of parameters. In the beginning of evaluation following four steps were quite frequent:

- Selection of Group by Peers
- Compatibility checks i.e. rating and description by Peers.
- Request for Membership by Peers
- Membership confirmation from Group.

TABLE I
Preliminary Result with different constraints to achieve stability

Rate of Discovery	**Acceptance/Request Ratio** Group Size (5)	Group Size (8)	Group Size(10)
0 - 100	80/100	100/100	100/100
	0.8	1.0	1.0
100 - 200	60/80	76/90	90/100
	0.75	0.85	0.9
200 - 300	30/50	50/75	63/85
	0.6	0.66	0.74
300 - 400	10/25	28/50	28/55
	0.4	0.45	0.5
400 - 500	0/5	3/18	5/21
	0	0.16	0.23

With the passage of time frequency of step 3 and 4 decreased, which means drop in membership requests from Peers and acceptance of membership from Groups. Once System becomes stable then even Peer doesn't apply for membership. The preliminary result with different set of constraints is shown below:

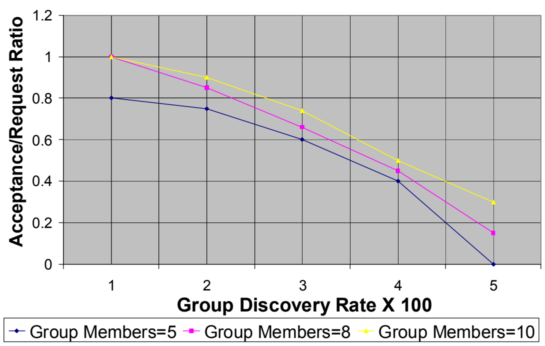

Figure 7. Graphical Presentation of Acceptance/Request Ratio vs. Peer Group Discovery Rate by different Peers

Time required to achieve stable state depends on the constraints set by system i.e. rate of Group Discovery, number of member Peers in a single Group and number of Groups joined by single Peer. We are confidant that organizing resources into communities will give new dimension to Grid Computing.

Outcome of the simulation is:

1) Discovery Process is continuous.
2) In beginning Peer and Groups membership has no dependency on Ranking.
3) Groups become selective much earlier then Peers as number of Groups is less than number of Peers.
4) Peers also achieve stability and make fewer attempts for new membership but this never ceases, but it decreases to great extent.

7. Conclusion

We have presented the concept of categorizing peers in communities on the basis of their expertise and interests. Social networks are a natural way for people to go about seeking information. Organizing peers in one form or another makes the discovery of resources efficient, whilst minimizing computational overheads. Categorizing the peers in communities is simple, open and easy to implement, and the initial overhead of developing communities pays-off latter at the time of resource discovery. Communities are more stable, and stability increases with the passage of time, communities have a simple learning time and are more adaptive to operate in a dynamic environment. We have proposed the external and internal rating for communities and peers respectively which may be used to support a given Quality of Service, effective participation of autonomous peers and better interaction among communities and member peers. Finally, we discuss the different services required to manage the group and requirements of the member peers. A JXTA implementation of a prototype system is discussed to describe the salient features of our approach. A key theme of this work is to determine how communities should be structured to support resource discovery, and how particular roles within a community can be used to determine interactions between participants within a community, and those between participants across community. This work extends techniques and results discussed in [11].

8. References

[1] Davis, R. and R. G. Smith: 'Negotiation as a Metaphor fro Distributed Problem Solving'. Artificial Intelligence 20, 63-109, 1983.

[2] Parunak, H. V. D.: 'Distributed Artificial Intelligence', Chapt. Manufacturing Experience With the Contract Net, pp. 285-310, Research Notes in Artificial Intelligence. Los Altos, CA: Morgan Kaufmann Publishers, 1987.

[3] P2People http://www.iti.gr/db.php/en/projects/P2PEOPLE.html

[4] Kautz, Henry, Milewski, Al, and Selman Bart, "Agent Amplified Communication," AAAI '95 Spring Symposium Workshop Notes on Information Gathering in Distributed, Heterogeneous Environments, Stanford, CA.

[5] Kuokka, Daniel, and Harada, Larry, "Matchmaking for Information Agents," Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI) '95, 1995.

[6] Leonard Foner: 'Clustering and Information Sharing in an Ecology of Cooperating Agents, or How to Gossip without Spilling the Beans', Conference on Computers, Freedom and Privacy, 1995.

[7] [UP2P] http://chat.carleton.ca/~narthorn/project/UP2POverview.html

[8] Catalin Dumitrescu (UChicago), Ian Foster (UChicago/Argonne), Mike Wilde (UChicago/Argonne), "Policy-based Resource Allocation for Virtual Organizations", iVDGL/GriPhyN Presentation, Fall 2002

[9] Omer Rana, Asif Akram, Rashid Al-Ali, David W.Walker, Gregor von Laszewski, Kaizar Amin: Quality of Services Based Grid Communities. Web Services and Agent Systems. Kluwer Academic (2004)

[10] JXTA http://www.jxta.org/

[11] Steven Lynden and Omer Rana, "Coordinated Learning to support Resource Management in Computational Grids" Second International Conference on Peer-to-Peer Computing, (2002) 81-89

[12] Leonard Foner. Yenta: 'A multi-agent, referral-based matchmaking system'. 1st International Conference on Autonomous Agents, (1997) 301–307.

[13] Bin Yu and Munindar P. Singh: 'Searching Social Networks', Proceedings of Second International Joint Conference on Autonomous Agents and Multi-Agent Systems, 2003.