Dicovery of resources in a QoS enabled Grid infrastructure using SORD

Ioannis Liabotis, Lionel Sacks
Departement of Electrical and Electronic Engineering, UCL
{i.liabotis | lsacks@ee.ucl.ac.uk}

25th July 2003

Abstract

In this paper, we present the design of a self organising resource discovery protocol (SORD) for efficient resource discovery and job allocation for cluster computing/Grid scenarios. The protocol is fully distributed and is based on the peer to peer architecture and small world topologies. The evaluation of the protocol is done through simulation. The simulation scenario assumes a QoS enabled Grid infrastructure that provides Gold, Silver and Bronze class of service. We demonstrate that using SORD we can achieve efficient resource discovery minimizing the management traffic that a centralized approach would require.

1 Introduction

Advances in networking, communication, storage, computing and multimedia technologies together with new application areas drives the merging of computing and communications systems. This will result in a global information infrastructure with hundreds of thousand of subscribers and a large number of services. The development of a global information and computing infrastructure requires the effective management and utilization of resources. A number of applications require the delivery of digital information with strict time and quality constraints. For example multimedia applications require a level of Quality of Service (QoS), that defines the extend to which performance specifications such as responsiveness, reliability, availability, security and cost effectiveness may be violated.

A distinct requirement for the infrastructure support in these environments is the ability to allocate the workload placed on the cluster in such a way as to optimally use the distributed resources. Most systems providing the middleware support for the Grid environments, such as Globus [1] and Sun Grid Engine [5], rely on the centralised or hierarchical solutions for resource information reposit ories and job allocation points. In this paper, we present a decentralised, scalable, adaptive resource discovery technique for Grid environments. Using this protocol, the Grid administrators could effectively distribute the workload so as to maximise resource utilisation and availability, while not relying on a single centralised mechanism. In this fashion, resource location transparency could be effectively achieved and hidden from the application programmer/end user.

A basic service Grid environment is the resource discovery service. Users request a number of resources with specific properties in order to run their applications. The discovery service is responsible for finding those resources and forwarding the user request to the nodes that can handle it. The network capacity between those resources and their geographic distribution is also important in the decision making process of the resource discovery mechanism. In distributed applications the time spent during data exchange plays a significant role in the performance of the application. Furthermore in some case users of the grid might need their application to run close to (in terms of network delays and bandwidth) a specific geographic location.

2 Related Work

In the literature there are several proposals for resource discovery in wide-area systems. The most successful service for locating resources based on their names is DNS. DNS was designed by Paul Mockapetris in 1984 to solve escalating problems with the old name-to-address mapping system. DNS has a hierarchical structure and forms a distributed database. Its structure and caching system works well for relatively static naming data but it would not be able to cope with more dynamic information such as the resource availability of a Grid network. Furthermore DNS uses names as search criteria, while in a Grid environment the search is for a set of attribute and value. E.g. available memory more or equal to 64 MB. Web search engines return URLs based on search keys however these search engines deal with relatively static information such as web pages content. Matchmaker [4] which is Condor's [4] resource discovery component, uses an attribute-value perspective for searches using a centralized architecture. Globus uses MDS [2] as an information service for the retrieval of resource characteristics. MDS was initially centralized but it evolved to using a hierarchical structure as the number of resources available on the system grows. A. Iamnitchi and I. Foster in [3] propose a fully decentralized resource discovery mechanism for Grid environments. The mechanism is based in a peer-to-peer architecture using request forwarding algorithms. They evaluate four different request forwarding mechanisms: Random forwarding, experienced based + random, best-neighbor (based on the node that replied the maximum number of requests, and experience based + best-neighbor. They evaluate the different forwarding policies using a simulation model trying to identify the trade-offs between communication costs and performance of the system.

3 SORD: Self Organizing Resource Discovery

The solution proposed in this paper for resource discovery is a fully distributed protocol, which is based on the ideas of peer-to-peer computing [6]. Each node of the network acts autonomously in order to identify the best possible node that can serve a specific request. Each node is connected to a number of other nodes called neighbours. Initially the neighbours are the *nearest* topological neighbours and a few *far* neighbours. The use of a two level control mechanism is proposed. The fist one is query-based, while the second one is advertisement-based. The query-based mechanism will be invoked when an incoming request cannot be resolved by the node that received the request. By resolve we mean that the request receiver node has to find a node that itself has the available resources to resolve the request. When the recipient node does not have any information about the resources needed by the request, a query is sent to the neighbour nodes. Associated with each query is a query time to live (QTTL) that determines the number of times a query can be forwarded from a node to its neighbours. The QTTL will limit the traffic generated by the protocol.

The second control mechanism is advertisement-based. It is activated when a new node appears and advertises its resources. It is also initiated when a dynamic resource (such as CPU availability) is changing state, e.g. when a node becomes underutilised it can send an advertisement to all of its neighbours with information on the resource availability. Advertisements have an advertisement time to live (ATTL) that determines the number of hops that the advertisement can to be forwarded.

Information about the resource availability in the network of nodes is distributed using the query-reply and advertisement mechanisms. This information is cached in the nodes that receive the replies or the advertisements. Nodes use this information to gain experience on the resource availability. Initially when no past information is available queries are forwarded to the initial list of neighbours. When a reply is received the ID of the node that can handle the request is cached in a list of nodes that provide the requested resource, together with more specific information on those resources, e.g. CPU availability. The list of available resources is also updated each time an advertisement is received. Nodes that are not included in the future reply messages are removed from the list after a specified time period or when list exceeds a specified size.

Using a different cache for each type of resource, different overlay virtual network topologies are created for each different resource. Initially the nodes are connected to their nearest neighbours (topologically) and some random far neighbours creating a small world topology. (Small world topologies [7] allow the distribution of information, using shortcuts, to the appropriate recipients and have already been considered in the problem of routing with local information. As requests are coming to the nodes and the query and advertisement protocols are activated the resource availability caches are populated with state information. The list of neighbours is changing giving separate lists of neighbours for the different types of resources, thus generating different virtual topologies. Members of the virtual topology are nodes that have the specific resources that the virtual topology identifies; or *knows* about the requested resources. Figure 1 illustrates the concept of virtual topologies.

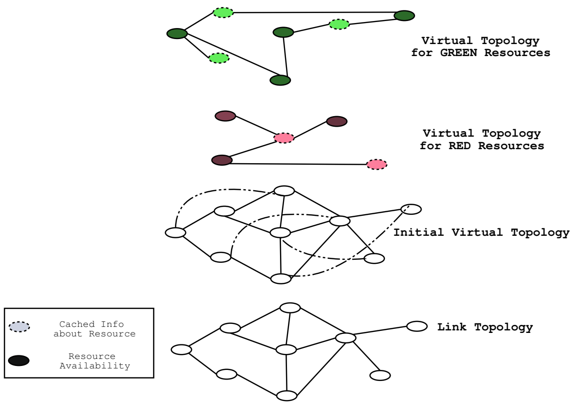

Figure 1: Virtual Resource Topologies

4 Discovering QoS

SORD has been simulated and evaluated in the scenario of a QoS enabled Grid network. A QoS enabled grid network facilitates three different classes of service, namely Gold, Silver and Bronze. Users subscribed to the Gold service are given the highest execution time guarantees. This is implemented by running Gold applications on dedicated servers. Bronze class is implemented by assigning user applications to nodes that allow only a small number of jobs to be running at the same time on a particular Grid node. Finally the Bronze service is implemented by assigning user jobs to nodes that do not implement any strict admission policy and they accept any number of jobs running concurrently. In the simulation scenario 10% of the Grid nodes implement the Gold service, 20% the Silver Service and 70% the Bronze service.

The SORD protocol was evaluated counting the number of messages required to discover the resources needed. The query specifies the CPU rating needed by the application and the class of service. Given that scenario, the protocol generated three virtual topologies, one for each of the classes of service. All three virtual topologies refer to CPU resources. A centralized, random approach was used in order to compare the performance of the SORD protocol. In the centralized approach queries where forwarded to random nodes of the network until a node was able to fulfill the resource requirements. Figure 2 illustrates the number of queries that SORD and the random approach generated. The simulation was run using three different lattice networks with 400, 1225, and 2025 nodes respectively. The random query approach generates higher number of queries compared to the SORD protocol especially in the case of GOLD service, where the number of nodes that implement that service is small. In some case the random queries perform better that

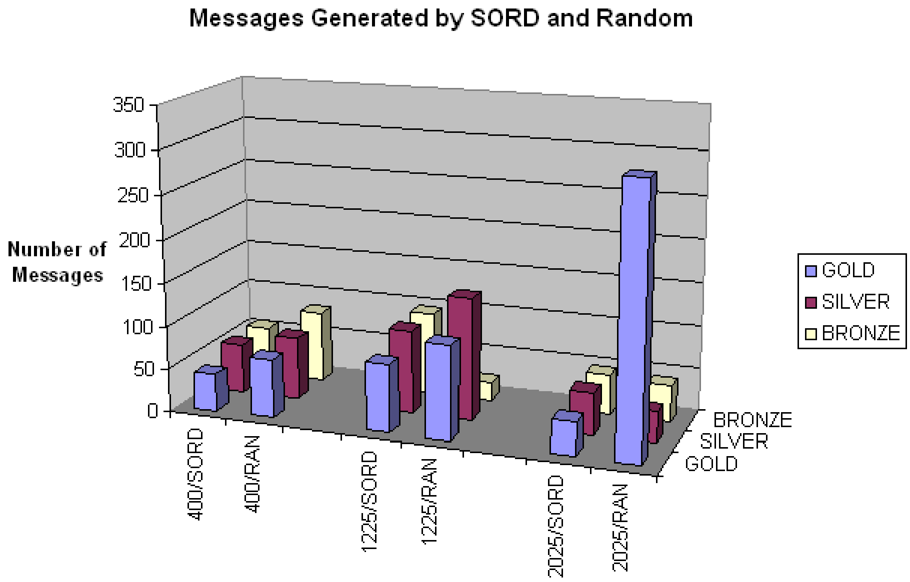

Figure 2: Number of Messages generated by SORD and Random varying the network size

SORD but this is when there is a relatively high number of available resources.

5 Conclusions

In this paper, we described an adaptive resource discovery mechanism that aids the optimal resource discovery in a farm of Grid nodes. The protocol is fully distributed. We also identify a set of rules that construct virtual overlay topologies over the existing IP connectivity of the Grid nodes. The resource discovery mechanism has been simulated in the schenario of a QoS enabled Grid network. The management traffic is used a performance measure and it is demontrated that SORD outperforms a centralised management approach.

References

[1] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *The International Journal of Supercomputer Applications and High Performance Computing* , 11(2):115–128, Summer 1997.

[2] Ian Foster. The anatomy of the Grid: Enabling scalable virtual organizations. *Lecture Notes in Computer Science*, 2150, 2001.

[3] Adriana Iamnitchi and Ian Foster. On fully decentralized resource discovery in grid environments. In *International Workshop on Grid Computing*, Denver, Colorado, 2001. IEEE.

[4] M. Litzkow, M. Livny, and M. Mutka. Condor - a hunter of idle workstations. In *8th International Conference of Distributed Computing Systems (ICDCS 1988)* , San Jose, CA,USA, 1988. IEEE CS press.

[5] Sun Microsystems. Introduction to the cluster grid - part 1 - part 2.

[6] A. Oram. Peer-to-peer: Harnessing the power of disruptive technologies, 2001.

[7] D.J. Watts. Collective dynamics of 'small-world' networks, nature, 1998.