International Journal of Soft Computing and Engineering (IJSCE)
ISSN: 2231-2307, Volume-3 Issue-3, July 2013

A Novel Approach for Grid Service Reliability Modeling Optimal Task Scheduling Perceiving Fault Recovery

Nana Kacharu Zalte, Roshan M. Pandav

Abstract*— Since last few years, grid technology has come into sight as a significant tool for solving and computing high intensive problems from different area. Grid reliability analysis and modeling are not easy tasks because of the complexity and large scale of the system. While concerning on large scale system, large subtasks requires time-consuming computation, consequently the reliability of grid service could be rather low. Our paper tries to focus on this reliability and task scheduling in the grid. In the existing system all researchers focused on the remote node fault recovery where greater waste is consumed on time and resource. Furthermore those systems did not incorporate the fault recovery and the practical constraints of grid resource on optimization. Resultantly our paper considers the Local Node Fault Recovery mechanism into grid systems, and presents a solution to simultaneously maximize the grid service reliability modeling and analysis with this kind of fault recovery thereby minimizing the cost. Our proposed Grid Service Reliability & Node Recovery (GSRNR) mechanism considers some practical, some constraints such as the life times of subtasks, the numbers of recoveries performed in grid nodes, and thus grid service reliability models under these practical constrictions are developed. Presuming the proposed grid service reliability model, a multi-objective task scheduling optimization model is presented, and Min Max scheduling algorithm is developed to solve it effectively.

***Index Terms*— Grid Computing, Fault Tolerance, Grid Service Reliability, Local Node Fault Recovery**

I. INTRODUCTION

In global grid systems contain various organizations that integrate or share their resources on the global grid. Any program running on the grid can use those resources if it can be successfully connected to them and is authorized to access them. It enables the aggregation and sharing of geographically dispersed computational data and other resources as a single, integrated resource for solving large-scale compute and data intensive applications. The client of the grid can access the resources available in the grid without bearing in mind about heterogeneous environment.

Still user can get information in the form of abstracted and the resources of the grid are virtualized form.

Management of these resources is an important infrastructure in the grid computing environment. And the procedures for a program to use the remote resources are controlled by the RMS (Resource Management System). Since grid resources are highly heterogeneous and dynamic, more faults may be raise in grid environment. Fault tolerance mechanism is an important in grid computing environment. Therefore it is important to note that the grid infrastructure should be designed to be fault tolerant.

Manuscript Received July, 2013.

Mr. Nana Kacharu Zalte, Computer Engineering Department, Government College of Engineering , Aurangabad, India.

Mr. Roshan M. Pandav, Computer Engineering Department, University of pune , India.

Fault tolerance is the capability of a system to achieve its function properly even in the existence of faults.

The RMS works in five layers that are program layer, request layer, management layer, network layer, and resource layer, respectively.

Description of Grid Computing

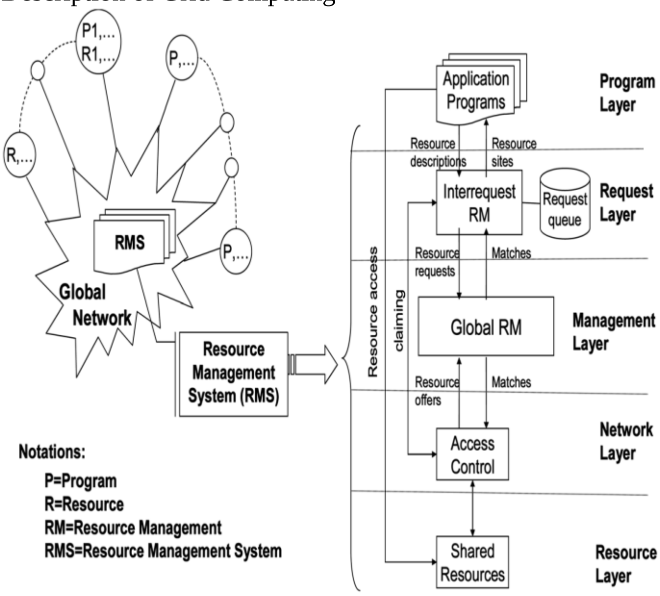

Fig.1: Grid System

The global grid system is depicted by Fig. 1 and how different resources are accessed by RMS is shown in figure1. Hence, when using the resource, resource failures can be caused by software faults, hardware faults, or a combination of both. By considering all the failures mentioned above, the grid service reliability can be defined by: Grid Service Reliability. The probability that a set of programs contained by a grid service can be successfully completed: In particular, the requested resources by the programs are matched correctly and in time, the programs succeed in connecting and communicating to those resources, and both programs and resources are reliable when working.

A. Theoretical Foundation and realization:

Different resources are distributed in the grid system and different services can use a given set of resources. Each resource is directly connected to the RMS by a single communication channel, which forms the star topology. The service task consists of subtasks that should be executed by resources and each subtask is characterized by fixed complexity, and by fixed amounts of input & output data. The request for service (task execution) arrives to the RMS which assigns the subtasks to different resources for processing. The resources are specialized and can serve only a single subtask if it is available. On the other hand, the same subtask can be assigned to several resources of the same type for parallel execution.

If the same subtask is processed by several resources, it is completed when first output is returned to the RMS. The entire task is completed when all of the subtasks are completed, and their results are returned to the RMS from the resources. Some subtasks require outputs from previous subtasks for their execution. Therefore, if resource failure or communication channel failure occurs before the end of the output data transmission from the resource to the RMS, the subtask cannot be completed. We have set some predefines rules and guidelines for performing on fault tolerance system in grid environment that are as follows;

Rule 1) When the RMS receive all the necessary information for execution of subtask, it should sends the information to the corresponding resource immediately.

Rule 2) Each resource should starts the processing of the assigned subtask immediately and each resource sends the output data to the RMS

Rule 3) Each resource should have constant processing speed, constant failure rate and constant bandwidth when it is available.

Rule 4) it considers the indices service reliability that is task is accomplished within a specified time.

B. Realization of GSRNR

- **Grid service reliability and Node Recovery** can be defined as the probability of all of the subtasks involved in the considered service to be executed successfully.
- There is no concern on what the sources of failures are; but what matters is whether the end results can return to grid resource management system (RMS) or not.
- Remote Node Fault Recovery (RNFR) tool; i.e., when a failure occurs on a node, the state information can be migrated to another node, and the failed subtask execution is resumed from the interrupted point.
- In a worst-case scenario, much time has been spent in local node execution when the execution is terminated by a failure, which brings great waste of consumed time and resource on using RNFR.
- Local Node Fault Recovery (LNFR) tool could be more practical than RNFR to resume the subtask execution on the failed node once the node is recovered.
- The migration expense compared with RNFR is saved with LNFR. Moreover, because fault recovery modules are located at grid resources, resource providers can set customizable constraints on fault recovery, which makes it easy to achieve distributed management of fault tolerance.

Our paper considers Grid service reliability and Node Recovery tool (GSRNR) in grid systems, and presents a solution to simultaneously maximize the grid service reliability modeling and analysis with this kind of fault recovery thereby minimizing the cost.

II. LITERATURE SURVEY

As one of the most important aspects of quality of service (QoS), grid service reliability can be defined as the probability of all of the subtasks involved in the considered service to be executed successfully [6], [7].

Tian-Liang Huang et al. [1] put forward a Fault Tolerance Policy on Dynamic Load Balancing (FTDLB) in P2P grids to improve the dynamic availability of resources. This policy duplicates jobs in different computing nodes to avoid job or hardware failure. In the meantime, the fault tolerance policy also considers the load balancing among different computing nodes while keeping the stable job turnaround time. FTDLB can tolerate the node's permanent failures while balancing load of real-time applications on P2P grids.

Recently, the modeling and analysis of grid service reliability has attracted lots of attention. Dai et al. [2] presented a virtual approach to modeling grid services, and derived the grid service reliability using the graphic theory. Dai et al. [8], and Levitin and Dai [7] studied grid service reliability for grid systems with star topology, and tree topology, respectively. Dai et al. [9] presented a hierarchical model from the mapping of the physical architecture, and the logical architecture in grid systems for grid service reliability analysis.

Levitin et al. [6] studied grid service reliability taking the precedence constraints on programs execution into account. From the point of view of grid service, it does not matter what the sources of failures are; what matters is whether the end results can return to grid resource management system (RMS) or not. Nevertheless, with the dramatic increasing of grid size and complexity, the grid system is much more prone to errors and failures than ever before. Moreover, the likelihood of errors occurring may be exacerbated by the fact that many grid services will perform long tasks that may require several days of computation [10].

Recently, much effort in fault avoidance and fault removal has been invested so as to improve grid service reliability. Paul and Jie [10] developed an approach to fault tolerance based on job replication in grid systems.

Affaan and Ansari [11] introduced a backup mechanism to achieve fault tolerance in grid systems. Jin et al. [12] put forward a fault tolerance mechanism in grid systems based on Java threads state capturing, and Mobile Agents.

Jozsef and Peter [13] introduced the concept of job migration to achieve fault tolerance in grid systems.

Moreover some researchers have studied the optimization on grid service reliability. Dai and Wang [16] studied optimal resource allocation for maximizing service reliability using a genetic algorithm.

Dai and Levitin [17] suggested an algorithm to study optimal resource allocation for maximizing performance while considering the service reliability factor in tree-structured grid systems. However, those works did not incorporate fault recovery, and did not investigate the influence of practical constraints of grid resources on optimization.

III. PROPOSED SYSTEM

The basic approach proposed in the above researches on fault recovery in grid systems is a "Grid service reliability and Node Recovery" (GSRNR) mechanism in which we are focusing Remote Node Recovery and Local Node Recovery if any fault present. When a failure occurs on a node, the state information can be migrated to another node, and the failed subtask execution is resumed from the interrupted point, or the failed subtask can be dynamically rescheduled on another node, and the node restarts the subtask from the beginning. It is very useful and effective for GSRNR to recover grid tasks from failures. However, some complex tasks may require several days of computation. For those tasks, it will take a lot of time for GSRNR on the transmission of state information. Furthermore, in a worst-case scenario, much time has been spent in local node execution when the execution is terminated by a failure, which brings great waste of consumed time and resource. In this case, we can concentrate on Local Node Recovery (LNR) could be more practical than Remote Node Recovery (RNR) to resume the subtask execution on the failed node once the node is recovered. LNR offers an opportunity to resume execution from failure, and saves the migration expense compared with RNR. Moreover, because fault recovery modules are located at grid resources, resource suppliers can set customizable constraints on fault recovery, which makes it easy to achieve distributed management of fault tolerance. However, with the introduction of LNR, the state of resource failures may be divided into unrecoverable failures, and recoverable failures.

The grid service is divided into some subtasks; The RMS should quickly and effectively schedule those subtasks to the appropriate nodes according to the particular requirements of those subtasks, and the QoS demands of grid users. In the scheduling, it needs to take into account not only the hard constraints of a subtask (the processing capacity , link bandwidth available CPU, memory, disk space, etc.), And the software constraints such as the demanded reliability level of grid service, and the constraints on total financial cost should be considered for scheduling.

Different from traditional distributed computing environments (DCEs), the RMS does not have complete control over all the resources in grid systems. Even though all online nodes, or resources, are linked through communication links with one another, only a small portion of nodes or resources available for a specific grid service is discovered by the RMS. At the same time, through systems selection, the RMS normally selects more than one resource from the discovered resources to assign a subtask to, so that the grid service reliability can be improved. In the case that there is only one RMS in the grid system, it can approximately regard the RMS and the selected resources as a star topology. The proposed system has some limitations;

(a) The RMS is perfect during the processing of the grid service, i.e., the RMS never fails; and the time of task processing by the RMS is negligible when compared with subtask's processing time.

(b) When a service request arrives at the RMS, the RMS responds to it immediately; when a subtask is assigned to a node, the node executes the subtask immediately.

(c) There is no precedence constraint on the order of execution of subtasks.

(d) Each node can execute only one subtask at any time.

(e) The failure processes of nodes and communication links can be modeled by Poisson processes, respectively

(f) The failures in different elements (nodes or communication links) are independent.

A. GSRNR FRAMEWORK AND DESIGN:

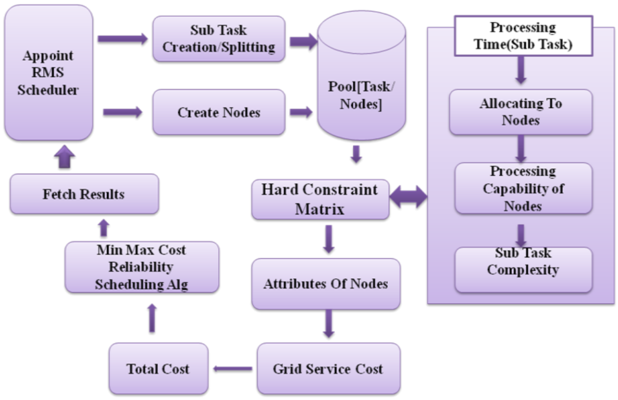

Fig 2: System Architecture

- **Task /Nodes Creation**

Management of complex, distributed, and dynamically changing job executions is a central problem in computational Grids. These executions often span multiple heterogeneous resources, cross administrative domains, and need to adjust to the changing resource availability to leverage opportunities and account for failures or policy induced failures. The executions themselves are dynamic. They often start other computations or have requirements exceeding resources that were originally allocated. In general the grid environment framing is carried out in this model having API;

- The task is created and spitted into sub task.
- Those sub tasks are collected and placed in pool.
- These tasks are used for scheduling.
- Using GridSim the nodes are created and organized.
- Those sub task are allocated in the nodes.

- **Hard Constraints Matrix**

For ease of describing the recoverability of hardware failures on node, a random variable is defined, which has two possible values (1, 0).

If $X_j^k = 1$

It means that the failure on grid node is recoverable. If , $X_j^k = 0$, it means that the failure on grid node k is unrecoverable.

One subtask is allowed to be assigned at one node, and one node can only be allowed to execute one subtask at most. The hard constraints on subtask scheduling have two possible values (1, 0); **value 1** if, it means that subtask can be allowed to be allocated on node; **value 0** if, it means that subtask cannot be allowed to be allocated on node.

- **Grid Service Reliability (GSR):**

When the RMS receives all the outcomes of subtasks, the grid service is considered to be completed successfully.

- **Scheduling:**

Decision is taken regarding the Total Cost, arrival Time of Jobs, Grid Service Reliability Cost. The scheduling criteria are obtained via the heuristic information obtained. Applying the algorithm (Min Max Cost Reliability Scheduling algorithm) the task are scheduled.

B. Min max cost reliability scheduling Algorithm

- for all tasks T_i in meta-task M
- for all resources R_j
- cost constraints are evaluated
- Selection probability of Jobs are evaluated => $Pr_{ik}=C_{ik}+R_{ik}$

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

- do until all tasks in M are mapped
- – for each task in M find the minimum probability of the jobs and the resource that obtains it
- – assign task T_k to the resource R_l that gives the earliest completion time
- – delete task T_k from M
- – Update Status of Resource Ri
- End do

C. Advantages of the Grid Service Reliability & Node Recovery Technique

- Maximizes the grid service reliability
- Minimizes the total cost of execution
- Increased high availability
- Improved efficiency

IV. PERFORMANCE EVALUATION

By going through these our proposed system, we got following results;
Construct a grid environment with a possible set of nodes. The RMS divides the service request into subtasks. When scheduling these subtasks, the hard and the soft constraints are considered. For each of the task in the grid system, grid service cost is calculated and then the summation of the grid service cost is taken as the total cost. Finally, Optimized Cost Scheduling Approach is developed for minimizing the cost and maximizing the grid service reliability. This approach ranks the nodes in the grid system in order to determine the fault tolerant cost on each node so that the future loads can be assigned to those nodes. Using GridSim tool, the following analysis is made and the values are generated with respect to the constraints considering failures.

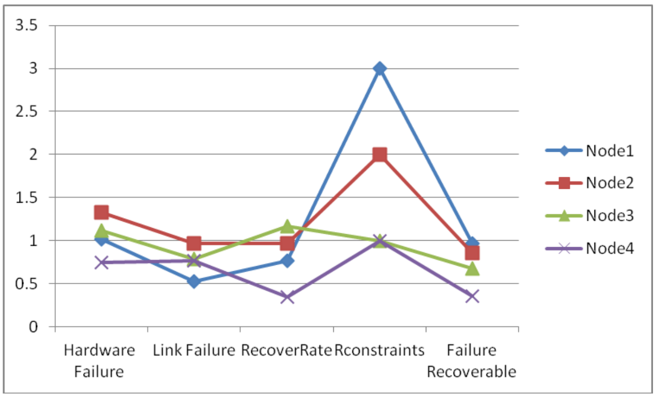

Fig 3: Attributes of grid node

Fig. 3 shows the attributes of grid node. Given graph display the hardware & software failure intensity & recoverability of nodes.

TABLE 1. **Attributes of grid node**

Attributes	Node1	Node2	Node3	Node4
Hardware Failure	1.02	1.33	1.12	0.75
Link Failure	0.52	0.97	0.79	0.77
RecoverRate	0.77	0.97	1.17	0.34
Rconstraints	3	2	1	1
Failure Recoverable	0.97	0.86	0.68	0.35
Processing Capability	11	21	29	39

Table 1 depict the hardware, software, link &Processing capability of nodes

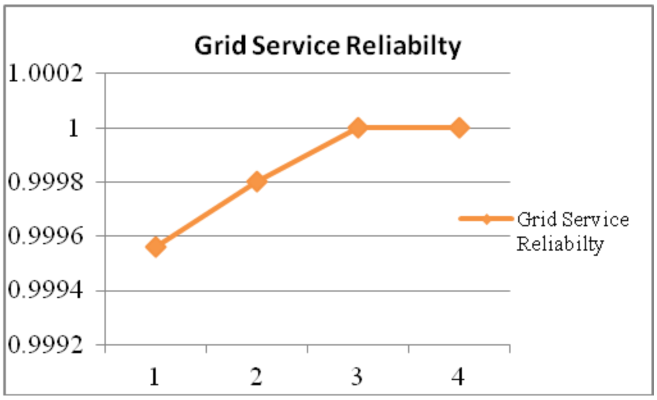

Fig 4 Grid service reliability with respect to the possible set of nodes

Fig 4 shows the grid service reliability range with a possible set of 4 nodes. The node failures are tolerated and it maintains stable grid service reliability in different situations considering failure recoverability.

TABLE 2. **Grid service reliability**

Nodes	Grid Service Reliability
1	0.999560920
2	0.999800106
3	0.999999999
4	1.000000000

Table 2 depicts the calculation of the reliability of the grid service with respect to a set of nodes ranging from 1 to 4.

V. CONCLUSION

In this paper, a fault recovery mechanism is introduced into the grid, and the modeling of grid service reliability considering fault recovery is presented. In order to make it more practical, a constraint on recovery amount is discussed in the modeling of grid service reliability. As for the implementation of fault recovery in grid resources, it can be achieved by embedding a fault recovery module in grid clients. In the module, there are options such as the allowed life times of grid subtasks, and the allowed numbers of recoveries performed. By those options, resource providers can be free to choose appropriate fault recovery strategies according to the local situations. Based on that, a task scheduling optimization model called Min Max Cost Reliability Scheduling Alg is proposed to maximize grid service reliability and minimize the total cost simultaneously is proposed, and this algorithm is used to solve this task scheduling problem. Maximizing the grid service reliability and minimizing the cost, as called to be soft constraints is attained in this approach.

ACKNOWLEDGMENT

I express sincere thanks to Prof. S. G. Shikalpure my project guide, who gave me his valuable and rich guidance and helped in presentation of this research paper.

REFERENCES

1. Tian-Liang Huang, Tian-An Hsieh, Kuan Chou Lai, Kuan-Ching Li, Ching-Hsien Hsu, and Hsi-Ya Chang." Fault Tolerance Policy on Dynamic Load Balancing in P2P Grids" 2011 International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11.
2. Foster, "The Grid: A new infrastructure for 21st century science," Physics Today, vol. 55, no. 2, pp. 42–47, 2002.
3. Y. S. Dai, M. Xie, and K. L. Poh, "Reliability of grid service systems," Computers and Industrial Engineering, vol. 50, no. 1–2, pp. 130–147, 2006.
4. S. C. Guo, H.Wan, G. B.Wang, and M. Xie, "Analysis of grid resource compensation in market-oriented environment," Eksploatacja I Niezawodnooeæ— Maintenance and Reliability, vol. 45, no. 2, pp. 36–42, 2010.
5. K. Krauter, R. Buyya, and M. Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing," Software— Practice and Experience, vol. 32, no. 2, pp. 135–164, 2002.
6. C. L. Li and L. Y. Li, "Multiple QoS modeling and algorithm in computational grid," Journal of Systems Engineering and Electronics, vol. 18, no. 2, pp. 412–417, 2007.
7. G. Levitin, Y. S. Dai, and B. H. Hanoch, "Reliability and performance of star topology grid service with precedence constraints on subtask execution," IEEE Trans. Reliability, vol. 55, no. 3, pp. 507–515, 2006.
8. G. Levitin and Y. S. Dai, "Grid service reliability and performance in grid system with star topology," Reliability Engineering and System Safety, vol. 92, no. 1, pp. 40–46, 2007.
9. Y. S. Dai, G. Levitin, and X. L. Wang, "Optimal task partition and distribution in grid service system with common cause failures," Future Generation Computer Systems, vol. 23, no. 2, pp. 209–218, 2007.
10. Y. S. Dai, Y. Pan, and X. K. Zou, "A hierarchical modeling and analysis for grid service reliability," IEEE Trans. Computers, vol. 56, no. 5, pp. 681–691, 2007.
11. T. Paul and X. Jie, "Fault tolerance within a grid environment," in Proceedings of UK e-Science All Hands Meeting, 2003.
12. M. Affaan and M. A. Ansari, "Distributed fault management for computational grids," in Proceedings of the Fifth International Conference on Grid and Cooperative Computing, 2006. [13] L. Jin, W. Q. Tong, J. Q. Tang, and B.Wang, "A fault-tolerance mechanism in grid," in Proceedings of IEEE International Conference on Industrial Informatics, 2003.
13. K. Jozsef and K. Peter, "A migration framework for executing parallel programs in the grid," in Proceedings of European across Grids Conference, 2004.
14. L. Xing and Y. S. Dai, "A new decision diagram model for efficient analysis on multi-state systems," IEEE Trans. Dependable and Secure Computing, vol. 6, no. 3, pp. 161–174, 2009.
15. Y. S. Dai and X. L. Wang, "Optimal resource allocation on grid systems for maximizing service reliability using a genetic algorithm," Reliability Engineering and System Safety, vol. 91, no. 9, pp. 1071–1082, 2006.
16. Y. S. Dai and G. Levitin, "Optimal resource allocation for maximizing performance and reliability in tree-structured grid services," IEEE Trans. Reliability, vol. 56, no. 3, pp. 444–453, 2007.
17. D. Coit and A. Smith, "Reliability optimization of series-parallel systems using genetic algorithm," IEEE Trans. Reliability, vol. 45, no. 2, pp. 254–260, 1996.
18. Y. C. Liang and A. E. Smith, "An ant colony optimization algorithm for the redundancy allocation problem (RAP)," IEEE Trans. Reliability, vol. 53, no. 3, pp. 417–423, 2004.
19. H. Z. Huang, J. Qu, and M. J. Zuo, "Genetic-algorithm-based optimal apportionment of reliability and redundancy under multiple objectives," IIE Transactions, vol. 41, no. 4, pp. 287–298, 2009.
20. G. D. Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," Journal of Artificial Intelligence Research, vol. 9, no. 2, pp. 317–365, 1998.
21. D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," IEEE Transactions on Evolutionary Computation, vol. 6, no. 4, pp. 333–346, 2002.