Evaluating Distributed Methods of Resource Discover in Ultra Large Scale Systems

Zahra Zanbagh

Department of Computer, College of Engineering, Yazd Science and Research Branch, Islamic Azad University, Yazd, Iran

ARTICLE INFO

Article history:
Received 25 June 2014
Received in revised form 1 July 2014
Accepted 31 August 2014
Available online 15 September 2014

Keywords:
Weighted tree – resource discovery – bitmap – cluster computation – ultra large scale system (ULSS)

ABSTRACT

Recently Software Engineering Institute has presented the Order of America Department of Defense as associated with the development of IT. The office of the Assistant Secretary of the U. S. Army funded the Software Engineering Institute to lead a report on the current challenges of ultra-large-scale (ULS) systems software. There are many heterogeneous resources in the ULS which are distributed geographically. We will see that for other algorithms. Resources discovery that satisfy users’ requests, is an important task. In this paper we propose a comparison of previous algorithms using simulations and results and show that the number of nodes visited in our resource discovery algorithms is less than that for other algorithms.

© 2014 AENSI Publisher All rights reserved.


INTRODUCTION

The office of the Assistant Secretary of the U. S. Army funded the Software Engineering Institute (SEI) to lead a 12-month investigation of ultra-large-scale (ULS) systems software. It posed this question to the SEI: “Given the issues with today’s software engineering, how can we build the systems of the future that are likely to have billions of lines of code?” (SEI, 2006). Although a billion lines of code was the initial challenge, increased code size brings with it increased scale in many other dimensions, posing challenges that strain current software foundations. Scale of ULS systems will change everything and make it impossible to rely on our current knowledge and techniques of software development. The scale of complexity and uncertainty in ULS system design will be so great as to resist treatment by traditional development methods, which are characterized by centralized control (true even of decentralized methods).

Such as open-source development and by the testing of a small number of hypotheses about what constitutes a good solution. ULS systems will be deeply embedded in the real world. These systems will comprise not only information technology (IT) components, but also machines of many kinds, individuals and teams, diverse sensors, information streams and stores (including verbal and non-verbal human communications), and so forth. We have traditionally viewed software as programming the computer components of such systems. We face a challenge in understanding and designing software in a new way.

One of the challenges that we will face in developing ULS systems is Design and Evolution. The scale of complexity and uncertainty in ULS system design will be so great as to resist treatment by traditional development methods, which are characterized by centralized control (true even of decentralized methods).

Such as open-source development and by the testing of a small number of hypotheses about what constitutes a good solution. The ULS systems are one of great developments in the field of engineering and computer science and provide a clear future in the global use of various optimal distributed resources (hardware and software) (Azim Sharifloo, 2008). Therefore with expanding systems and the importance of finding suitable resources for users, while saving time and space, resource discovery algorithms are very important. Users are not interested in where resources actually are. Just by giving a description about resources they desire, the resource discovery mechanism will find a set of resources that match the user’s description if there exists one.

The motivation behind this paper is to explore the resource discovery mechanism, which is suitable with the ULS Environments. That is, the resource discovery should find out the preferred resources quickly and return the result back to one who requests in the manner time. Consequently, we collect the information about the previous work related resource discovery approaches. Although this paper does not propose new successful resource discovery mechanism, we expect this survey could make us know about the exist facility that could
lead us to achieve our goal more or less. The remainder of this paper is organized as follows: Section 2 presents the concept, property and construction of a resource discovery algorithm. Section 3 explains traditional and distributed algorithms of resource discovery. Section 4 presented tree algorithms. Simulation experimental results and concluding remarks are given in Section 5. According to the simulation results, the amount of network traffic and on occupied link in the tree algorithms compared to other techniques was improved.

Design criteria for resource discovery techniques:

The rapid growth of scientific applications leads to new computational infrastructures, like the Grid expansion (Foster et al., 2008). ULS system is a large scale distributed environment which provides a high number of powerful resources to its users (Foster and Kesselman, 2004). It differs from classical distributed systems by their heterogeneous and dynamic nature. While resources are nearly identical and stable in a classical distributed system, in grid systems resources are highly heterogeneous and dynamic in terms of both stability and dynamicity of their properties.

Resource Discovery (RD) in ULS can be defined as searching and locating resource candidates, which are suitable for a job in which processing environments’ constraints are clearly specified. On the other hand, the RD problem is defined as realizing the RD in a reasonable time, considering the dynamicity and large scale of the environment. In this perspective, several methods have been proposed to solve the RD problem in ULS systems. They can be classified into three main categories (Hameurlain et al., 2008): methods based on centralized/hierarchical systems, methods based on P2P systems and methods based on agent systems. (Antonioletti et al., 2005) In most developments, centralized or hierarchical indexing mechanisms were proposed for the RD process. (Elmroth and Tordsson, 2005; Moltó et al., 2008; Ramos and Magalhaes, 2006; Yu et al., 2003). Hence, these methods were poorly adapted to the large scale and dynamic nature of the ULS in which nodes can leave or join the system at any moment. Most of those methods (Kaur and Sengupta, 2007) could not be adapted to today’s large-scale environments. Scalability and dynamicity in ULS restrict the usage area of centralized or hierarchical systems. Other approaches to use in Grids were investigated to overcome these problems. Therefore, new areas are researched with this scope in mind, and for some services, especially RD, researchers focused on agent and P2P systems to evaluate their capabilities on a Grid platform.

Resource discovery approaches:

Resource Discovery is systematic process of determining which grid resource is the best candidate to complete a job with following trade-offs.

• In shortest amount of time
• With most efficient use of resources
• At minimum cost

Resource discovery is challenging issue for efficient deployment of a ULS system. Dynamic availability and heterogeneous nature make it challenging task. There are various Approaches for resource discovery in ULS environments. The base for all these Approaches is query and agent based resource discovery. They are classified into three main categories in the literature (Hameurlain, 2008) methods based on centralized and hierarchical systems, peer to peer (P2P) based systems, and agent based systems. Centralized and hierarchical systems emerged as suitable approaches which provide easy to access tools for Grid services (Antonioletti, 2005). In such systems, resource information is stored and updated in central or hierarchically located servers, and resource discovery is realized by querying these servers.

Comparison between the Resource Discovery Approaches based upon various features including scalability, reliability, adaptability and manageability has been done and placed in Table I.

| Table I: Qualitative Comparison for above Approach. |
|---------------------------------|---------------------------------|---------------------------------|---------------------------------|
| Hierarchical | Peer-to-Peer | Centralized | scalability |
| Better scalable because of the hierarchical distribution of load | More scalable as it uses the four axes framework | Not scalable due to bottleneck problem | scalability |
| Reliable in terms of query correctness, better reliable in terms of single point of failure | Based on graph theory so reliability increases | Reliable in terms of query correctness but not reliable in terms of single point of failure | reliability |
| Tolerant to node dynamicity, better tolerant to indexing mechanism's dynamicity | Easily Extendable | Tolerant to node dynamicity, but not tolerant to indexing mechanism's dynamicity | dynamicity |
| Manage the consistency by using the data dissemination algorithms | Complex architecture hence difficult to manage. | Quite easy to manage as a lot of its working is dependent on single node. | manageability |

Resource Discovery Algorithm in ULS Systems:
This section gives an overview of technologies and algorithms related to this paper. When users want to execute jobs in a ULS environment, they have to find appropriate resources first. Resources in the ULS system often change frequently as new resources and services are added to the system, or old ones are changed. Therefore, characteristics of resources and services will change. There are many resource discovery mechanisms proposed.

One of the methods in the resource discovery uses semantic communities. Such a method (Foster, 2001) which proposes a resource discovery method based is on semantically linked virtual organizations. Its framework organizes a grid with semantic links and replaces similar nodes in one group as a semantic small world in order to make the resource discovery easier.

Qi et al. (Clematis, et al., 2005)[10] uses a table which saves the information of resources. Users use this table to find their requested resources. In (Chang and Hu, 2010), Ramos and de Melo propose a structure of master and slave. A master does the updating and and the slave restores the information from the machine.

Chang and Hu (Chang, 2010) proposed Resource Discovery Tree using bitmaps (Lei and Jiuyang, 2005). In this algorithm, two bitmaps are used. Firstly, the local resource bitmap which specifies the local resources present in that node. Index bitmap which specifies the resources present in the children of a node. A logical AND is performed between the user’s request and the local resource bitmap. If the result is non-zero then the request is forwarded to the children nodes. (Leila mohammad khanli, 2011) uses a tree structure in which the edges have weight. The advantage of this method is that any node in weighted tree has a unique path, so the user’s query against all of previous methods is not sent to the extra and unnecessary paths. it can directly reach the target node using a resource address which is stored in nodes. Furthermore, for resource discovery they only use one bitmap in every node which is for the storing of information about its local resources and the resources of its children and descendant. Also it preserves a address of resources and if they need a resource which is available in its offspring or descendant, they can directly and without any referring to unnecessary and extra nodes, reach the target node. This method significantly reduces the system traffic and increases the performance of system

Simulation environment:

In this section, experimental setup with the obtained simulation results is demonstrated. The experiment was performed in MATLAB environment and the resources and requests were randomly distributed among the nodes. The present experiment was performed on a weighted tree and compared with other approaches.

As in Mastroianni et al. (Mastroianni, et al., 2005) and Chang and Hu (2010) also it was assumed that the resource discovery tree for the simulation had the height of 3 or 4.

In the first experiment tests, (Leila mohammad khanli, 2011) compare algorithm (FRDT) with a resource discovery tree with a different number of index servers. Like [11], in this experiment there are 200 nodes in resource discovery tree and we perform this experiment with 100 queries. We place the resources randomly in each node and then queries are sent through tree paths and compare the number of nodes that visited in each method. In Fig. 1, the difference between the numbers of visited nodes with two methods is observed. In this experiment, the number of visited nodes is investigated with changing the number of index servers. For example, when the number of index servers in tree is 10, so there are 10 nodes in level 1 and 190 nodes in level 2 (19 children for each node in level 1) for a tree with height 3. Because FRDT in the forward path just visits one node in every level so in Figs. 1 and 2, the simulations related to FRDT almost show the fix values. In the second experimental tests, we assume there are 300 queries and a tree with height 4. In Fig. 2, (Leila mohammad khanli, 2011) compare the number of nodes that queries send in the FRDT algorithm with the resource discovery tree and show that the number of nodes visited in the FRDT method is lower than the previous method. In the fourth experimental tests, FRDT is compared with flooding-based method, MMO and resource discovery tree algorithm. In the current experiment supposing that there are 300 queries, in Fig. 3, it is indicated that the average number of nodes that queries are sent to is lower than other methods in FRDT. The experiment is performed in a tree with height 4.

![Fig. 1: The number of nodes that queries are forwarded to for 100 queries.](image-url)
Conclusions and future works:

Resource Discovery is the process of finding the satisfactory resources according to the user’s request, including resource description, resource organization, resource lookup and resource selection. In this paper we analyzed various grid resource discovery Approaches. We compare all these Approaches on the basis of performance factors like scalability, reliability, adaptability and manageability. On the basis of this comparison we have an idea about choosing an appropriate approach to discover a particular resource. Our assumption said that peer-to-peer approach with tree algorithms has succeeded in the world of resource discovery. We expect the resource discovery concepts of tree may be adapted in ULS systems. However it is not easy to recognize the ULS system with peer-to-peer system, so we should explore the new model continuously which can be describe the components which are sufficient for both systems.

REFERENCES

Azim Sharifloo, Amir, Mehdi Mirakhorli, and Fereidoon Shams, 2008.“How could ULS systems achieve architecture benefits?.” In Proceedings of the 2nd international workshop on Ultra-large-scale software-intensive systems, pp: 41-44. ACM.,


