DISTRIBUTED RESOURCE INFORMATION IN GRID FOR EFFICIENT RESOURCE DISCOVERY

Gagandeep Kaur[1], Major Singh Goraya[2]
Sant Longowal Institute of Engineering and Technology, Longowal, Punjab, India
Email:[1]ergaganlongowal@gmail.com, [2]mjrsingh@yahoo.com

Abstract— Grid resources show intermittent availability during application execution. As a result the performance of grid applications declines. To execute the applications with a minimum level of performance and reliability in grid, efficient resource management and scheduling strategies are needed. This paper presents an approach to distribute the resource information for efficient resource discovery in grid environment. Grid resources' information which is earlier stored in a single centralized resource information server (RIS) is distributed among multiple sub servers based on the application domain. This strategy is inherently characterized with the efficient failure recovery of RIS as well as sub servers. Performance comparison of the presented approach with the existing is given in the paper.

Index Terms—Distributed resource information in grid, resource discovery in grid, resource server in grid, reliability

I. INTRODUCTION

Grid has large no. of computational, storage, informational and database resources. In parallel processing divides the each individual task into no. of sub tasks and each sub task is managed separately. Scheduler manages and allocates the resources to different jobs meeting their requirements by using appropriate resource information [1]. Grid computing aims at large scale applications in different domains as it has the inherent capability of parallel processing and large scale distribution
[2]. In grid computing, enormous heterogeneous and dynamic resources are networked together to constitute a large scale computing infrastructure. The enormous computing power of grid makes it suitable to execute the complex computing applications/tasks in the distributed environment. Grid resources work together to solve different computing problems as per their complexity and requirements. Large number of grid resources creates a complex distributed computing infrastructure which needs efficient and intelligent resource management as well as resource discovery techniques.

A number of resources may join and leave the grid intermittently. As a result, the resource information in grid needed to be updated regularly. Hence, resource management and scheduling are the challenging. Each time a resource changes its state (leaves or join) in grid, centralized RIS needs to be updated to include latest resource information. The centralized RIS may become saturated to regular resource information updation in a larger grid. Users of different domains submit their jobs for execution to grid by the interface known as grid portal and collecting the results from grid portal after successful completion of processing of jobs. RIS keeps record of all the information related different resources present in the grid to serve the users belonging to different application based domains. Hence it makes the RIS more complex as it stores resource information and update that information very frequently [3].
This paper presents an approach to distribute the RIS among distributed sub servers. The main

objective of distributing resource information in grid is to discover the best resource in terms of low delay, low turnaround time and high throughput. Rest of the paper is organized as. Next section includes background and motivation. Section III presents the proposed approach. Section IV includes then simulation scenario and result discussion. Last section presents the concluding remarks.

II. BACKGROUND AND MOTIVATION

The concept of Grid is described as a readily available computational power which is efficient and secure with less human interaction [4]. It is aimed to deliver comprehensive access to geographically distributed hardware, software, information resource [4]. The techniques used for resource scheduling and management in literature are able to schedule and manage the resources, but many shortcoming are left. In fault tolerant scheduling system [3] have the problem of centralized server leading to single point of failure. Papers [2][5] has the lack of incorporation of transient failures. In papers [6][7] only mean response time is considered as the QoS measure.

In this paper, a distributing resource information based scheduling approach is presented to efficiently discover and schedule the resources without affecting the performance of network. Failure of RIS is recovered by joining the information from different sub servers. Network performance in grid is quite important. The presented approach includes both the solutions. It resolves the failure of RIS and other sub servers with low overhead of resource discovery in grid, because searching time is reduced by distributing the information among different application domain based servers.

III. DISTRIBUTED RESOURCE INFORMATION APPROACH

Distributing resource information based technique is used to avoid the impact of failure of RIS in efficient discovery of resources and scheduling in grid. Arranging the information among different sub-servers has many advantages when RIS fails. It can be easily recovered from sub servers by integrating their information.

In this approach, first servers are organized into various different application based domains. The information of RIS is distributed on the basis of different application domains. Each server stores information related to particular application based resources. When a request is received on a scheduler it first looks for the application domain and then forward the task to appropriate sub server. Sub server then provides the information about the idle resource that can fulfill the requirements of applications to the resource manager. Application is scheduled on the idle resources for execution. Grid portal is used as interface between the user and grid to submit the task to grid and get the results from grid. Resources which want to join the grid first register themselves with the grid and these resource related information is also updated in RIS and sub server. Grid portal will send a job request to scheduler, scheduler schedule the job in grid and collect the result from particular resource after successful completion of job and send to grid portal.

Steps:

1. Grid resources are registered with RIS. If any new resource joins the grid first it registers with RIS.
2. RIS then distribute the resources related information among different sub-servers.
3. User submits jobs with required QoS parameters to the grid portal.
4. Resource scheduler sends the request to RIS for the available resources information.
5. Resource scheduler sends the request to the different sub servers for available resources.
6. Sub servers send their list of resources to the scheduler.
7. Scheduler submits the job to suitable resource for processing.
8. After the completion of processing the results are return back to the scheduler.
9. Scheduler sends result to the user.
10. In case if RIS and Replica of RIS fails then servers are easily recover by merging the sub servers. See Fig. 1.
11. In case if any of the sub server fails then it could also be recovered from RIS. See Fig. 1.

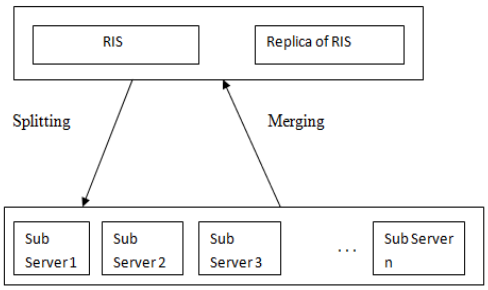

Fig. 1. Merging and Splitting (Modified Proposed approach)

Functions of the scheduler:

Grid resources register with RIS,
RIS store resource information related to resources.
RIS distribute resource information among different
domain based sub servers.
Resource scheduler receives n jobs from grid portal with Qos:
For all jobs starting from first
{ Request a list of suitable resources for jobs in sequence
from the sub servers
Receives list of suitable resources from sub servers.
Submit job to suitable resource. }

IV. SIMULATION AND RESULT DISCUSSION

The impact of presence of failure using single RIS is simulated. Two sub servers are considered for experimentation.

A. *Experimental Setup and Metrics*

We designed our own simulator which was used for the experiments.

1.The jobs and resources pattern

The jobs and resources pattern file was randomly generated using exponential distribution.

2. Performance Metrics

The following metrics were used for performance evaluation-

a) *Turnaround time*: Turnaround time is the one of the most important metrics to determine the performance of the system. It is the time interval between job submission and job completion.

b) *Throughput*: Throughput is defined as: total number of jobs/total time to complete n jobs.Where n is the total number of jobs submitted.

V. RESULTS

Simulation results are plotted in Fig.2. and Fig.3. Results show that the presented resource discovery approach is better than the centralized RIS approach. In our experiments for the validation of results we have designed our own simulator. The existing centralized RIS architecture is modified into distributed resource information server. In the existing approach of scheduling system, scheduling is done by scheduler with the help of resources information which is provided by resource information server. There are some drawbacks in the existing architecture that there is only single centralized resource information server. RIS can store all the information related to grid resources. On which continuous updates, searching and other operations are done, which makes server more complex. On the other hand, if the server fails then there is no way to recover the server because whole the information was lost and we cannot recover the server without this information. So this results in the failure of the whole system. In modified architecture the existing problems are overcome as we have divide the main server into different sub servers based on different types of applications. Then we distributed these sub-servers among different application based domains so in case if main server failed we can easily recover it by merging the sub servers together. On the other hand if one or more sub servers fail then they are recovered with the help of main server. The job is first submitted to the scheduler it can check the status of the main server and assign the job to the main server if it does not fail. If it fails then scheduler can check the application or job type and assign job to the particular sub server. In our experiment we have consider two types of applications and divide the main server into two sub servers. One is business sub server that contains the information of resources related to business applications and another server consist of resources related to military applications. In this we get the high performance than the existing one which can be shown as follows:

After a number of simulations, the following results were gathered. Based upon these results, a detailed analysis is presented.

A. Turnaround time:

The turnaround time increases with increase in number of jobs submitted. Fig. 1 shows that the turnaround time of the proposed system is better than the other scheduling system for different number of jobs. As the number of resources increases the delay time will increase and thus the turnaround time for executing applications will increase.

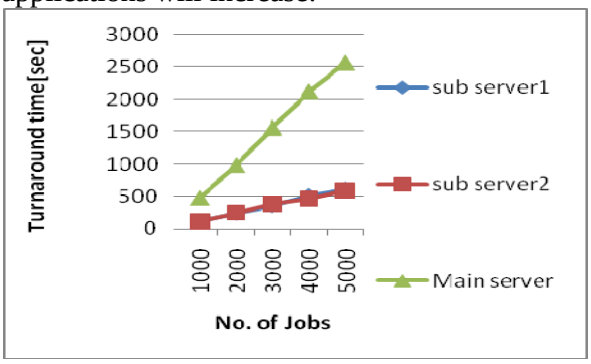

Fig. 1. Comparison on basis of Turnaround time

B. Throughput:

Throughput is another important metrics used to evaluate the performance of the system [3]. Number of Jobs are 1000, 2000, 3000, 4000, and 5000.Throughput of both the system decreases with increase in number of jobs. Fig. 2 shows that the throughput of the proposed system is better than the existing system.

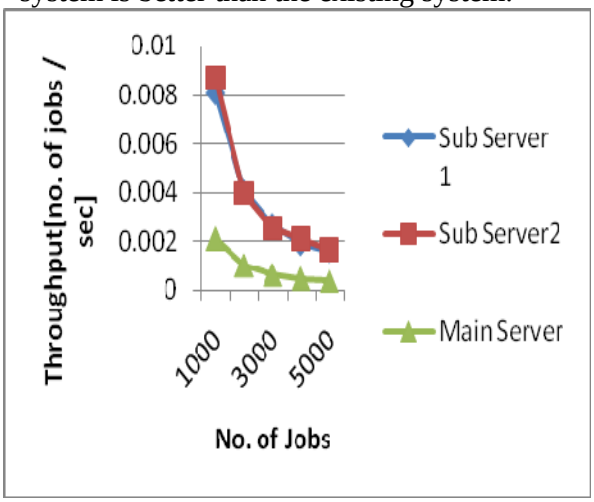

Fig. 2. Comparision on basis of Throughput

VI. CONCLUSION

In this research, the centralized RIS is divided among different sub servers for efficient resource discovery in grid. The performance of the application execution in grid using the proposed approach of resource information is compared with the existing centralized RIS. The performance metrics proves that the proposed approach is better than the previous approach. The proposed approach effectively schedules jobs in the presence of failure of main server.

In future research, results will be collected by taking different performance metrics.

REFERENCES

[1] H. Lee, L. Chung, S. Chin, J. Lee, D. Lee, S. Park, and H. Yu, "A resource management and fault tolerance services in grid computing," *J. Parallel Distrib. Comput.*, vol. 65, pp. 1305-1317,2005.

[2] H. Jin, D. Zou, H. Chen, J. Sun, and S. Wu, "Fault- Tolerant Grid Architecture and Practice," *Comput. Sci. & Technol.*, Vol. 18, pp. 423-433, 2003.

[3] M. Amoon, "A fault – tolerant scheduling system for computational grids," *Computers and Elect -rical Engineering*, vol. 38, pp. 399-412, 2012.

[4] Manish Parashar,Senior member, IEEE and Craig A. Lee, Member, IEEE, "Grid Computing: Introduction and overview"

[5] Y. S. Dai, X. L. Wang, "Optimal resource allocation on grid systems for maximizing service relia -bility using a genetic algorithm," *Reliability Engineering and System Safety*, vol. 91, pp. 1071-1082, 2006.

[6] P. Kokkinos, E. A. Varvarigos, "Scheduling efficiency of resource information aggregation in grid networks," *Future Generation Computer Systems*, vol. 28, pp. 9-23, 2012.

[7] S. S. Sathya, K. S. Babu, "Survey of fault tolerant techniques for grid," *Computer Science Review*, vol. 4, pp. 101-120, 2010.