Volume 3, Issue 3, March 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Flexible Rollback Recovery in Grid Using Time Based Checkpoint

Praveenraj.D*
Department of CSE
Anna University, India

Rajarajan.K
Department of CSE
Anna University, India

Poornima.S
Department of CSE
Anna University, India

Baalaaji.S
Department of MBA
Pondicherry University, India

***Abstract*— When a process fails in a node present in a grid network, it is recovered by rolling back to the previous state making use of checkpoints. To avoid unnecessary large rollbacks, the checkpoint frequency can be increased. Increase in checkpoint frequency however creates a memory overhead in the network. This paper proposes a new checking pointing approach called time based check points along with communication induced checkpoints. Time based checkpoints considerably reduce the wastage of memory and communication induced checkpoints avoid Domino's effect in the network caused by inter node communication. Therefore, these two approaches would provide a fault tolerance system and increase the efficiency of the grid environment.**

Keywords— Check point, Grid Computing, Rollback Recovery

I. INTRODUCTION

Distributed systems consist of multiple processor nodes connected through a computer network. Cluster computing and grid computing are the most suitable ways for establishing a distributed system. Various resources from various geographic locations form a grid environment. The tasks in grid computing are scheduled by two ways; static and dynamic scheduling [10]. If the processes are scheduled to the resources before execution, it is called static grid computing. In dynamic grid computing the processes are scheduled dynamically at runtime to the heterogeneous resources.

Large applications are executed in a dynamic grid environment with a number of nodes. A process may be divided into several sub processes and be executed by hundreds of nodes. If a node crashes, few other nodes would also get affected, as these nodes may include dependent sub processes. In such a way, a single node crash may lead to tremendous wastage of process time, computation power and memory. Conventionally, to minimize these issues, a rollback recovery by checkpoints is used.

Upon a failure, checkpoint based rollback recovery restores the system state to the state of most recent consistent set of checkpoint [4] i.e. recovery line. Checkpoints are used to periodically store the process state of the nodes and are used at the time of system crash. Checkpoint frequency is the number of checkpoints per particular period. If the check point frequency is small and if the system crashes when a process is about to end, a larger rollback is necessary to recover to the previous state. This leads to wastage of computational power and time.

This issue can be solved by increasing the number of checkpoints in the grid environment [2]. In this case, when a system crashes, a small roll back is sufficient to get back to the previous state. However, if the number of check points is increased, and if check points are available for smaller intervals, there is tremendous wastage of memory. Besides, if there are no crashes found on the nodes, these checkpoints create a memory overhead in the network.

Generally, the nodes in grid computing are capable of performing high and complex computations. Therefore, the computations required in storing the checkpoints are simple and practically negligible. However, memory is an issue in storing large number of check points. When a rollback is made to recover a crashed system, it may cause rollbacks in other nodes with dependent sub processes. Thus this creates a chain reaction and is called Domino effect.

This paper proposes an idea to use a new check pointing scheme called time based check pointing. These checkpoints would considerably decrease the wastage of memory by recycling the memory used in saving the checkpoints. Here, each check point has a lifetime and gets destroyed once a set of new consistent check points are created. Along with these checkpoints, another type of checkpoints is created called communication induced checkpoints. These checkpoints are created only when a node communicates with another, in order to decrease the Domino's effect caused by successive rollbacks.

II. RELATED WORKS

Checkpoints are generally used in multiprocessing environments to provide fault tolerance. Conventionally, checkpoints store the current process state. Increase in checkpoint frequency causes memory overhead in the network.

Ge-Ming Chiuet al. [1] suggested that diskless check pointing can be used to provide fault tolerance in distributed or parallel computing systems. In this technique, a processor saves a checkpoint in a peer processor called checkpoint storage node using XOR operation .Parity technique is used to reduce the size of memory space required for storing checkpoint data. Gupta B et al. [2], suggest that the effect of domino phenomena can be limited by using forced checkpoints. Apart from providing a single phase non blocking algorithm, the algorithm takes only one consistent checkpoint per process. Hui Liu et al. [3] points out that increase in checkpoint frequency increases memory overhead and hence they introduce a fault tolerance algorithm without using checkpoints. Amir Jafar et al. [4] overcomes the problem of node failures and provides a dynamic configuration over extensive run-time by presenting two fault-tolerance mechanisms called Theft Induced Check pointing and Systematic Event Logging. Sanjay Bansal et al. [5] provide an overview of the critical issues involved in fast and efficient recovery based on check pointing and also explain the types of check pointing and corresponding recovery issues.

III. INTRODUCTION TO PROPOSED SYSTEM

A. *System Model*

Let us consider a grid environment with n number of processors P_1,P_2,P_3…Pn that are connected in a network and each processor may be located in various geographical area. The most prominent types of Grid are [11].Compute Grid- multiple processors processing a single process. Data Grid- multiple storage systems hosting one very large data set. Utility Grid is the systems from multiple organizations collaborating for a common issue. This paper focuses on providing fault tolerance in a 'Compute Grid'. Every processor has its own physical memory and communication channel with all other processors in the grid.

When a task is given, the task is partitioned in to n subtasks and each subtask is executed by distinct processor Pi, $0 \leq i \leq n$. These subtasks communicate with each other by message passing in the network. Each processor takes a checkpoint based on a certain criteria (time, message passing, etc.). When a failure occurs the process can be restarted from the earlier checkpoint that is stored in the memory.

The following section provides an overview of the proposed scheme and describes its specific goals. The next section shows how these goals can be fulfilled.

B. *Basic Operation of Proposed Scheme*

When checkpoints are created in a grid, there is a possibility of checkpoint redundancy. Checkpoint redundancy can be defined as a state where in a number of checkpoints save the same process state. To resolve this problem a new idea called Time Based Checkpoint is proposed that also considers message passing between the processors. Therefore, along with Time Based Checkpoints, this paper also includes communication induced checkpoints.

1) Domino Effect: In distributed systems when a process is roll backed to its previous state upon a failure, there is a chance of rollback other dependent processors. This may result in a series of rollbacks in the environment and is referred to as Domino's effect.

IV. TIME BASED CHECK POINT

Check points that are created and destroyed periodically are referred to as time based check points. Here in, when a new check point is created and found to be consistent, the old check points are destroyed. Once the check point is destroyed, the memory that it occupied is released. Therefore, in a system that makes use of time based check points, the number of checkpoints and the memory involved in storing these check points remain constant.

Let as consider a task T is subdivided into t_1, t_2…t_m and scheduled into m number of nodes .processor process the task while processing processor takes checkpoint C_i periodically while the process take the next checkpoint C_{i+1}then Ci get deleted it occurs recursively. Small amount of process is taken but it saves large memory.

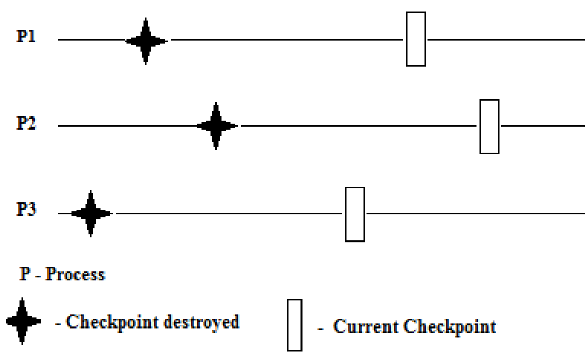

Fig. 1 Time Based Checkpoints

When a message is passed from processor P_1 to processor P_2 to process a task and if the process fails in processor P2 it has to be roll backed to latest consistent checkpoint. After rollback, the processor P_2waits for a message from P_1 but P_1 has a record of message being sent to P_2. The message that has been sent initially by the processor is left in the communication channel and is called an Orphan message. Hence, time based checkpoints would not be sufficient for a successful rollback mechanism. Communication induced checkpoints come into play in this scenario. The problem with these check points are that, when a processP2 fails that is dependent on processP1, there would not be a consistent roll back recovery. Here, the rollback in the process P2 may cause a rollback in the communicating process P1 too. This communication may lead to a rollback in another process thereby forming a series of rollbacks (Domino's effect). Hence, time based checkpoints would not be sufficient for a successful rollback mechanism.

To overcome this issue, we move to forced checkpoints that are created only when there is a communication between the processors in a network.

V. COMMUNICATION INDUCED CHECK POINT(FORCED CHECK POINT)

Communication induced check points are created when there is a communication between processes in a grid network. These checkpoints are not time based and are created only in the processes those are in need of it. These types of checkpoints are not deleted until the entire process is complete. When a process fails during communication, the status of the communication checkpoints would let the process to rollback to its previous state by decreasing unnecessary successive rollbacks (Domino's effect). As these check points are dependent on the processes, the number of communication induced check points created are lesser than the number of time based check points created.

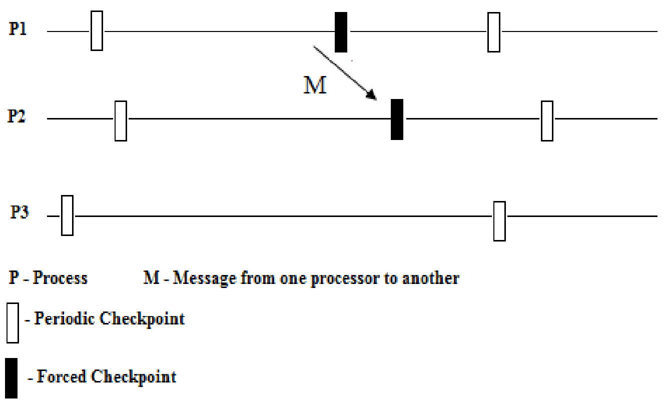

Fig. 2 Communication Induced Checkpoints

VI. EXPERIMENTAL EVALUATION

This section illustrates the experimental evaluation of implementing time based checkpoints and compares it with uncoordinated checkpoints in a grid environment. The grid environment is implemented using Java RMI .Remote method is a specification that enables one java virtual machine (JVM) to invoke methods in an object located in another java virtual machine. In RMI, two JVMs are running on different computers or running as separate processes in the same computer. In general, an RMI client server model object is implemented with a single server along with multiple clients that accesses that server. However, in this experimental set up, a single client acts as a job scheduler and contains multiple servers that perform subtasks/tasks.

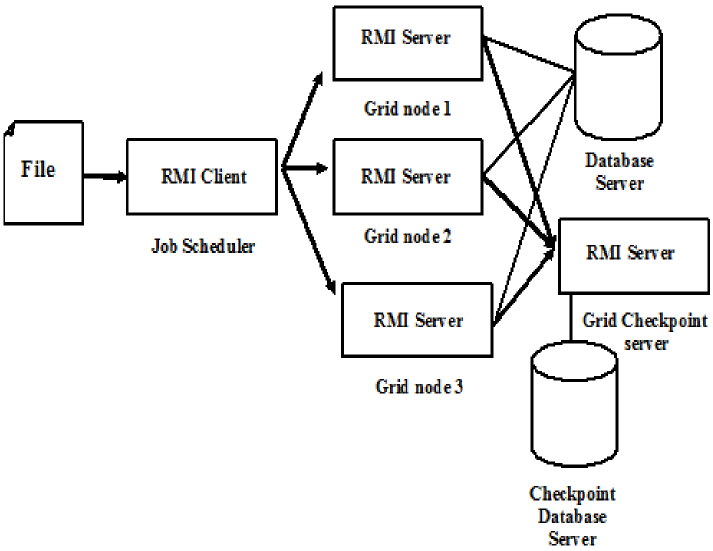

Fig. 3 Grid Implementation Architecture

Here, the task to be completed is considered as a file containing large number of records. The job scheduler counts the number of records in the file and the number of available nodes in the grid environment .The scheduler then divides the records in such a way that equal number of subtasks are allocated to the available grid nodes (here the servers are the nodes). Every time a subtask is completed, all the nodes save the completed subtasks in the database server. The statuses of these tasks are periodically stored in the checkpoint database server. This would create a large number of checkpoints in the checkpoint database server. To overcome this, an intermediate RMI checkpoint server is introduced, which deletes the already available checkpoints, when a new consistent checkpoint is saved.

A. Uncoordinated Checkpoint Vs Time Based Checkpoint

Uncoordinated checkpoints store the current process state periodically. This causes checkpoint redundancy which in turn is overcome by the concept of time based checkpoints.

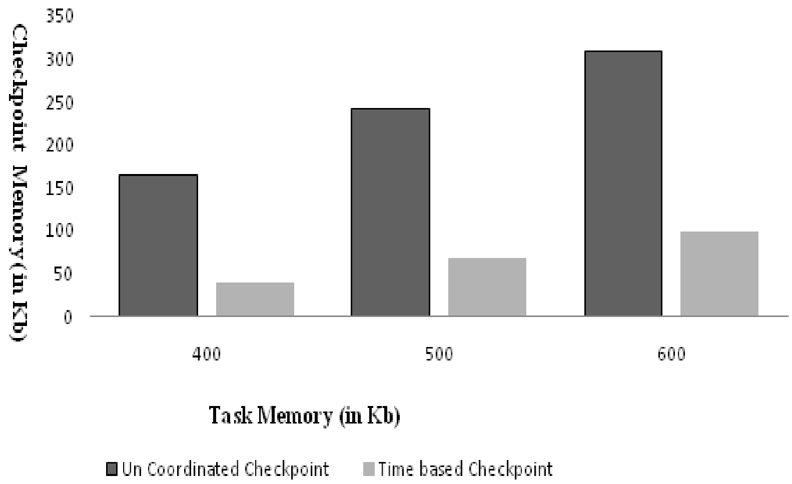

Fig. 4 Memory comparison

In this implementation, records in a file are copied in to remote database by using three grid nodes. Copying the file to the database is the task to be completed by the grid nodes and the status of the process is stored in checkpoints. For a file of size 300kb, the checkpoints occupy 100 kb memory in the conventional checkpoint technique. When the proposed method is implement the memory require in storing the checkpoints reduces drastically to 40 kb. This depicts that time based checkpoints are a far way better than the conventional checkpoint techniques in considering the memory used.

VII. CONCLUSION

This paper addresses the problem created by the conventional rollback recovery in a grid environment. It suggests the use of time based checkpoints which would considerably decrease the wastage of memory. Furthermore, a communication induced checkpoints approach is also provided, that would decrease the Domino effect caused by successive rollbacks. This paper therefore produces a fault tolerant efficient mechanism in a grid environment.

REFERENCES

[1] Ge-Ming Chiu; Jane-FerngChiu,"A New Diskless Checkpointing Approach for Multiple Processor Failures" IEEE Transactions on Dependable and Secure Computing vol.1, No.205, August 2011.

[2] Gupta B,RahimiS,Liu Z., Novel low-overhead roll-forward recovery scheme for distributed systems, IEEE journals on Computers & Digital Techniques, Vol.1,Issue.4, July 2007

[3] Hui Liu, Davies T.Chong Ding, Karlsson C., Zizhong Chen "Algorithm-Based Recovery for Newton's Method without Checkpointing", IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW),September 2011.

[4] Samir Jafar, Axel Krings, Senior Member, and Thierry Gautier," Flexible Rollback Recovery in Dynamic Heterogeneous Grid Computing", IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, VOL.22, NO. 2, JANUARY 2008.

[5] Sanjay Bansal, Sanjeev Sharma," Identification of Critical Factors in Check pointing Based Multiple Fault Tolerance for Distributed System", Journal of Emerging Trends in Computing and Information Sciences.

[6] S.BaghavathiPriya, Dr.T.Ravichandran," Fault Tolerance and Recovery for Grid Application Reliability using Check Pointing Mechanism", International Journal of Computer Applications (0975 – 8887)Volume 26– No.5, July 2011.

[7] Eugen Feller, John Mehnert-Spahn, Michael Schoettner , Christine Morin," Independent Checkpointing in a Heterogeneous Grid Environment", Rapport de recherche n° 7399 _ September 2010.

[8] Jean-Louis Roch, Thierry Gautier, Axel W. Krings, "Theft-Induced Checkpointing for Reconfigurable Dataflow Applications" ,Laboratoire ID-IMAG (CNRS-INPG-INRIA-UJF – UMR 5132)38330 Montbonnot Saint-Martin, France.

[9] Raphael Y. de Camargo, Andrei Goldchleger, Fabio Kon, Alfredo Goldman, "Checkpointing-based rollback recovery for parallel applications on the InteGrade Grid middleware", Dept. of Computer Science - University of S˜ao Paulo Rua do Mat˜ao, 1010. 05508-090, S˜ao Paulo - SP, Brazil.

[10] E.N. (Mootaz) Elnozahy, Lorenzo Alvisi, Yi-Min Wang, David B. Johnson, "A Survey of Rollback-Recovery Protocols in Message-Passing Systems".

[11] White paper, "Grid Computing and SAS" 2004.

[12] Ropars, T.;Brunet, E.; Snir, M.;Cappello, F," Uncoordinated Checkpointing Without Domino Effect for Send-Deterministic MPI Applications ", International Parallel & Distributed Processing Symposium (IPDPS), May 2011.