FAULT TOLERANCE IN MOBILE GRID COMPUTING

Aghila Rajagopal
KLN College of Information Technology
Pottapalayam-630611, Sivagangai Dt., Tamilnadu, India
ssg_akila@mamce.org

M.A. Maluk Mohamed
MAM College of Engineering
Tiruchy -chennai trunk Road, Siruganur, tiruchirappalli- 621 105,
Tamilnadu, India
ssg_maluk@mamce.org

ABSTRACT

This paper proposes a novel model for Surrogate Object based paradigm in mobile grid environment for achieving a Fault Tolerance. Basically Mobile Grid Computing Model focuses on Service Composition and Resource Sharing Process. In order to increase the performance of the system, Fault Recovery plays a vital role. In our Proposed System for Recovery point, Surrogate Object Based Checkpoint Recovery Model is introduced. This Checkpoint Recovery model depends on the Surrogate Object and the Fault Recovery is handled by means of Replication Approach. The mobility problem is overcome based on Distributed Surrogate Object model. In our comparison, our proposed fault recovery model proves its efficiency and reliability in Grid Environment.

Keywords: Object Based Mobile Grid, Surrogate Object, Fault Recovery

1. INTRODUCTION

Distributed System makes effective use of the idle computation, which are available. Many problems are faced when the computing power of the nodes are harnessed such as its heterogeneity nature, processing power and operating system. Research has been carried out to tackle these issues.

Models like Now as mentioned by Anderson, Culler and Patterson¹ is connected using wired connection, and they produced an improved performance in parallel programming. These models are formed only for homogenous devices. Later Condor as mentioned by Litzkow and Solomon², Piranha by Gelernter and Kaminsky³, ADM in Casas, Konuru, Otto, Prouty, and Walpole⁴, ARC by Joshi, and Janaki Ram.D⁵, Sprite by Douglis.R and Ousterhout.J⁶, and V System by Cheriton.D.R⁷ had solved the problem of heterogeneity. The inter task communication is not considered. As the task is divided into subtasks for computation, the subtask might require the result of other subtask for its computation. It means that the inter task communication is required in distributed system. Distributed Pipe by Binu K .Johnson, R. Karthikeyan, D.Janaki Ram⁸, a Transparent Programming Model was explored.

This paper proposes a novel approach of Surrogate Object Based Paradigm on considering the inter task communication among the process and also provides Fault Recovery Mechanism through Checkpoint Replication method. The rest of the paper is organized as follows. Section 2 explains the overview of the proposed Model, that includes the architecture of the proposed object based mobile grid. Section 3 comes out with the implementation explanation. Performance Analysis about the system is explained in Section 4. Section 5 finishes with the conclusion.

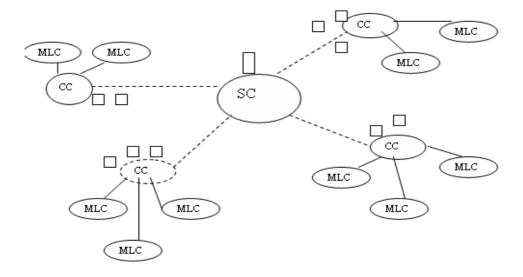


Figure 1. Structure of object based mobile grid

2. PROPOSED MODEL

Coordination process is the process which initiates the computation. Subordinate process is the process which participates in parallel grid computation. Coordination processes are called as Mobile Cluster Grid Module (MCGM).

2.1 Initialization

Every process needs to register with the system in order to avail the system services. The service can be availed by initializing and closing procedure.

2.2 Subtask model

On receiving the grid information, the system decides the number of MCGM to be involved, the computation allotted to each MCGM.

2.3 Information Gathering

When the mobile node migrates, MCGM initializes its data structures to hold its initial data, collects the data and begin the computation.

2.4 Communication Model

Each MCGM has to communicate with its neighbour MCGM to exchange boundary values. MCGM collects the information about its neighbours and the number of distributed pipes to be opened.

2.5 Checkpoint Model

Checkpoint saves the process state in a stable storage periodically during the execution. Surrogate Object acts as a cache for storage of Checkpoint.

3. IMPLEMENTATION

3.1 System Coordinator

SC maintains the individuals CC and facilitates the communication between them. SC manages all the nodes, coordinates all the functions, distribution of tasks and it also maintains the cache of task migration history. SC maintains a Global Surrogate Object GSO. GSO maintains a replica of information about the Surrogate Object. When CC identifies that SC has failed, then the next SC is identified using an Election Algorithm by Garcia-Molina⁹.

3.2 Co-coordinator

System has multiple CC's. Each CC has a group of nodes. CC also maintains the history of execution which takes place within the cell region. When a mobile node moves from one cell region to another region, the Surrogate Object² takes care of this.

3.3 Surrogate Object

Surrogate objects SO act as a place holder for the mobile node. It maintains the information about the mobile node and plays an active role during its absence.

3.4 Mobile Local Coordinator

Every Mobile Host MH is coordinated by the Mobile Local Coordinator MLC. It coordinates the local activities of the Mobile Host. All the activities to and from the MH are taken care by the MLC.

3.5 Cell Region

The Mobile Host which are within the Cell region are grouped based on the memory capability of the node. As when the nodes enter into the system they register based on their capability.

3.6 Timeouts

The nodes take an infinite amount of time in executing the subtask. Hence a Time out mechanism is needed. Timeout is maintained by the SC and CC.

4. PERFORMANCE ANALYSIS

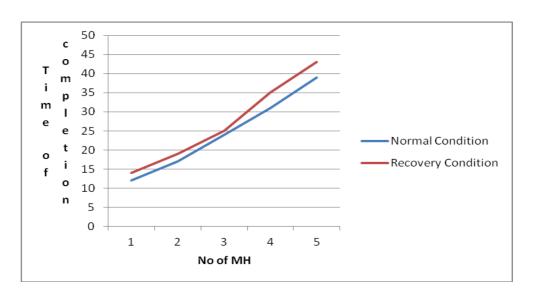


Figure 2. Time of Completion

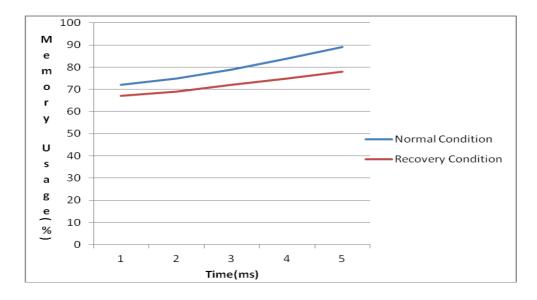


Figure 3. Time vs memory usage

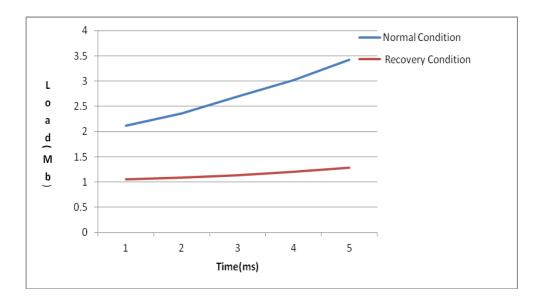


Figure 4. Time vs load

In figure 2 we have shown the efficiency of our Repfinder process in both the time and MH coordinates, where the both the coordinates are directly proportional showing steady improvement. It is more effective than the normal conditions which are indicated in green and red lines for recovery condition and normal condition respectively. Then resource utilization is optimally maintained with recovery condition with more than twenty percent than the previous condition and is shown in figure 3. Figure 4 shows a drastic depletion of load problem and our recovery condition proved steady and a clear stream of load balance than the normal initially with one percent and at last more than three percent load balance is optimized through Repfinder.

5. CONCLUSION

This paper incurs an efficient system for Fault recovery in parallel processing of Mobile Grid Computing through Repfinder by exact retrieval of data from the subsystem. Repfinder is meant for data availability, which creates additional replicas, and the replicas are taken into the account when the failures are detected. Repfinder provides efficient data flow with minimum transmission failure. It selects replication level for failure and initiates retransmission from that level, where it is blocked. It deploys an efficient storage system by combining retransmission with structured hierarchical overlays. For this issue the system automatically directs the system (CC and MLC) to the level in which the stored file is accessed through the subsystem. Once the system fails, retransmission process will be initiated

with negative response from SC, a request will be sent. If the data availability matches with the required data then the system will be closed, else the cycle will be made until proper transmission is achieved. Thus the Repfinder provides an efficient zero error mobile grid computing environment and the scalability of the system is good.

6. REFERENCES

- [1] T.E. Anderson, D.E. Culler, and D.A. Patterson, A case for NOW (networks of workstations). Micro, IEEE, 15(1), p54-64, 1995. http://dx.doi.org/10.1109/40.342018.
- [2] M. Mohamed, An object Based Paradigm for Integration of Mobile Hosts into Grid. International Journal of Next- Generation Computing, 2(1), p1-23, 2011.
- [3] D. Gelernter, and D. Kaminsky, Supercomputing Out of Recycled Garbage: Preliminary Experience with Piranha. Presented at the 6th international conference on Supercomputing, Washington, DC, USA, July 19 - 24, 1992. http://dx.doi.org/10.1145/143369.143444.
- [4] J. Casas, R. Konuru, S. Otto, R. Prouty, and J. Walpole, Adaptive Load Migration Systems for PVM. Presented at the 1994 ACM/IEEE conference onSupercomputing, November 14-18, 1994. http://dx.doi.org/10.1109/SUPERC.1994.344302.
- [5] R.K. Joshi, and D.J. Ram, Anonymous Remote Computing: A Paradigm for Parallel Programming on Interconnected Workstations. IEEE Transactions on Software Engineering, 25(1), p75-90, http://dx.doi.org/10.1109/32.748919.
- [6] R. Douglis, and J. Ousterhout, Transparent Process Migration: Design Alternatives and the Sprite Implementation. Software: Practice and p757-785, 1991. Experience, 21(8), http://dx.doi.org/10.1002/spe.4380210802.
- [7] D. Cheriton, The V Distributed System. Communications of the ACM,31(3), p314-333, 1988. http://dx.doi.org/10.1145/42392.42400.
- [8] F. Tandiary, S.C. Kothari, A. Dixit, E.W. Anderson, Batrun: utilizing idle workstations for large scale computing. IEEE Parallel & Distributed Technology: Systems & Applications, 4(2), p41-49, 1996. http://dx.doi.org/10.1109/88.494603.
- [9] H. Garcia-Molina, Elections in a distributed computing system. IEEE p48-59, 1982. **Transactions** on Computers, 31(1), http://dx.doi.org/10.1109/TC.1982.1675885.