Resource Discovery using PageRank Technique in Grid Environment

Noorisyam Hamid, Fazilah Haron and Chan Huah Yong
School of Computer Sciences,
Universiti Sains Malaysia
11800 Minden,Penang, Malaysia
{syam, fazilah and hychan}@cs.usm.my

Abstract

The grid deals with large scale and ever-expanding environment which contains million of users and resources. For this reason, resource selection has been a challenging task especially in meeting user's demand for a quality of service (QoS). A quality of service is the ability to serve a job by providing quality and reliable resource in fulfilling the user's need. Quality and reliable resource selections naturally yield excellent and quality results. The background of the users and where the resource belongs to are important in determining the quality of a resource. This paper concerns with efficient and quality-based resource discovery using Condor ClassAd and PageRank technique in order to achieve a quality resource matching. The paper discusses how quality of users and resources are determined and considered in the discovery process prior to allocating jobs to resources.

1. Introduction

Grid technology provides facility which enable sharing of a large scale, distributed and heterogeneous computing resources [1]. Hence, an efficient resource discovery mechanism is necessary to meet user's demand on quality of service (QoS).

Most grid resource discovery or schedulers [6, 12, 13] focus on selecting and allocating jobs to suitable resources by mapping the requirements and constraints between users and resources. None considers the background and quality of the users and resources involved. Users may end up with low quality or inconsistent resources leading to disappointing results. They may play tricks by requesting more resources than required. This can lead to unfair competition in getting resources among users.

This paper emphasizes on improving the current resource discovery technique in grid scheduler by taking into account the quality and reliability of both, users and resources. We begin our discussion by positioning our work in relation to existing work (section 2) and followed by a discussion on how quality and reliable users and resources are identified (section 3). We then proceed with detail discussion on how to incorporate PageRank. Section 4 summarizes and suggests the future direction of the work.

2. Related Work

2.1 The World Wide Web vs. Grid

Resource in the World Wide Web (WWW) are the web pages which comprises of text- and hypertext-based entities that are independent and uncontrolled [3]. In grid environment, resources are represented as different form of entities such as computing power, databases, files, applications software and storage. We focus on compute resource which is used to perform and execute any submitted jobs. In order to enable resource sharing in grid, authentication is needed and only authorized user is allowed to participate. This is why the grid is referred to as a well-controlled environment.

Google search engine is one of the most popular WWW search engine which sorts and returns the search results based on quality and reliable web pages. The search engine employs an interesting technique called PageRank [2, 3, 4] to rank the large number of web pages based on importance. This is done by utilizing the link structure of the web page [2, 3, 4]. Google's PageRank is a numeric value that represents the importance of a page on the web. A page has a high rank if the sum of its backlinks is high. We believe the PageRank idea can be applied in grid by treating the resource usage as the link structure to identify the quality and reliability of the resource.

Google search engine uses meta-search as a matching technique where user put their keywords in the search box to find specific information [12]. User's keywords are then sent to the search engine's index server to extract relevant information based on user's

query. In grid environment, there are various matching techniques which are used in the grid scheduler for the purpose of discovering resource [6, 12, 13]. Condor ClassAd is one of the matching technique currently implemented in Condor scheduler. It is a semi-structured data model or language that can be used to specify the characteristics, constraint and preference of principles [6, 8]. In other words, ClassAd is a symmetric attribute-based matching technique. Figure 1 shows the example of ClassAds of requestor and provider. In Condor, a matchmaker is used to discover compatible ClassAd between principals (providers and requestors). Unfortunately, ClassAd only evaluates the physical characteristics/specifications of the machine and job, but ignores the quality of the principals.

```
Request ClassAd:
[ Type = "Job"; Owner = "User1";
  Constraint = other.Type == "Machine"
  && Arch == "INTEL" && OpSys == "SOLARIS251"
  && Disk >= 10000; Rank = other.Memory;]

Resource ClassAd:
[ Type = "Machine"; Name = "m1"; Disk = 30000;
  OpSys = "SOLARIS251"; ResearchGrp = "user1",
  "user2";
  Constraint = member(other.Owner,
  ResearchGrp)
  && DayTime > 18*60*60;
  Rank = member(other.Owner, ResearchGrp)]
```

Fig.1. Two examples of Condor ClassAds. For each resource-request pair, constraint clauses are checked for compatibility against the other's properties. Rank is used to select among multiple matches.

3.0 Our Approach

Our Quality-based Grid Resource Discovery *(Q-GreD)* aims at providing a grid resource discovery which takes into account the quality and reliability of both users and resources. We propose a resource discovery technique which is based on Condor ClassAd but incorporate the idea of PageRank (which we refer to as ResourceRank) in determining the quality and reliability of the grid resources.

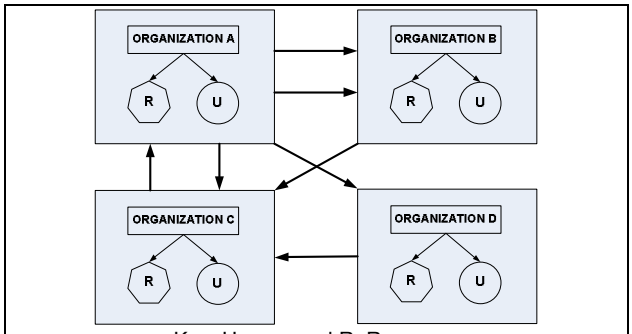
In PageRank, if a page links to another page this means that it is casting a vote as an indication that the other page is good. If many pages link to a page then that page has more votes and its worth should be higher [2]. Similar idea can be applied to grid where submission of jobs to resources indicates that the resource is good. Resource can obtain higher ResourceRank score, if many users from different organizations submit jobs to that resource or there exist users with high ResourceRank using the resource. In general, this is true for stable and reliable resources which are consistent in providing QoS. Furthermore, these resources tend to have many users and more jobs. Different from PageRank approach where 'importance' is tied to each individual web page [3], the ResourceRank is computed by adding up ResourceRank score of each resource provided by the organization.

$$RR(A) = (1-d) + d\,(RR(T_i)/C(T_i) + RR(T_n)/C(T_n))$$

Where
$RR(A)$ - ResourceRank of resource A,
$RR(T_i)$ - ResourceRank of resource T_i which uses the resource A,
$N(T_i)$ - the number of times user in organization T_i submits job to current resource in an organization
d - damping factor which usually set to 0.85

Fig.2. Resource Rank Algorithm (adapted from [2])

Figure 2 shows the algorithm used to calculate the rank score. ResouceRank is actuallty a backlink calculation [2, 4] where in order to obtain the score of a resource, we need to get the ResourceRank score of users that currently vote or use that resource. Since users' votes can influence the ResourceRank score, there is a possibility of an organization plays a trick by asking the same user to keep submitting jobs into their resources. In order to prevent this problem, we modify the PageRank algorithm by including a new variable $N(T_i)$. If the same user submits jobs into a resource, the user's ResourceRank score will be divided by the number of times that user submits jobs to that particular resource. Figure 3 illustrates how ResourceRank is calculated.



$RR(A) = 0.15(base) + 0.1275(C) = \mathbf{0.2775}$
$RR(B) = 0.15(base) + 0.1275/2(A) = \mathbf{0.2135}$
$RR(C) = 0.15(base) + 0.1275(A) + 0.1275(B) + 0.1275(D) = \mathbf{0.5325}$
$RR(D) = 0.15(base) + 0.1275(A) = \mathbf{0.2775}$

Fig.3. Example calculation of ResourceRank score

Assuming that there are four virtual organizations; A, B, C and D with one user and one resource each. Two arrows from organization A to B implies user from organization A has submitted two jobs to

resource in organization B while an arrow from organization D to C implies user from organization D has submitted a job to resource in organization C. ResourceRank score of organization A is obtained by calculating the backlink of other user's ResourceRank; in this case is the user in the organization C who submits jobs to resource in the organization A. The base value is the initial value of each resource where there is no incoming vote or outgoing vote. User's ranking value depends on the ResourceRank of the organization. Organization C provides the most quality and reliable resources since there are many users' votes for resources in organization C. The final calculation indicates that the higher the ResourceRank score, the quality and the more reliable the resources provided by the organization.

3.2 Matching Technique in Q-GReD

ClassAd relies on the following expression to perform matching and find similarity between requestor and provider.

```
1.  Constraint = attribute type which
                 principal would like to be
                 matched

Where,
Similarity (a, b) = {True if a=b else false}
```

Constraint is used by the matchmaker to validate every attribute value whether there are any similarities between provider and requester. It will return 1, if attribute values on the provider are similar and return 0 if not.

```
2. Rank = preferences
```

Requestor and provider are considered as matched when both are evaluated to TRUE. Rank is used when Condor finds more than one principal's ClassAd which met the constraints. In other words, rank denoted the goodness of the candidate.

We incorporates ResourceRank into the rank equation in Condor ClassAd. Hence, ResourceRank becomes a new constraint that must be considered when matchmaking is performed. The matchmaking are carried out according to the following steps:

1. The discovery and matchmaking is strictly based on ResourceRank value on each principal.
2. Requestor with high ResourceRank has a high possibility to claim and to be placed into quality and reliable resource.
3. For the case of requestor, if ResourceRank = *N* then the claim will only allowed to provider with ResourceRank = *N* and below.
4. For the case of provider, the claim is valid and accepted to any requestors which have the same ResourceRank value or above.

Requestor interacts with *Q-GReD* by placing their requirements or specifications on targeted resources as constraints. Figure 4 represents the modified ClassAds with ResourceRank for each requestor in organization A and B and also provider's ClassAd in organization C and D.

```
Request ClassAd:
[ Type = "Job"; Owner = "UserA";
ResourceRank = "10"; Organization = "A";
Constraint = other.Type == "Machine"
&& Arch == "INTEL" && OpSys == "SOLARIS251"
&& Disk >= 10000; Rank = other.Memory +
other.ResourceRank;]

Request ClassAd:
[ Type = "Job"; Owner = "UserB";
ResourceRank = "4"; Organization = "B";
Constraint = other.Type == "Machine"
&& Arch == "INTEL" && OpSys == "SOLARIS251"
&& Disk >= 10000; Rank = other.Memory +
other.ResourceRank;]

Resource ClassAd:
[ Type = "Machine"; Name = "m1"; Disk = 30000;
OpSys = "SOLARIS251"; ResourceRank = "6";
Organization = "C";
Constraint = other.Owner,
&& DayTime > 18*60*60;
Rank = other.ResourceRank]

Resource ClassAd:
[ Type = "Machine"; Name = "m2"; Disk = 30000;
OpSys = "SOLARIS251"; ResourceRank = "3";
Organization = "D";
Constraint = other.Owner,
&& DayTime > 18*60*60;
Rank = other.ResourceRank]
```

Fig.4. Requestor's Requirements

In *Q-GReD*, user in organization A is only allowed to request resource from organization C and D. The reason being the ResourceRank of user A is = 10 which is over than ResourceRank for the organization C and D. The idea is to utilize the resource where quality resource is only for quality job. User B which has lower ResourceRank than user A is only allowed to request resource from organization D. Our approach is better than static approach implemented by Condor where a fair allocation can be ensured among the competing users. In Condor, after the matchmaker

finds the compatible ClassAds, the system proceeds with claiming process which is based on the priority factor (set by provider) [7, 8, 10]. We found that the priority factor is not a good metric of the fairshare since the distribution of the priority is depend on the resource provider. In our work, ResourceRank replaces the priority factor and plays an important role in distributing the priority.

4.0 Conclusion and Future Work

In this paper, we have presented *Q-GReD*, a quality-based resource discovery technique in grid environment. The main goal of the *Q-GReD* is to provide a better resource discovery in a challenging large-scale grid environment. *Q-GReD* concerns with selecting quality and reliable resources for job allocation. Future work includes implementing the *Q-GReD* technique and studying other potential matchmaking algorithms.

5.0 Reference

[1] Foster, I. and Kesselman, C. *Globus: The Grid: Blueprint for a Future Computing Infrastructure*. Morgan Kaufmann, 1999.

[2] Chris Ridings and Mike Shishigin, *PageRank Uncovered*, 2002.

[3] Sergey Brin and Lawrence Page, "The Anatomy of a Large-Scale Hypertextual Web Search Engine". In *Proceedings of the 7th World Wide Web Conference*, Brisbane, Australia, 1998. http://www-db.stanford.edu/~backrub/google.html

[4] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. "The pagerank citation algorithm: bringing order to the web". In *Proceedings of the seventh conference on World Wide Web*, Brisbane, Australia, April 1998.

[5] Douglas Thain, Todd Tannenbaum, and Miron Livny, *Condor and the Grid*, in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, Grid Computing: Making the Global Infrastructure a Reality, John Wiley, 2003. ISBN: 0-470-85319-0

[6] Todd Tannenbaum, Derek Wright, Karen Miller, and Miron Livny. *Condor - A Distributed Job Scheduler*, in Thomas Sterling, editor, Beowulf Cluster Computing with Linux, The MIT Press, 2002. ISBN: 0-262-69274-0

[7] Nicholas Coleman, Rajesh Raman, Miron Livny and Marvin Solomon, *Distributed Policy Management and Comprehension with Classified Advertisements*, University of Wisconsin-Madison Computer Sciences Technical Report #1481, April 2003.

[8] Rajesh Raman, Miron Livny, and Marvin Solomon, "Matchmaking: Distributed Resource Management for High Throughput Computing". In *Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing,* July 28-31, 1998, Chicago, IL.

[9] Alain Roy and Miron Livny, "Condor and Preemptive Resume Scheduling", *Published in Grid Resource Management: State of the Art and Future Trends*, Fall 2003, pages 135-144, Fall 2003, Edited by Jarek Nabrzyski, Jennifer M. Schopf and Jan Weglarz, published by Kluwer Academic Publishers.

[10] Rajesh Raman. *Matchmaking Frameworks for Distributed Resource Management*. PhD thesis, University of Wisconsin, October 2000.

[11] Ding Choon Hoong and Rajkumar Buyya. *Guided Google: A Meta Search Engine and its Implementation using the Google Distributed Web Services.* Technical Report, GRIDS-TR-2003-1, Grid Computing and Distributed Systems (GRIDS) Laboratory, The University of Melbourne, Australia, January 2003.

[12] R. Buyya, D. Abramson, and J. Giddy, "Nimrod/G: An Architecture for a Resource Management and Scheduling System in a Global Computational Grid", *4th Intl. Conf. on High Performance Computing in Asia-Pacific Region (HPC Asia 2000)*, China.

[13] *Portable Batch System Administrator Guide*. Release: OpenPBS 2.3, Printed: August, 2000.