Best Resource Node Selection in Grid Environment

Zar Lwin Phyo
University of Computer Studies, Mandalay
zarlwinphyo@gmail.com

Abstract

Grid computing can provide abundant resources to assist many studies and it can be employed in more and more projects that have so far yielded numerous findings. However, in grid environment, many distributed resources are integrated in different sites and resources are often shared by multiple user communities and can vary greatly in performance and load characteristics. Therefore, a central problem in grid environment is the selection of computation nodes for execution. Automatic selection of node is complex as the best choice depends on the user request structure as well as the expected availability of computation and communication resources. This paper considers the ongoing research on Rough Sets based selecting the best resource node in computational grid. Our node selection method is based on the CPU capacity, communication bandwidth and memory and disk availability on the resource nodes. The goal of this paper is to enable the automatically select the best available resource node on the network for execution within a reasonable time. In this paper, random data is used to test the validity of proposed method. The result showed that this method can provide the suitable probability of the best resource selection.

1. Introduction

Grid enable the *sharing*, *selection* and *aggregation* of a wide variety of geographically distributed resources including supercomputers, storage systems, databases, data sources, and specialized devices owned by different organizations[5]. The resources are heterogeneous in terms of their architecture, power, configuration, and availability. They are owned and managed by different organizations with different access policies and cost models that vary with time, users, and priorities [6].

Application of grid computing can be roughly divided into two types: computation grid and data grid [5].Computational grid mainly solves problems with demand for CPU computation over long period of time. Many servers, workstations, supercomputers and personal computers distributed in various locations are effectively integrated to construct a virtual supercomputer. In grid computing, software or current settings of machines do not have to be adjusted. They only have to be installed with software packages, such as Globus Toolkit [3], so as to be equipped with the ability to participate in and share resource of the grid system. Data grid is similar to computation grid. But it is mainly focus on the storage of and access to massive amount of data for scientific computation [7].

In grid environment, the abilities of computing resources are vary, and task often arrive dynamically. Therefore how to select an optimal resource for a given job among heterogeneous and dynamic sharing resources is critical importance for grid computing. When a user submits a job, the grid environment needs to

search the appropriate resource node from many different resource nodes.

In this paper, we proposed a resource selection framework that is intended to identify an optimal resource for a given application. When a user submits a job request, the node selection mechanism will accorded the characteristics of request to search a suitable storage node. In other words, the resource nodes can be arranged effectively by our proposed method to maintain the consistency of resource nodes and to access data effective.

2. Related Works

In this section, we are going to mention some related research in resource selection in distributed environment. Many of the researchers have proposed resource selection based on various criteria such as, complexity, cost, time, while some are focused on an agent-based selection approach. Broadly speaking, the optimal resource selection problem is addressed by either embedding application specific information in the selection module, or targeting particular classes of applications, or utilizing load- or performance-based selection policies [4]. The author in [7] proposed a mechanism of storage node selection to reduce the cost for data grid environment. This is driven by storage capability, bandwidth and CPU capability. In [1], the authors used agent based architecture to achieve resource discovery. The agent returns one or more resources that are closely match the request. They consider the load index and the memory availability of resource for resource selection.

3. Rough Set

Rough sets theory was introduced by Zdzislaw Pawlak in 1982 [8]. It extends classical set theory. It has often proved to be an excellent tool for the analysis of vagueness and uncertainty inherent in making decisions. It also proved to be very useful for analysis of decision problems concerning objects described in a data table by a set of attributes and by a set of decision attributes. The incoming data table as a whole can be created as an information system. This *information system* consists of *universe*, *attributes*, *values of attributes*, and a *function* [9]. Symbolically, an information system is represented as: $S = < U,Q,V, f >$ where the universe U is the collection of objects that are of interest to us. Q is the collection of all attributes and can be divided into two disjoint, nonempty subsets—C and D—where C is the collection of all condition attributes and D is the collection of all decision attributes. Any subset of the universe $X_i \subseteq U$ is called a *concept* or a *category*. Any family of concepts is referred to as *knowledge* about U. In rough set theory, a set of all similar objects is called an elementary set, which makes a fundamental atom of knowledge. Any union of elementary sets is called a crisp set and other sets are referred to as rough set.

4. Framework for Node Selection

Our framework for node selection includes an interface for jobs to specify their execution requirements and will examine partial result for the resource selection problem using rough set theory in the next section.

4.1. Job Specification Interface

For automatic node selection, the structure and processing requirements of job are the driving inputs. This knowledge can be difficult or impossible to discover automatically, hence it is important to allow the job to specify these characteristics. The information that is transferred through this interface includes the

following: relative priority of communication and computation, needed processor speed, size of needed memory, average time needed for execution. This is an important component of our automatic node selection framework and select the one or more resource node that are closely match the job request.

4.2. Proposed System Architecture

Architecture of proposed system is illustrated in Fig.1. When user submits a job, broker tries to find the best node for this job. Firstly, broker send request message to modeler and collector. When modeler received this message, it query to node's information from node database. This node database contains up to date dynamic information of all node by using collector agent. In this paper, we consider the load index, communication network and availability of the memory at the time as dynamic information. According to this information, modeler elects the suitable nodes and sends this node group to broker. Finally, broker chooses the best node from this group by using rough set theory and assigns the job to the best node.

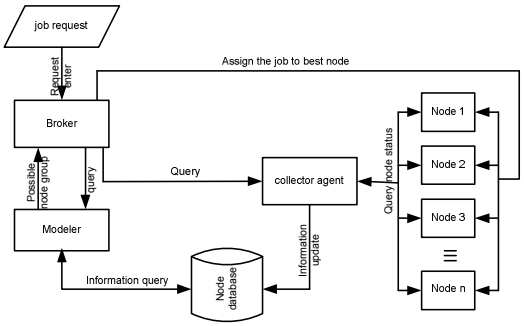

Figure 1. Overview of proposed system architecture

4.2.1. Collector Agent

In this paper, the aim of collector agent is to collects the related information of all the resource nodes in the grid when the request message is received from broker. After all this information is collected, it updates the node database.

When the collector agent is launched, collector agent will collect related information of each node participating in this grid, such as CPU utilization, communication bandwidth, CPU speed, memory and disk availability, etc.

4.2.2. Modeler

When the request message is received from broker, modeler queries the nodes' information from node database. After suitable nodes are elected, which are satisfied the size of job, modeler send this group to broker. Modeler will use the following criteria for the selection of best resource for the I/O request (1), the resource characteristics must be as much as possible with the client's requirements, (2), and the dynamic memory space of resource must meet the user request.

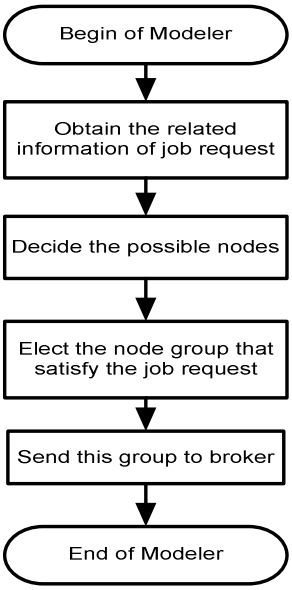

Figure 2. Process of modeler

4.2.3. Broker

When a job is submitted to the broker, then the broker needs to choose the best node from suitable node group using rough sets. After that broker assign the job to best node.

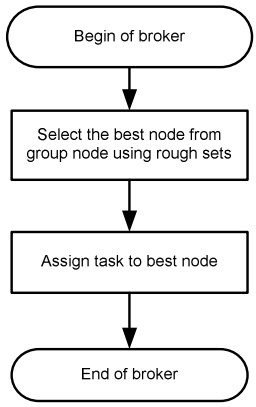

Figure 3. Process of broker

5. Example

In this section, an example is given to discuss the proposed method. In our example, we consider only three dynamic attributes such as storage capability(SC), network bandwidth(NB) and CPU capacity(CC). The information of all storage nodes is as shown in Table 1.

In this example, we assumed that between 1M to 10 GB of storage capacity is 0.25, 11GB to 30GB is 0.5, 31 to 50 is 0.75 and greater than 50GB is 1. Less than 5Mbps of network bandwidth is assumed that 0.3, 5Mbps to 10 Mbps is 0.6 and greater than 10 Mbps is 1. In CPU capacity, less than and equal to 300 MHz is assumed 0.5 and greater than 300 MHz is 1.

Table 1. The data of resource node

Node'ID	SC(GB)	BW(Mbps)	CC
1	25	2	266
2	89	10	450
3	60	8	300
4	9	12	200
5	17	2	500
6	30	1	450
7	10	8	200
8	6	2	266
9	18	1	400
10	42	2	200
11	40	8	300
12	9	10	500
13	17	12	266
14	23	10	450
15	15	12	200
16	18	8	300
17	6	2	500
18	4	1	266
19	73	8	450
20	3	10	266
21	35	12	500
22	46	8	300
23	36	2	200
24	18	10	400
25	20	12	266

When a job is submitted to broker, it delivers this request to modeler. When modeler received this message, it query the information of all nodes and sent suitable nodes with their information to broker. Broker do rough sets on suitable node group based on obtained information for job request. For rough sets analysis, we defined importance of each attribute is defined by:

$$I(X) = \frac{\left|\underline{R}(x)\right|}{\left|\overline{R}(x)\right|} \qquad (1)$$

Where $\underline{R}$ is defined by:

$$\underline{R}(x) = \{x \in U : B(x) \subseteq X\}, \quad (2)$$

And $\overline{R}$ is defined by:

$$\overline{R}(x) = \{x \in U : B(x) \cap X \neq \phi\}, \quad (3)$$

where x is the attribute of N_i. Then $S(N_i)$ is computed by:

$$S(N_i) = \sum (I_i \times A_i) \quad (4)$$

Where I_i is the important of attribute and A_i is the value of each attribute.

In this example, there are two users A and B. Assume user A and B sends the job request sequentially. The information includes the size of job. The size of job for user A is 23G and 35G for user B.

When user A submits the job request, modeler collects the possible resource nodes based on the storage capacity and send this group to broker. This is shown in Table 2. Broker chooses the best node from this node group using rough set.

According to Table 3, storage node'ID 21 is chose as best node for user A because node 21 got the maximum value of S.

Table 2. Possible node for user A

Node' ID	SC(GB)	BW (Mbps)	CC (MHz)
1	0.5	0.3	0.5
2	1	0.6	1
3	1	0.6	0.5
6	0.5	0.3	1
10	0.75	0.3	0.5
11	0.75	0.6	0.5
14	0.5	0.6	1
19	1	0.3	1
21	0.75	1	1
22	0.75	0.6	0.5
23	0.75	0.3	0.5

Table 3. Result for user A

Node' ID	SC (GB)	BW (Mbps)	CC (MHz)	S
1	0.42	0.3	0.38	1.1
2	0.85	0.6	0.76	2.21
3	0.85	0.6	0.38	1.83
6	0.42	0.3	0.76	1.48
10	0.64	0.3	0.38	1.32
11	0.64	0.6	0.38	1.62
14	0.42	0.6	0.76	1.78
19	0.85	0.3	0.76	1.91
21	0.64	1	0.76	2.4
22	0.64	0.6	0.38	1.62
23	0.64	0.3	0.38	1.32

Table 4. Possible node for user B

Node' ID	Storage Capacity (GB)	Band width (Mbps)	CPU capacity (MHz)
2	1	0.6	1
3	1	0.6	0.5
10	0.75	0.3	0.5
11	0.75	0.6	0.5
19	1	0.3	1
22	0.75	0.6	0.5
23	0.75	0.3	0.5

Table 5. Result for user B

No de' ID	Storage Capacity (GB)	Band width (Mbps)	CPU capacity (MHz)	S
2	0.65	0.6	1	2.25
3	0.65	0.6	0.5	1.75
10	0.49	0.3	0.5	1.29
11	0.49	0.6	0.5	1.59
19	0.65	0.3	1	1.95
22	0.49	0.6	0.5	1.59
23	0.49	0.3	0.5	1.29

According to Table 5, the broker chooses the best node for user B is node 2.

6. Conclusion and Future Work

In this paper, resource selection method using rough set theory is suggested. Currently, we have considered the way of best node selection and also use random data to test the validity of proposed method and then the result showed that this method can provide the suitable probability of the best node selection.

In the future, the proposed design mechanism will be implemented using Simulator and evaluated under the simulation environment.

References

[1] A. Sarhadi and M.R. Meybodi, "New Algorithm to Resource Selection in Computational Grid Using Learning Automata", International Journal of Intelligent Information Technology Application, 204-208, 2009.

[2] Liu, L. Yang, I. Foster, and D. Angulo, " Design and evaluation of a resource selection framework for grid application framework for grid applications", Proceedings of the 11th IEEE international symposium on high performance Distributed computing, P-63, Washington,DC, USA, IEEE Computer Society, 2002.

[3] Globus Toolkit, http://www.globus.org/toolkit.

[4] J. Seung-Hye, T. Valerie, W. Xingfu,P. Mieke, D.Ewe, M.Gaurang and V. Karan, "Performance prediction-based versurs Load-based site selection: Quantifying the Difference", Proc.of PDCS-2005, Las Vegas, Nevada, 2005.

[5] M. Baker, R, Buyya and D. Laforenza, "Grids and grid technologies for wide-area distributed computing," Software: Practice and Experience, Vol,32, No.15, pp.1437-1466,2002.

[6] R. Buyya, M. Murshed and D.Abramson, "A Deadline and Budget Constrained Cost-Time Optimization Algorithm for Scheduling Task Farming Applications on Global Grids,". International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, Nevada, USA, June 2002.

[7] S.C. Wang, H.J. Lyu, K.Q. Yan, M.L. Chiang and S.S. Wang, "Enhancing the Quality of Data Grid Service with Node Selection Mechanism", International Conference on Advanced Information Technology (AIT), 2010.

[8] Z.Bonikowski, E.Bryniarski and U.W.Skardowska, "Extensions and intensions in the rough set theory," *Journal of Information Sciences*, 107,149-167-1998.

[9] Z.Pawlak, "Rough Sets," International Journal Computation and Information Science, 11,341-356, 1982.