

# Pecking Order Continuous Rescue Fault Tolerance Scheduling In Grid Computing

Kala.K, Stephe Rachel.I, Balasubramanian. C

PG Scholar, Dept of Computer Science and Engineering P.S.R.Rengasamy College of Engineering,  
for Women, Sivakasi, India.

Assistant Professor, Dept of Computer Science and Engineering P.S.R.Rengasamy College of Engineering,  
for Women, Sivakasi, India.

Professor, Dept of Computer Science and Engineering P.S.R.Rengasamy College of Engineering,  
for Women, Sivakasi, India.

**Abstract** — Grid computing is an emerging technology in the next-generation of parallel and distributed computing methodology that aggregates diverse heterogeneous resources for solving various kinds of large-scale applications in science and engineering. Generally grid scheduling approaches will improve resources utilization in grids. In Existing System, the reliability of task for an application is measured effectively on Grid virtual nodes. The main work in grid area is to deal with the fault prediction and fault tolerance; it will be more complex due to the dynamic nature of the grid computing. The proposed mechanism improves the scheduling of tasks to various grid resources. Moreover, the hierarchical fault scheduling in computational grid is to deduct the resource failure in hierarchical manner. This hierarchical scheduling mechanism only occurs at the level of Global server and Local server; whereas fault tolerance is take place at the global scheduler and fault prediction occur at the local scheduler. Thus, lucratively improve the task reliability and availability of an application in grid by implementing the Fault Prediction and Fault Tolerance (FP and FT) Algorithms.

**Keywords** — Grid Computing, Grid Scheduling, Reliability, Fault Prediction, Fault Tolerance

## I. INTRODUCTION

A grid computing infrastructure provides access to high level computational capabilities in a reliable, pervasive, consistent and inexpensive manner. Geographically distributed resources cooperate with each other to solve big problems. It enables users to use its resources for large-scale computing applications in science, engineering and commerce [1]. Grid computing system is different from conventional distributed computing systems by its focus on large-scale resource sharing, where processors and communication have significant influence on grid computing reliability [3].

The sharing that the grid computing is concerned with is not primarily file exchange but rather direct access to systems, software data, and other resources, which is required by a range of collaborative problem-solving and resource-brokering strategies. Thus, the communication between computing programs and resources significantly affects the grid computing reliability.

Grid scheduling is a procedure responsible for transmission users' jobs onto available grid resource. The ambition of this process is to take full advantage of various optimization criteria such as machine handling, equality, flow time or to promise non minor QoS (Quality of Service). Preparation process should be flexible and fast so that it is able to powerfully respond on energetic changes in the Grid environment (failure, job arrival, imprecise job runtime estimate, etc.).

Prediction failure is the type of techniques. Upcoming occurrences are forecasts in data centers using runtime implementation states of the system and the information history of failures observed. It gives precious information for resource allocation, computation reconfiguration and system maintenance [1]. In distinction to classical consistency methods, failure prediction is based on runtime monitoring and a selection of models and current methods using by the state of a system and the past knowledge as well.

Fault tolerance is the potential of a system to perform its function correctly even in the presence of faults. Fault tolerance makes the system more dependable. Fault tolerance is also defined as preserving the delivery of expected services despite the presence of fault-caused errors within the system itself. Errors are detected and corrected but permanent faults are located and removed while the system continues to deliver acceptable services [3].

Grid system reliability is defined as the probability for all of the grid computing programs to be executed successfully in the grid computing system. In other words, the grid program reliability measures the reliability of one program executed in the grid. However, for the grid computing system, it is important to obtain a global reliability measure that describes how reliable the grid system is for a given distribution of programs and resources. One way of measuring the reliability of the grid computing system is by determining grid system reliability which is defined as the probability that all the computing programs are executed successfully.

II. RELATED WORKS

Fault Tolerance Min-Min algorithm has been proposed to calculate the failure rate, the fitness value and compare with the fault tolerant scheduling policy. The primary objective of this proposal is to reduce the makespan which is the total time taken to complete a set of jobs and to ensure the idle time of resources. The main advantage is FTMM algorithm using for proactive fault tolerance technique and static scheduling and it can be extended for dynamic scheduling [8].

Rough set analysis algorithm used to for resource scheduling in the grid system by using scheduler machine which uses case based reasoning technique (CBRT) and Rough set analysis algorithm is also to fix the occurrences of faults during the resource scheduling. The goal of this approach is to increase the fault tolerant confidence so that the performance of grid system can be enhanced to the highest levels. In existing grid-scheduling approach can select the best fault tolerance nodes and also detect a failure node and simply manage it by using one of the provided strategies such as multiversioning, Reservation Queue, Replacement and transferring job to the nearest neighbour. The obtained results by our simulation indicate that the new approach may be very effective for adaptive grid scheduling due to reliability, fault tolerance, and then decrease of job completion time. [9].

FT-Pro, is a mechanism of adaptive fault management system, which selects migration,

checkpointing or it taken to zero action to minimize the time of application execution. In the existence of failures based on the failure prediction and the cost based evaluation models are used to take the dynamic decisions at run time. The principle objective of this scheme is to minimize the unwanted checkpoints and to avoid failures so that the time completion of parallel applications can be minimised [10].

III. SYSTEM MODEL

The grid environment is created with global server, local servers, GIS(Grid Information server), Gridlets given by a user, number of resource the components of grid environment is connected by a network topology.

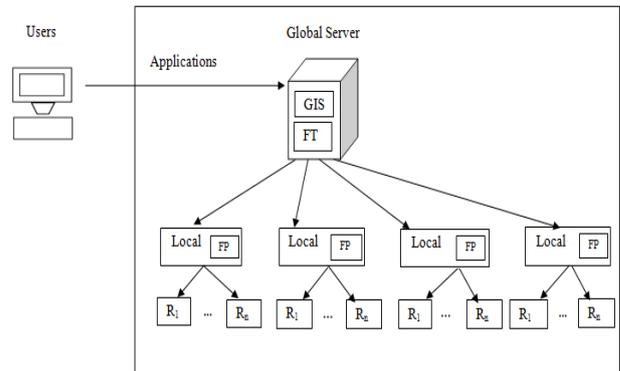


Fig.1 Grid System Environment

The dynamic numbers of users as well as the number of gridlets are created. After the creation gridlets are assigned to the user. User sends request to the global server to execute the task by submitting the gridlets. Global server is collecting information from all virtual nodes and terminal users. Global scheduler, which resides on global server resource management system to schedule an application into different virtual nodes. Each virtual node is connected to resources and also collecting information from resources. Local scheduler, which resides on local server resource management system to schedule tasks of an application with resources. The grid computing environment consists of coordinated resources which solves the problem in dynamic virtual organization. This is required by a range of collaborative problem and it in evitable contains different types of failures. The failure includes 1. Network Failure such as Packet Loss, Packet Corruption, A network disconnection and partition and Link failure. 2. Time Faults such as Early Faults and Late Faults (Performance failure). 3. Job Failure such as Resource Failure, Network Failure and Time Overhead. 4. Hardware and software fault such as CPU, memory in storage devices and other peripherals, Program Failure. These failures create the grid computing environment unreliable. So, Grid reliability characteristics are ignored an it can lead to reduced application performance, such as scheduling time and speedup, due to wasted operations particularly resubmission of job.

A. Global Server

Any number of independent jobs are split into several task and that are routed into specific resource to complete the job in reliable manner for that the system introduce the concept of dynamic fault tolerance mechanism to tolerate the fault during progress. Thus by improves the reliability of scheduling.

The global server collects the information from all virtual node and users. Global server divides an application into number of task. Global scheduler, which resides on global server resource management system to schedule applications into nodes.

1. Proposed Fault Tolerance Algorithm

The brief description of the proposed Fault Tolerance algorithm is presented.

```

If resource failure occur // System Failure
    Get the error message
    If minor problem // Eg: Modification in the data
        Reload the essential data
    End if
    If major problem // Eg: System crash
        Reload the entire back up
    End if
End
    
```

Algorithm 1: Fault Tolerance Algorithm

B. Local Server

Each virtual node, there is a local server, which is responsible for collecting information from all resources within that virtual node. Local scheduler, which resides a local server resource management system to schedule tasks of an application with resources. Here, the fault prediction technique used to predict the fault to improve the internal scheduling.

The Expected Finish Time ( $T_f$ ) is calculated by using the following equation (1) ie., the  $T_f$  is the summation of Starting time, waiting time and execution time.

$$T_f = T_s + T_w + T_e \quad \text{---- (1)}$$

Where,

- $T_f$  = Finishing Time
- $T_s$  = Starting Time
- $T_w$  = Waiting Time
- $T_e$  = Execution Time

Probability that the tasks failed to be executed previously in resource is calculate by

$$FR_{Ri} = 1 - \frac{T_s}{T_{total}} \times 100 \quad \text{---- (2)}$$

Where,

- $T_s$  – Number of tasks successes to be executed previously in resource i.
- $FR_{Ri}$  – Probability that the tasks failed to be executed previously in resource i.
- $T_{total}$  – Number of jobs submitted to the resource i.

1. Fault Prediction Algorithm

Failure prediction is to predict the occurrence of failures during runtime based on the current system state.

```

Start Backup_data()
    For each resources
        Maintaining the log history and backup_data
    End for
End
If resource speed is getting low or job execution time-delay occurs then
    Predicted as system failure using Eq. (1)
    Fault Rate (FR) for a job can be calculated by using Eq. (2)
    Find the alternate resource by calling Redirection of job ()
End if
Redirection of Job()
If free resources are available then
    Redirect the job to the ideal resource
Else if non-free resources and non-equal tasks assigned then
    Find the resources with low number of task and Redirect the job
Else if non-free resources and equal tasks assigned then
    If speed of the resource is high then
        Redirect the job
    Else if task_status is Quit or task_status is Finished then
        Redirect the job
    End
End
    
```

Algorithm 2: Fault prediction algorithm

Finally, the algorithm HFTS (hierarchical fault tolerance scheduling) achieves the reliability of the Grid System during scheduling.

IV. RESULTS

The system uses the fault prediction and fault tolerance techniques to improve the reliability and speedup in scheduling. Results as follows,

The graph depicts system reliability and speedup.

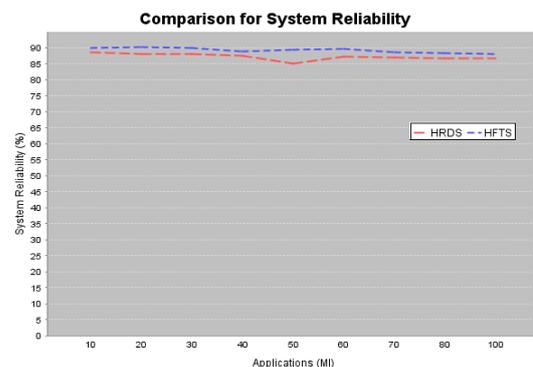


Fig. 2 Reliability Vs Applications

In Fig 2, the number of applications take at x axis with the unit of MI (Mille Instruction). The system reliability takes at y axis. The traditional HRDS algorithm achieves 85% of reliability while 50 applications are processed. The proposed HFTS algorithm achieves 89% of reliability for the same number of applications.

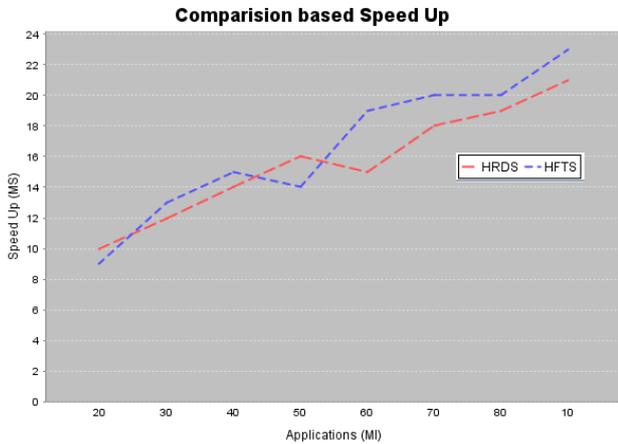


Fig. 3 Speedup Vs Applications

In Fig 3, the number of applications take at x axis with the unit of MI. The system speedup takes at y axis with the unit of MS (Millisecond). The proposed schemes increase the processing with respect to increased number of applications.

### V. CONCLUSION

In this work, it is mandatory to design and implement fault tolerance scheduling to meet applications' reliability requirements while achieving good performance. Due to the failure of application, and the absence of fault tolerance scheduler, the following problems may occur first, at the lower reliability level, the higher reliability applications will run and vice versa which produce the poor performance. The fault tolerant scheduler, with the absence of reliability overhead will lead to the problem during decision making of scheduling. To solve this problem, we build a hierarchical scheduling architecture that can effectively measure both the global application and local task reliability. Then, we propose a FPA that can predict the failure in small scale VN. Finally HFTS (hierarchical fault tolerance scheduling) is proposed and incorporates the applications reliability overhead into scheduling to achieve high system performance and improve system reliability and reduce the speedup.

In the future, this work will improve the reliability of grid system environment by using data integrity and data security, so that the system can able to predict the submitted job is error or not. The security is providing by encryption techniques to secure the jobs

which then fault tolerant mechanism used to predict the job failures.

### REFERENCES

- [1] A. J. Oliner, R. K. Sahoo, J. E. Moreira, M. Gupta, and A. Sivasubramaniam, "Fault-aware job scheduling for BlueGene/L systems," In Proceedings of IEEE/ACM International Parallel and Distributed Processing Symposium (IPDPS), 2004.
- [2] Daniel D'iaz, Xo'an C. Pardo, Mar'ia J. Mart'in, Patricia Gonz'alez have developed "Application-Level Fault-Tolerance Solutions for Grid Computing", International Journal of Engineering, Month 2010.
- [3] Y.S.Dai, M.Xie, K.L.Poh,"Reliability Analysis of Grid Computing Systems", 1991 vol(13).
- [4] Felix Salfner, Maren Lenk, and Mirosław Malek, "A Survey of Online Failure Prediction Methods", ACM Journal Name, Vol. V, No. N, Month 2009.
- [5] Gokuldev S, Valarmathi M, " Fault Tolerant System for Computational and Service Grid", International Journal of Engineering and Innovative Technology (IJEIT) Volume 2, Issue 10, April 2013.
- [6] Gopi Kandaswamy, Anirban Mandal, and Daniel A. Reed, "Fault Tolerance and Recovery of Scientific Workflows on Computational Grids". International Conference on Grid Computing, Dec. 5-8, 2012.
- [7] HwaMin Leea, KwangSik Chungb, SungHo China, JongHyuk Leea, DaeWon Leea,Seongbin Parka, HeonChang Yua, "A resource management and fault tolerance services in grid computing", J. Parallel Distrib. Comput. 65 (2005) 1305 – 1317.
- [8] Jasma Balasangameshwara and Nedunchezian Raju, "Performance-Driven Load Balancing with a Primary-Backup Approach for Computational Grids with Low Communication Cost and Replication Cost", Ieee Transactions On Computers, Vol. 62, No. 5, May 2013.
- [9] Laiping Zhao, Yizhi Ren z, Yang Xiangy, and Kouichi Sakurai, "Fault-Tolerant Scheduling with Dynamic Number of Replicas in Heterogeneous Systems", IEEE International Conference on High Performance Computing and Communications, 2010.
- [10] Lee I, R. K. Iyer and D. Tang, Error/Failure Analysis Using Event Logs From Fault Tolerant Systems, Proceedings 21st Intl. Symposium On Fault-Tolerant Computing, 1991,Pp. 10-17.
- [11] Muthumani N,"A Survey On Failure Prediction Methods", International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 2 Feb 2011.
- [12] Mrs.G.Malathi and Mrs. S.Sarumathi, "Survey On Grid Scheduling", Journal of Computer Applications, Vol-III, No.3, July - Sept 2010.
- [13] Penka Martincová, Michal Záborský, "Comparison of Simulated GRID Scheduling Algorithms", 4/2007.
- [14] P. Keerthika and N. Kasthuri, "An Efficient Fault Tolerant Scheduling Approach For Computational Grid", American Journal of Applied Sciences, 2012, 9 (12), 2046-2051.
- [15] P. Latchoumy and P. Sheik Abdul Khader, "Survey On Fault Tolerance In Grid Computing", International Journal of Computer Science & Engineering Survey (IJCSES) Vol.2, No.4, November 2011.
- [16] Ren, S. Lee, R. Eigenmann and S. Bagchi, Resource Failure Prediction In Fine-Grained Cycle Sharing System, IEEE HPDC,Paris,France, 2006.
- [17] Turnbull, D. & Alldrin, N. "Failure Prediction in Hardware Systems. Technical Report", University Of California, San Diego, 2003.
- [18] Xiaoyong Tanga, Kenli Li, Meikang Qiu, Edwin H.M. Sha, "A hierarchical reliability-driven scheduling algorithm in grid systems", J. Parallel Distrib. Comput. 72 (2012) 525–535.