How to Cite:
Pradeep Kumar, P., & Blessie, E. C. (2022). Reinforced cost effective architecture for fault tolerance mechanism through meta heuristics. *International Journal of Health Sciences*, *6*(S2), 2500–2506. https://doi.org/10.53730/ijhs.v6nS2.5568

Reinforced cost effective architecture for fault tolerance mechanism through meta heuristics

Pradeep Kumar P
Research Scholar, Department of Computer Applications, Nehru College of Management
Email: pradeepmanju4545@gmail.com

E. Chandra Blessie
Assistant Professor, Department of Computing(AIML), Coimbatore Institute of Technology
Email: chandra_blessie@yahoo.co.in

***Abstract*---**Grid computing removes the limitations that exist in traditional shared computing environment, and becomes a leading trend in distributed computing system. It aggregate heterogeneous resources distributed across Internet, regardless of differences between resources such as platform, hardware, software, architecture, language, and geographical location. Such resources, which include computing, storage, data, communication bandwidth resources and other resources, are combined dynamically to form high performance computing capability of solving problems in large-scale applications. Dynamically sharing resources gives rise to resource contention. One of the challenging problems is deciding the destination nodes where the tasks of grid application are to be executed. From the perspective of system architecture, resource allocation and job scheduling are the most crucial functions of grid computing. Resource allocation and job scheduling are the core functions of grid computing. These functions are based on adequate information of available resources. Timely acquiring resource status information is of great importance in ensuring overall performance of grid computing. This work aims at building a distributed system for grid resource monitoring and prediction. Major achievements include the design and evaluation of system architecture for grid resource monitoring and prediction through Meta heuristic conditions. We discuss the key issues for system implementation, including machine learning based methodologies for modelling and optimization of resource prediction models in terms of handling fault tolerance. Evaluations are performed on a prototype system. Our experimental results indicate

International Journal of Health Sciences ISSN 2550-6978 E-ISSN 2550-696X © 2022.
***Corresponding author**: Pradeep Kumar, P.; Email: pradeepmanju4545@gmail.com*
Manuscript submitted: 27 Jan 2022, Manuscript revised: 18 Feb 2022, Accepted for publication: 09 March 2022

that the efficiency and accuracy of our system meet the demand of online system for grid resource monitoring and prediction.

***Keywords*---**resource allocation, fault tolerance, meta heuristics, task scheduling, machine learning.

Introduction

A computer network, often simply referred to as a network, is a collection of computers and devices interconnected by communications channels that facilitate communications among users and allows users to share resources. Grids are a form of distributed computing whereby a "super virtual computer" is composed of many networked loosely coupled computers acting together to perform very large tasks. "distributed" or "grid" computing, in general, is a special type of parallel computing that relies on complete computers (with onboard CPUs, storage, power supplies, network interfaces, etc.) connected to a network (private, public or the Internet) by a conventional network interface, such as Ethernet. Grid computing environment is a distributed, shared environment implemented via the deployment of a persistent, service infrastructure that supports the creation and resource sharing within Distributed communities. In addition to timeliness requirements, quality of service (QoS) requirements must be addressed in various hard and soft real-time systems. In order to improve the stability of the system, each task selects different QoS levels by varying its period or execution time. Noticeably, the aforementioned QoS-aware real-time systems must then incorporate inherent high reliability features. Growing evidence shows that scheduling is an efficient approach to achieving high performance of applications in parallel systems like clusters. A wide variety of scheduling algorithms have been developed to provide fault tolerance for clusters supporting real-time applications. The paper is organized into section as follows, section 2 surveys the related work, Section 3 describes the proposed system, and Section 4 explains the experimental results and finally section 5 conclude the paper.

Related Work

In this section, various literatures related to the workload and resource scheduling has been analysed against various aspects as follows. To support creation of nimble applications for computational grids, J.S. Vetter and D.A. Reed (2000) **'**Real-Time Performance Monitoring, Adaptive Control, and Interactive Steering of Computational Grids' [6] believe one must eliminate the barrier that separates program creation from execution and post-mortem optimization.

P.A. Dinda (2006) has Designed and implementation, and performance of an extensible toolkit for resource prediction in distributed systems'[7].The systems predict the future values of such streams from past values and are composed at runtime out of a large and extensible set of communicating components that are in turn constructed using RPS's extensible sensor, prediction, wavelet, and communication libraries. T. Xie and X. Qin (2006)' Scheduling Security-Critical Real-Time Applications on Clusters '[8] proposed security overhead model that can be used to reasonably measure security overheads incurred by the security-

critical tasks . F. Harada, T. Ushio, and Y. Nakamoto (2007) 'Adaptive Resource Allocation Control for Fair QoS Management'[9] , resource utilization is assigned to each task through an online search for the fair QoS level based on the errors between the current QoS levels and their average.

Proposed -Scheduling Model

In this section, we will introduce the models, notions, and terminology used. Also we present the architecture of our grid resource monitoring and prediction system Grid resource state monitoring cares about the running state, distribution, load and malfunction of resources in grid system by means of monitoring strategies. Grid resource state prediction focuses on the variation trend and running track of resources in grid system by means of modelling and analyzing historical monitoring data.

Information service is deployed on information node; it interacts with grid users and runs the storage, query, as well as publication of resource state information. Two types of mechanism are defined for information acquisition: local register and group register. Local register timely collects information, from resource sensors to monitoring service, and from prediction model to prediction service. While group register timely collects information from both services and aggregates them for storage or publication. In order to provide a friendly interface to grid users, a web server is set on information node for customizing request and publishing information.

Scheduling Principles

Our system follows the following two scheduling principles:

- Given the same execution time, the algorithm should schedule tasks on nodes with low failure rates.
- For a group of nodes with the same failure rate, the algorithm needs to assign tasks in order to the nodes providing short execution times.

Meta Heuristic for Task Management against the Resource Fault

Metaheuristics may make few assumptions about the optimization problem being solved, and so they may be usable for variety of the conditions in resource and task scheduling in the grid computing [8][9]. Meta heuristics works on stochastic optimization Method to generalize the deterministic case for deterministic heterogeneous resource types in the grid. Deterministic case of the resource variants depends upon the approximation methods.

Algorithm for the Meta Heuristic Optimization of Resource

Create a Heuristic model to be connected by the job
Bind the Resource to an empty Communication using optimization technique $\sum\mu_i+\sum\mu_r$
Where I = task ,R = resource
Listen to this port utilizing $\sum\mu_i$

BEGIN
While condition = TRUE do
If a new connection request arrives **then**
The optimization server accepts the connection from the Work load $\sum \mu_r$
Process id is computed as $K_{p=1}=1+\alpha\ (4+\beta)$
If process Id = Meta information of Work ID
then
Back to listening to the designated port
Else {in the child process}
Child: close the listening port
Child: receive the request information from the Task
Child: perform sampling transfers $\alpha\ (4+\beta)$
Child: build a mathematical model and process the sampling results α
Child: send back the optimized parameters to the Workload β
Child: close the connection
Child: terminate
End if
Else {no new connection request comes}
END

Experimental Analysis

Grid resource is described in terms of their various characteristics, such as resource ID, name, total number machines in each resource, total processing elements (PE) in each machine, MIPS of each PE, and bandwidth speed. Once GridSim starts, the resource entities register themselves with the Grid Information Service (GIS) entity. The broker entity queries GIS entity for resource discovery, based on the user entity's request. The GIS entity returns a list of registered resources, and their contact details. Group the jobs based on the total MI of the resource This group is associated with a resource ID. Submit the job groups to their corresponding resources for job computation using a dispatcher. Get the results and record statistics. Grid sim tool architecture is described in the figure 1

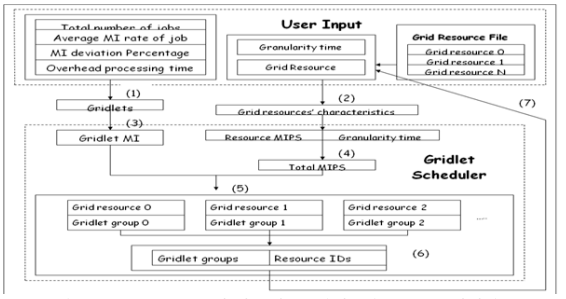

Figure 1: Model of Grid sim Tool kit

The critical path of a workflow is the longest execution path between the entry and the exit tasks of the workflow. Most of these heuristics try to schedule critical tasks (nodes), i.e., the tasks belonging to the critical path, first by assigning them to the services that process them earliest, in order to minimize the execution time of the entire workflow [15]. Deadline distribution method adopted in the proposed

work is based on a similar heuristic, but it uses the critical path to distribute the overall deadline of the workflow across the critical nodes. After this distribution, each critical node has a sub deadline which can be used to compute a sub deadline for all of its parent nodes, i.e., its (direct) predecessors in the workflow.

Performance Comparison

This section shows the performance impact of Throughput and Response time against the existing technique for resource scheduling and fault tolerance. Execution Time and Throughput has taken as performance metrics to evaluate the grid Computing performance through the monitoring and prediction results through Meta heuristics strategies.

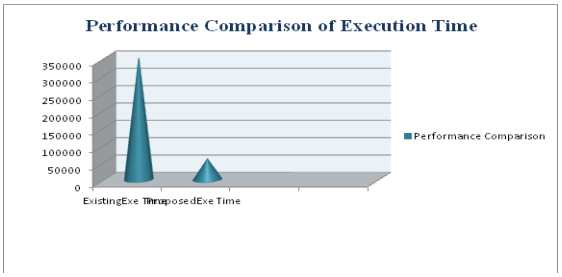

Figure 2 Execution Time Taken by the Systems (resources) for Same Task

It can be inferred that the proposed monitoring subsystem does not bring obvious influence on computing nodes. Hence, the design of monitoring subsystem is acceptable.

Table 3 Tabulation of Statistical Results of Overhead

Statistics	**Prediction Usage Rate of CPU and Memory (%)**			
	Monitoring A	Monitoring B	Monitoring C	Monitoring D
Minimum	13	40	50	83
Maximum	15	89	56	53
Average	14	64	53	68

The algorithm to instantiate the throughput function with respect to the number of parallel streams can avoid the ineffectiveness of the prediction models due to some unexpected sampling data pairs. In the research work, this new service has been implemented in the Resource Scheduler, where the prediction points can be obtained using Meta heuristics information. Figure 2 Provides the performance of the existing and proposed model.

Conclusion and Discussion

We have implemented a distributed Reinforced Cost Effective Architecture for Fault Tolerance Mechanism utilizing the Meta heuristic conditions including the fault tolerance properties that seamlessly combines grid technologies, resource

monitoring and machine learning based resource state prediction. To sufficiently utilize the system resource based on the storage, communication bandwidth to reach a high performance computing in the heterogeneous resource clusters. Dynamic sharing of the resource configuration has produced the resource contention in the grid computing, it has been resolved through job scheduling through Timely acquiring resource status information. This system consists of a set of distributed services to accomplish all required resource monitoring, data gathering, and resource state prediction functions. Hence the fault tolerance of the resource in job scheduling has carried out with static multidimensional condition which updates the global manager of the grid resource.

References

1. F. Berman, R. Wolski, H. Casanova, W. Cirne, H. Dail, M. Faerman, S. Figueira, J. Hayes, G. Obertelli, J. Schopf, G. Shao, S. Smallen, N. Spring, A. Su, and D. Zagorodnov, "Adaptive computing on the grid using AppLeS," IEEE Trans. Parallel Distrib. Syst., vol. 14, no. 4, pp. 369–382, Apr. 2003.
2. V. V. Krzhizhanovskaya and V. V. Korkhov, "Dynamic load balancing of black-box applications with a resource selection mechanism on heterogeneous resources of the grid," in Conf. Parallel Comput. Technol. (PaCT), LNCS, Berlin/Heidelberg, 2007, vol. 4671, pp. 245–260.
3. H.Y. Chang, K.C. Huang, C.Y. Shen, S.C. Tcheng, and C.Y. Chou, "Parallel Computation of a Weather Model in a Cluster Environment," J. Computer-Aided Civil and Infrastructure Eng., vol. 16, no. 5, pp. 365-373, Sept. 2001.
4. T. Xie and X. Qin, "Scheduling Security-Critical Real-Time Applications on Clusters," IEEE Trans. Computers, vol. 55, no. 7, pp. 864-879, July 2006.
5. C.M. Krishna and K.G. Shin, Real-Time Systems. McGraw-Hill, 2001.
6. T.F. Atdelzater, E.M. Atkins, and K.G. Shin, "QoS Negotiation in Real-Time Systems and Its Application to Automated Flight Control," IEEE Trans. Computers, vol. 49, no. 11, pp. 1170-1183, Nov. 2000.
7. J.S. Vetter and D.A. Reed, "Real-Time Performance Monitoring, Adaptive Control, and Interactive Steering of Computational Grids," Int'l J. High Performance Computing Applications, vol. 14, no. 4, pp. 357-366, 2000.
8. D.M. Swany and R. Wolski, "Multivariate Resource Performance Forecasting in the Network Weather Service," Proc. ACM/IEEE Conf. Supercomputing, pp. 1-10, Nov. 2002.
9. P.A. Dinda and D.R. O'Hallaron, "Host Load Prediction Using Linear Models," Cluster Computing, vol. 3, no. 4, pp. 265-280, 2000.
10. E. Caron, A. Chis, F. Desprez, and A. Su, "Design of Plug-in Schedulers for a GRIDRPC Environment," Future Generation Computer Systems, vol. 24, no. 1, pp. 46-57, 2008.
11. P.A. Dinda, "Design, Implementation, and Performance of an Extensible Toolkit for Resource Prediction in Distributed Systems," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 2, pp. 160-173, Feb. 2006.
12. T. Xie and X. Qin, "Scheduling Security-Critical Real-Time Applications on Clusters," IEEE Trans. Computers, vol. 55, no. 7, pp. 864-879, July 2006.
13. F. Harada, T. Ushio, and Y. Nakamoto, "Adaptive Resource Allocation Control for Fair QoS Management," IEEE Trans. Computers, vol. 56, no. 3, pp. 344-357, Mar. 2007.

14. S. Ghosh, R. Melhem, and D. Mosse´, “Fault-Tolerance Through Scheduling of Aperiodic Tasks in Hard Real-Time Multiprocessor Systems,” IEEE Trans. Parallel and Distributed Systems, vol. 8, no. 3, pp. 272-284, Mar. 1997.
15. R.U. Payli, E. Yilmaz, A. Ecer, H.U. Akay and S. Chien. ‘DLB: A Dynamic Load Balancing Tool for Grid Computing’, Parallel CFD Conference, May 24-27, 2004, Grand Canaria, Canary Islands, Spain.