A Multi-agent Based Efficient Resource Discovery Mechanism for Grid Systems

Yunsong TAN¹, ², †, Jianjun HAN³, Yuntao WU²

¹School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
²School of Compute Science and Engineering, Wuhan Institute of Technology, Wuhan 430073, China

Abstract

Grids are likely to be the mainstay of distributed computing. Efficient resource discovery mechanism is one of the fundamental requirements for Grid computing systems. In this article, we present a survey of recent work on the integration of multi-agent systems and peer-to-peer based resource discovery services for Grid systems. This discovery service would have several desirable features, including high scalability, high reliability, self-organization and self-healing which allows for an efficient search of resources by maximizing the benefit of the replication mechanism and facilitate the query operations. A simulation analysis has been performed to evaluate the performance of the mechanism.

Keywords: Grids; P2P; Multi-agent; Resource Discovery

1. Introduction

Since Grid hosts provide a large set of distributed and heterogeneous resources, an efficient Grid information service is a pillar component of a Grid. Grids are likely to be the mainstay of distributed computing. Current Grid information services offer centralized or hierarchical information services, but this kind of approach is going to be replaced by a decentralized one, supported by P2P [1,2] interconnection among Grid hosts. In the last years, a number of P2P techniques and protocols have been proposed to deploy Grid information services: for example, super-peer networks [4] achieve a balance between the inherent efficiency of centralized search, and the autonomy, load balancing and fault-tolerant features offered by distributed search.

Various resource discovery mechanisms have and are being developed in the paradigm of distributed systems. Goal of almost every mechanism is efficient and effective resource management in fault tolerant and scalable manner. Since in the real world of computing the underlying environment is heterogeneous and highly unpredictable therefore the mechanisms have to be optimized and some times combined for proper resource discovery and management. Grid inherits most of the properties of conventional distributed systems. Resource management in Grid has more or less same goals of other distributed systems, but with the difference that Grid is organized in much better way.

The goal of this research is to design a protocol for resource discovery in Grid applications. There are a
few well established characteristics that effective resource discovery protocols for Grid applications should demonstrate. The following characteristics have served as guidelines for the proposed protocol [5]:

Robustness: Grid applications often desire precision and accuracy. A robust protocol should locate resources when they are present and meet the expectations of the application.

Scalability: A typical Grid deployment has a vast number of resources and an effective resource discovery protocol must be scalable.

Efficiency: The protocol should locate resources while consuming a minimal amount of bandwidth. Resources on the Grid have several dynamic properties and a resource discovery scheme should be able to handle this volatility.

This paper aims at bridging the gap between conventional Grid computing and its potential application in open environments by proposing an agent-based peer-to-peer Grid computing architecture, whilst also providing reasonable compatibility and interoperability with conventional Grid systems and clients.

The rest of this paper is structured as follows: Section 2 reviews some related work for resource discovery. Section 3 introduces the agent-based resource discovery architecture and presents our resource discovery scheme using muti-agent. Section 4 evaluate the performance of the scheme. The paper concludes in Section 5.

2. Related Work

In this section we describe in detail the concept of multi-agent systems and the environments of P2P based Grid system.

2.1. Peer-to-Peer Environments and Grid

P2P search methods can be categorized as structured or unstructured. The structured approach assumes that hosts and resources are made available on the network with a global overlay planning. In Grids, users do not usually search for single resources (e.g. MP3 or MPEG files), but for software or hardware resources that match an extensible set of resource descriptions. Accordingly, while structured protocols, e.g. based on highly structured overlays and Distributed Hash Tables, are usually very efficient in file sharing P2P networks, unstructured or hybrid protocols seem to be preferable in largely heterogeneous Grids. Unstructured search methods can be further classified as blind or informed. In a blind search (e.g. using flooding or random walks ), nodes hold no information that relates to resource locations, while in informed methods (e.g. routing indices and adaptive probabilistic search ), there exists a centralized or distributed information service that drives the search for the requested resources.

The Grid computing environment presents various challenges to a query system. the system has to operate with a large number of heterogeneous nodes based on various domains i.e. different architectures and resource types. The situation is more complicated by the fact that any resource can enter and leave the system unpredictably. In addition to these conditions, lack of control and administration should also be handled by resource discovery systems. Grid resource discovery and dissemination compliment each other. The query is done by the application to find a suitable resource where as dissemination is initiated by the resource which needs to be discovered.

Various resource description seem to determine the resource discovery mechanisms in current systems.
Parameterized queries are sent to the nearest directory on the grid, Globus MDS is an example of such mechanisms[8]. There are two classifications of query based system depending on whether it’s a distributed or centralized database. Updates in resource description databases is based on the resource dissemination approach. In agent based mechanisms active code fragments are sent across the machines in the grid and those fragments are locally interpreted at each machine. The basic difference among query based and agent based mechanisms is that in agent based the agent makes resource discovery decisions based on its own logic whereas in query based systems the resource discovery is done by the predefined logic. Resource dissemination is classified by the update mechanism used to update resource information. For example to reduce data transfer and time latency the resource state information could be sent using an alternate protocol than the detailed resource description information. In the batch/approach the information is batched up on each grid machine and the disseminated periodically on the grid. The information could be disseminated in push or pull approach. If the information is sent from the originating machine then it is pushing the information and the other machine which is receiving the information is pulling the information. Two of several examples of batch approach usage are Condor [10] and European Data grid [8]. The information is immediately disseminated from originator’s end in on demand approach. The information is pushed in this approach.

2.2. Multi-agent Systems

A multi-agent system (MAS) [1] is a loosely coupled network of software agents that cooperate to solve problems that may be beyond the individual capacities or knowledge of each particular agent. In a MAS, computational resources are distributed across a network of interconnected agents. When compared to a centralized system, a MAS does not suffer from the single point of failure problem. Furthermore, a MAS has less performance bottlenecks or resource limitations. Finally, MAS efficiently retrieves, filters, and globally coordinates information from sources that are spatially distributed. To fulfill their goals, agents in a MAS need to use resources provided by other agents. To use a resource, an agent must contact the agent that provides it. However, in an open MAS, an agent does not know which agents provide which resources. Furthermore, it does not know which other agents participate in the system. A common approach to the resource discovery problem is to introduce middle agents or directories that maintain information about which agents provide which resources. Thus to find a resource, an agent has first to contact the middle agent. However, middle agents can become bottlenecks and contradict the distribution goals set by a MAS along the dimensions of computational efficiency, reliability, extensibility, robustness, maintainability, responsiveness, and flexibility. The components of MAS are as follows:

3. A Multi-Agent Scheme for Resource Discovery on Grids

3.1. Resources Mapping on the Grid

In this section the scheme is defined and discussed. The aim of this scheme is to disseminate Grid resources and spatially map them on the Grid according to their semantic classification, in order to gather a consistent number of resources of the same class in a restricted region of the Grid. It is assumed that the resources have been previously classified into a number of communities of Interests, according to their semantics and functionalities [4,10].
The scheme exploits the random movements and operations of a number of mobile agents that travel the Grid using the P2P interconnections. This approach is inspired by agent-based systems [1,5,6], in which swarm intelligence emerges from the collective behaviour of very simple mobile agents, and a complex overall objective is achieved. Each agent can pick a number of resources on a Grid host, carry such resources while moving from host to host, and mobile agent deposits them on another Grid host.

3.2. Efficient Resource Discovery Mechanisms

Each agent travels over the Grid through the P2P interconnections among Grid hosts. For the sake of simplicity, the protocol has been analyzed in a P2P network in which peers are arranged in a grid-like topology [5]: each peer is connected to neighbour peers, including horizontal, vertical and diagonal neighbours. At random times, each agent makes a random number of hops. Along the P2P network, the following query \( p, M \) specified the agent’s algorithm. The algorithm depicts the steps of the query mechanism: let \( q \) be the query to be processed, \( M \) the list of neighbours and \( r \) the result of a query processing event.

\[
\text{query}(p, M) \\
1 \quad r = \emptyset \\
2 \quad \text{if } (\neg \text{processed}(p)) \\
3 \quad r = \text{process}(p) \\
4 \quad \text{response } (r) \\
5 \quad \text{processed}(p) = \text{true}; \\
6 \quad \text{end if} \\
7 \quad \text{if } (r = \emptyset) \quad // \text{ else terminate execution} \\
8 \quad \text{randomly select } n_i \in M \\
9 \quad \text{if } (n_i \neq \emptyset) \quad // \text{ else terminate execution} \\
10 \quad n_i = \text{forward}(q, n_i) \\
11 \quad \text{if } (n_i = \emptyset) \quad // n_i \text{ will apply the same process} \\
12 \quad \text{query}(p, M-n_i) \\
13 \quad \text{else}
\]

Fig. 1 Agent Based P2P Networks
Once an agent specialized in a class gets to a Grid host, if it is currently unloaded, it must decide whether or not to pick the resources of class that are managed by the current host. The probability of picking the resources of class is defined through a pick random function; to favor the spatial mapping of resources, such probability must be inversely proportional to the number of resources of class that are currently located in the local region of the Grid. Whenever an agent specialized in a class gets to a new Grid host, it must decide whether or not to drop the resources of class, in the case that it is carrying any of them. As opposed to the pick operation, the dropping probability should be directly proportional to the relative and absolute accumulation of resources.

3.3. Component of Query Mechanisms

The scheme can be a significant step towards the design and construction of a P2P-based information system in a Grid environment. However, to better understand its role, it is necessary to discuss how can be related to the overall information system design process, which could be composed of the following three components, showed by the Figure 2.

1. classification or clustering of Grid resources;
2. replication and mapping of resources with the scheme;
3. query service.

The first component allows users to identify the features and functionalities of the resources they need. Classification of resources can be performed with different techniques, as discussed in the Section 3. The scheme assumes that resources have already been classified. The third component, the query service, assumes that the resources have been re-organized through the scheme, or at least that is working while query requests are being forwarded. The use of permits to handle resource discovery by combining the flexible and scalable features of a blind approach.

A query message first travels the Grid network with a blind/random mechanism; however, the search procedure is turned into an informed one as soon as the query approaches a low entropy region, i.e. a region which has gathered resources belonging to one particular class. In each low-entropy region a peer is elected as a representative peer. It is possible to devise a procedure that allows representative peers to exchange information with each other and allows for the construction of an overlay network connecting representative peers. This way, a query can be routed towards the nearest representative peer, regardless of the class of resources in which it is specialized. Then, this representative peer could re-route the query to another representative peer which is specialized in the class of resources under interest, or simply send an informative message to the requesting peer. When a query finally gets to the proper representative peer, it will easily find a high number of useful results.

3.4. Resource Management and Scheduling Mechanisms

Grid uses resource matrices to track the status of computing resources. It defines the type of resource, where the resource resides, and the resource’s status (called profile) or the description of the resource. A peer’s local resources are registered by the profiling agent when the peer starts. The profiling agent also
updates the profiles of the local resources when they change. The following sample shows a definition of
the processor profile. When a module requires a resource, its container may match the required resource
with those in the resource matrix first. If none of them matches the requirement, the container starts a relay
process.

```
{ capability
    {Machine type
        Processor type
        Contributed memory amount
        Contributed cycles amount
        Contributed storage amount}
    Load
    {modules #1,
        modules #2 ...}
}
```

Fig.2 the Components of Query Mechanisms

Alternatively, the container may start the replay process immediately upon receiving the module’s
request. How the container behaves is determined by the type of resource. For example, local service
resources have precedence over remote service resources, but there is no such difference in terms of
processor resources. During the relay process, the participating peers look up the required resource in their
resource matrices. If matching resources exist, the references to these resources are returned. If multiple
replies exist, container starts a resource selection process to determine which resource is the most suitable
one. The outcome is then returned to the module for its subsequent operations. And if the type of resource
located has local precedence, it will be cached in the resource matrix.

A resource matrix only caches a limited number of references. Each time a cached reference is retrieved,
it is regarded as “updated”. The least updated entry will be removed if the cache is full and a new reference
comes in. Once the reference to a resource is obtained, the resource is accessible to the module through
Grid. Each time a resource is accessed, its reference is quoted and passed to the resource’s residing peer R.
Then R will perform the actual operations and send the results back to the module. When the module
finishes using the resource, it notifies R so that the resource can be released on R.
A peer also keeps records of other peers which have cached references to its local file, service, and module instance resources, so that the references can be updated when the actual resources migrate to other peers. In Grid, files can be uploaded to the backbone nodes through CMS. Unlike other resources, local files never appear in the resource matrix. When a file is located and used, it can be cached by the peer that uses the file, if there is sufficient storage on that peer.

The executables of any modules are regarded as files, and need to be uploaded to the Grid before the execution of the module. Grid has a two-stage scheduling mechanism. Once a module is uploaded, its residing peer Q will trigger a relay process, informing other peers of the potential workload. Other peers will reply to Q if they can execute the module. Q then determines the suitability of these peers (including Q). The module will be moved to the winner if the winner is a backbone node; otherwise it is transferred to the winner and cached there. At the second stage, when the module is about to be created, a relay process will be started to locate the module. Once it is located, it will be scheduled and executed by its residing peer. Table 4.12 shows the reference to the instance of the module used in the procedure calls. The only difference in a service’s execution process is that the service has to be discovered before its module execution process. Figure 3 demonstrates that process.

![Fig.3 Service Invocation Process](image)

### 4. Performance Evaluation

In this section, we discuss some relevant simulation results which demonstrate the effectiveness of the resource discovery scheme in a Grid environment, if it is used in conjunction with the resource mapping protocol. Figure 4 depicts the value of the mean frequency of operations evaluated at different times while the mapping process proceeds. In particular, the present section introduces the main system and protocol parameters. The performance of the scheme was analyzed by evaluating the performance indices defined and explained. This figure confirms that representative peers are selected almost exclusively with condition.

The simulation parameters and the performance indices used in our analysis. The number of peers was varied from 50 to 20000. Each peer generates one or more agents with a given probability. However, a higher activity of agents also causes an increase in the traffic load (Figure 5). A correct setting of the value
of agents should take into account the trend of these performance indices and in general should depend on system features and requirements, for example on the system capacity of sustaining a high traffic load.

![Fig.4 Mean Frequency of Operations](image)

![Fig.5 Mean Traffic Load (hops/s), with Different Numbers of Agents](image)

5. Conclusion

This paper introduces an approach based on multi-agent systems for building an efficient information system in Grids. We have presented a survey on multi-agent based resource discovery and have further described our approach for a resources discovery process in P2P based Grids systems. A number of self-organizing agents travel the network by exploiting P2P interconnections; agents replicate and gather information. Such a logical reorganization of resources is exploited by resource discovery scheme, which collects a large number of resources having the desired characteristics. Simulation analysis shows that, as the reorganization of resources proceeds, allows users to discover more and more resources in a shorter amount of time, without increasing the traffic load experienced by Grid hosts.

Acknowledgement

The work described in this paper was supported by a grant from the Research Plan Project of Hubei Provincial Department of Education (No. Q20091501).
References


