ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

A SURVEY OF DISTRIBUTED FAULT TOLERANCE STRATEGIES

SunilGavaskar.P [1], Subbarao Ch D.V [2]

Research Scholar, Department of Computer Science and Engineering, S.V.University, Tirupathi, India [1]

Professor, Department of Computer Science and Engineering, S.V.University, Tirupathi, India [2]

Abstract: Grid computing is defined as geographically distributed, heterogeneity (different hardware, software and networks), resource sharing, multiple administrators, dependable access, and Pervasive access within dynamic organizations. In grid computing, the rate of failure is much greater than in traditional parallel computing. Therefore, the fault tolerance is an important property in order to achieve reliability, availability and QOS. In this paper, we give a survey on various fault tolerance techniques and fault management in different situations with related issues. The fault tolerance service deals with various types of resource failures. This survey provides the related research results about fault tolerance in grid infrastructure and also the future directions about fault tolerance techniques, and this survey attempts to provide guide for researcher.

Keywords: Fault Masking, Failure, Replication, Reliability, Availability.

I. INTRODUCTION

Cluster systems are becoming widely used because they can achieve high performance at very low cost. According to the TOP500 list of supercomputer sites in the world for November2003 [1], 208 machines are cluster systems. These cluster system consists of thousands of processing nodes. The cluster system consists of many components such as node processors, disks, and networks called resources, increasing the number of nodes greatly degrades the overall system reliability.

Failure recovery approaches for cluster systems should satisfy the following requirements (a).**Low performance overhead:** cluster system must achieve good performance (b) **Low cost overhead:** Failure recovery scheme of cluster system must have low cost.(c).**High fault coverage:** If Number of nodes increases, those does not leads to high reliability of each node, it is essential to tolerate multi-node failures[5,6].(i.e. Individual node reliability should be low and overall reliability should be high called fault masking in fault tolerance.).When Cluster computing nodes are tightly connected to one another by low latency and high throughput networks such as Gigabit Ethernet or Myrinet, the scheme of coordinated checkpointing is a best way to keep the high availability of cluster systems.

Uncoordinated checkpointing allows each process to take a checkpoint when it is convenient, this is attractive when distributed parallel systems like grid environments. One of the disadvantages of this technique is the possibility of the domino effect [7]. To avoid this message logging techniques are used, the form of message logging is not suitable for engineering and scientific applications because messages are moved frequently between processing nodes, therefore total size of such message log becomes huge if messages are logged. Checkpointing analysis uses the following five criteria: Checkpoint time, recovery time, rollback degree, disk consumption, and fault coverage.

Checkpoint time: Time required transferring the checkpoint from one node to another node.

Recovery time: Time required to recovering from failure using obtained checkpoint.

Rollback Degree: Number of Checkpoints used to obtain previous state or erroneous state. It depends on the frequency of the N-checkpoint.

Disk Consumption: The Disk space required for storing checkpoints with respect to number of failures.

Fault Coverage: Successfully configuring from failures. Some time it depends on Degree of mirroring, disk space and thus it is not practical.

When Compared to the Uncoordinated Checkpointing, in Coordinated checkpointing, all the processors coordinated to define a global consistent state and then save that state to storage, it is attractive than uncoordinated checkpointing due to the saving of states[8].

Grid computing is form of distributed computing that involves hardware and software infrastructure, that enables coordinated of resource sharing(Computational power, data, storage and network resources) across the dynamic and geographically dispersed organizations[2,9].Management of these resources becomes complex as the resources are heterogeneous, geographically distributed in nature, owned by different individual or organizations with their own

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

policies, have dynamically varying loads and availability, have different access models[9].

In order to achieve better performance of computational grids, the fault tolerance is basically important since the resources are dynamic and geographically distributed in nature. However the probability of a failure is much greater than in traditional parallel computing and the failure of resources affects job execution. Fault tolerance is the ability of a system to perform its function successfully even in the presence of faults and it makes the system more dependable. The fault tolerance is an important attribute of computer system in presence of fault caused errors within the system itself, errors are detected and corrected. The permanent fault are located and removed while the system continues to deliver acceptable service. Fault tolerance service is essential to satisfy QOS requirements in grid computing. Fault tolerance service deals with resource failures (Process failure, processor failure and network failures), the resource failure/job failure may lead to violating timing deadlines and Service Level Agreement (SLA), which causes degraded user expected QOS.

The rest of the paper is organized as follows. Failure identification techniques are given in II. Fault tolerance techniques are presented in section III. Fault tolerance in Grid environment is presented in IV. Conclusions are given in section V.

II. FAILURE IDENTIFICATION

Fault tolerance Attributes: Fault tolerance consists of: (1) detecting faults and failures in grid resources. (2).recoveries to run system successfully. Fault tolerance dependability is related to QOS aspects provided by the system, such as reliability and availability.

(A).Reliability: Reliability indicates that a system can run continuously without failure. Reliability is closely related to Mean Time to Failure (MTTF) and Mean Time between Failures (MTBF).MTTF is the average time the system operates until a failure occurs, MTBF is the average time between two consecutive failures. Mean Time to Repair (MTTR) is difference between MTTF and MTBF, we obtain MTBF=MTTF+MTTR.

(B).Availability: Availability ratio is calculated using the factors like MTTF, MTBF and MTTR as follows: Availability=MTTF/MTBF=MTTF/ (MTTF+MTTR).

Example: A system that fails every hour on the average but comes back up after only a second. Such a system has the rate of MTBF becomes just 1 hour, as result of that a low reliability and availability of the system is always high: Availability=4699/4700=0.99978.

Threats types are: (a).Faults or Failure: Fault or failure can be either a hardware defect or a software bug. (b).Errors: an error is a manifestation of the fault/failure/bug.

The hardware faults can be classified into permanent, transient, intermittent, benign or malicious. In general agent oriented fault tolerance grid framework was used in which faults are divided into six categories.

(1).Hardware Faults: CPU, memory, storage.

(2).Application and OS Faults: memory leaks, resource unavailable

(3).Network Faults: node failure, link failure, packet loss

(4).Software Faults: Un-handles exception, unexpected input

(5).Response Faults: Value faults

(6).Timeout faults

In grid resource management for fault detection, there are two ways [12]: (1).Pull model: It uses different grid components to send periodic signals to fault detector, If signals becomes absent from the grid component then the fault detector recognizes that failure has occurred at particular component, as result of that the fault detector implements appropriate measures which are predefined in fault tolerance. (2).Push model: Fault detector send signals to the grid components. However, the fault detector component is responsible for detecting failures such as network failures, node crashes, process and processing failures.

Agents maintain information about executing process memory, available resources, network conditions and component mean time to failure, however on the basis of critical states and information the agent enables the grid system to tolerate faults [11].The omission faults will arise when resources become unavailable. Interaction faults may be due to different services supporting different protocols, security incompatibilities and policy problems. Timing faults will arise if one service locks another service when Timeout occurs.

III. FAULT TOLERANCE TECHNIQUES

(1).Fault masking: The fault masking technique is to prevent faults in a system from introducing errors. Examples: Error correcting techniques and majority voting.

(2).Reconfiguration: The Reconfiguration technique is to eliminate the faulty component and restore system to some operational state. The following are some of the applicable reconfiguration approaches Fault detection, Fault location, Fault containment, Fault recovery.

(3).Fault tolerance mechanisms: There are two types of fault tolerance mechanisms (a).Proactive: In grid environment the failure considered before job scheduling, and delivers job for execution.(b).postactive:It handles the job failure after it has occurred[12].

(4) Fault tolerance recovery: In grid systems the recovery methods rely on exploitation of redundancy, which is key idea to fault tolerance. In grid systems without redundancy there is no fault tolerance, here redundancy are two types (a).Temporal redundancy: This kind of redundancy performs repeated attempts to restart failed resources or services. (b).Spatial redundancy: The spatial redundancy provides

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

multiple copies of computing resources. Three techniques that probably used in the spatial redundancy are check-pointing, replication, rescheduling or identifying different resources to rerun failed tasks. Redundancy types presented in table1 are used along with static, dynamic, and hybrid configurations. Fault tolerance in grid systems are achieved as follows.

TABLE I
REDUNDANCY TYPES

Hardware Redundancy Hardware redundancy is Provided with additional hardware to override the effects of a failed component. For example, instead of having single processor, we can use two or three processors .	**Software Redundancy** Independently produces two or more versions of that software or program to run same set of inputs. This kind of design diversity ensures that same set of inputs will not fails in all versions.
Information Redundancy Here additional bits are added to the original data bits so that an error in the data bits can be detected or even corrected. It requires the hardware redundancy to perform or process the additional check bits.	**Time Redundancy** Additional time needed to process the function of a system as result of that fault detection and fault tolerance is achieved. (i.e. Re-execution of same task or program on the same component)

A. Performance Overhead in Checkpointing

The performance overhead in checkpointing consists of checkpoint overhead and recovery overhead. Checkpoint overhead usually dominates over recovery overhead in coordinated checkpointing; the coordinated checkpointing does not suffer from the rollback propagation problem. Vaidya used two metrics for checkpoint overhead scheme [4]. (a).checkpoint overhead: The increase in the execution time of an applications caused by the check pointing. (b).checkpoint latency: The duration of time required to save the checkpoint.

Eminent research has been carried out so far to reduce checkpoint overhead.

Copy-on-write [10] and incremental checkpointing [5] reduce the size of a checkpoint by saving only the updated part. These methods can reduce both overhead and latency, but are not very effective in large scientific or engineering applications because most of the data arrays are updated during execution. However this is NOT true if checkpointing interferes with the system bandwidth required by application programs.

From the above the failure recovery schemes for cluster systems should satisfy the following three requirements. The first is low performance overhead. Since one of the major advantages of the cluster systems is the good performance / cost ratio, the second characteristic, low cost overhead. The third is high fault coverage, because the number of nodes increases and the reliability of each node is not very high, it is essential to tolerance multi-node failures. The main problem with co-ordinated checkpointing is its lack of scalability. The computation interval and the checkpoints overhead are much smaller than the mean time between failures (MTBF).

B. Recovery in Checkpointing

Recovery in checkpointing emphasis on the MTTR, in checkpointing application periodically saves state of its execution as checkpoints in hard disk, if failure occurs system restarted from the last checkpoint rather than from the beginning. Some of the strategies used in checkpointing are **coordinated checkpointing,** processes synchronize checkpoints to ensure their saved states are consistent with each other, so that the overall combined, saved state is also consistent, here number of computing nodes gets larger the probability of multi-node failures increases which leads to a large degree of redundancy is required in checkpointing. In **Uncoordinated checkpointing** scheme, processes schedule checkpoints independently at different times and this scheme do not account for the messages. **Communication-induced** checkpointing attempts to coordinate only selected critical checkpoints[13].Some of the multiple node failures checkpoint schemes[14,15,16] are **Central file server**(CFS) where all the processors checkpoint to a stable central file system without checkpointing to their local disks, because it increases the checkpointing time and hence degrades performance. **Checkpoint mirroring** (MIR), where each processor checkpoints to its local disk and copies its checkpoint to another processors local disk for tolerating multiple node failures. **Skewed checkpointing**, in this method checkpointing is skewed every time. In skewed checkpointing, although each checkpointing itself contains only one degree of redundancy. **On demand checkpointing** the checkpointing is allowed only on the basis of Independent and dependent task considerations, the task dependency is considered to identify the best scheduling and corresponding checkpoints are activated to improve the reliability of fault tolerance model. The concurrent tasks that should always maintain lower checkpoints than that of the independent tasks [17].

B. Replication

The goal of replication is to ensure that at least one replica is always able to complete the computation in the event others fail. However one replica may be designated as a primary copy for purposes of external interaction, whereas others assume the role of backups. Replication is widely used as a fault tolerance technique but number of backups is a main drawback. In replication, number of backups increase as result of that number of fault coverage increases which causes management of backup is very costly. Fusion based techniques overcome this problem. It is emerging as a popular technique to handle multiple faults. Replication depends on the availability of alternative present to run replicas. Some issues in replication are: consistency, number of replicas as a reaction on changing system properties

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

(number of active resources, resource failure frequency and system load), degree of replication, and replica on demand.

C. Fault tolerant network measures

The fault tolerant network measures can be obtained by using multiple network links and spare nodes connecting it or replace failed nodes.
(1).Reliability: The reliability of network depends on probability of all nodes are operational and can properly communicate with each other in entire time interval. (i.e. between 0, t). (2).Bandwidth: The maximum rate at which messages can flow in a network. It degrades nodes or links fail in a network. (3).Routing: Generally in fault tolerance crossbar network or rectangular mesh networks are used. Routing strategy is to get a message from source to destination despite a subset of the network being faulty [18].In routing if no shortest path or most convenient path is available because of link or node failures, reroutes the message through other paths to its destination.

IV. FAULT TOLERANCE IN GRID ENVIRONMENT

The basic grid model consists of number of hosts, which contains several computational resources in heterogeneous or homogeneous in nature.

A. *Levels of Fault tolerance in Grid*

We analysed that there are two levels of fault tolerance **Service level fault tolerance**, it discusses about system wide policies aiming to increase dependability of services provided. Dependability of system is the ability to avoid service failures that are more frequent and more severe than is acceptable. Here Quality of Service is the factor. The second type is **Resource level fault tolerance**; here fault tolerance works at application level in each and every one of the resources in the system. Resource level failures occur due to heterogeneous nature of the system and increases complexity in system. Examples of resource level failures are node crashes, link failures, loss of network link, unexpected machine turns.

B. *Fault tolerance Scheduling*

Some of the basic building blocks of grid model composed of user, resource broker, grid information service (GIS), resource. At first job is submitted to the broker then Broker splits the job into various tasks and distributes to several resources according to users requirements and availability of resources. GIS maintains information of all resources and helps the broker for scheduling. When resource broker schedule job the GIS maintains Resource Fault Occurrence History (RFOH), on the basis of this information uses different intensity of checkpointing and replication while scheduling the job on resources which have different tendency towards fault, it increases the percentage of jobs executed within specified time [19].Using RFOH information in genetic algorithm we can optimize resource while job scheduling [20]. It reduces the probability of selecting resources that have more fault occurrence information. Grid Tuple Space (GRIDTS) is a decentralized and fault tolerant grid infrastructure [21] to pick resources and execute jobs, instead of using centralized scheduler. The GRIDTS uses various fault tolerance techniques like checkpointing, transaction, and replication.

Job Scheduling it's kind of proactive fault tolerance, Job scheduling is mapping of jobs to specific physical resources, the NP-complete problem and different heuristics may be used to reach an optimal solution. **Resource Scheduling**, the resource scheduling is process of mapping resources based on the requirements, characteristics which are specified in a query. Various individual resources are centrally controlled, these resources enters or leave the grid system at any time. For the above said reasons resource scheduling can be very challenging in large scale grids.

C .Replication in Grid Environment

Based on the deep research on the replica management in traditional data grid, some of the proposed strategies are dynamic replica strategy here replicas creation on the basis of number of file access and node load, selection of replicas and replacement of replicas depends on creation time, number of file access and file size of replicas, this kind of strategy provides low response time and rapid data download and good performance [22]. A Priori Replica strategy optimizes distance between the data hosted on the grid as result of that gain in execution time due to the improvement in the file transfer time[23].Popularity based replica placement works for hierarchical data grids by minimizing replication costs(both accesses and updates).Its a decentralized algorithm and avoids limitations in centralized techniques. Object oriented replicas, here agents are used to store/maintain replicas as objects and its states therefore maintaining suitable number of replicas is possible, load on nodes can be easily handled [24].

C .Recovery of workflows in Grid Environment

Directed Acyclic Graph (DAG) has been widely used in scientific computational workflow modelling, workflow execution failure can occur for various reasons: variations in execution environment configuration, lack of availability of required services, resource conditions overloaded, system running out of memory. In order to identify and handle failures then support for reliable execution, and divided the workflow failure handling into two levels[25] namely task level and workflow-level and provide user-defined exception

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 2, Issue 11, November 2013

to specify treatment for certain failures which occurs in data movement.

V. CONCLUSION

This study surveyed from fault tolerance focus on distinct functional areas to progress in making grid system more reliable. In order to provide reliable grid systems, it includes software resources, user applications, checkpointing, scheduling, Agents strategies, load balancing, and Workflows. Our survey discusses about all the above areas with different problems and some problems remain to be solved. Finally, Grid middleware's, self adaptive fault tolerance framework, workflow management based Service-oriented Architecture (SOA), SLA are more dependable and trustworthy in grid environments.

ACKNOWLEDGMENT

We deeply thank Dr. Ch D.V.Subbarao for his valuable help.

REFERENCES

[1] http://www.top500.org.
[2] I.Foster and C.kesselman, (2004) "The Grid: Blueprint for a New Computing Infrastructure, 2nd Edition, and pp: 748.
[3] S.I.Fieldman and C.B.Brown,"Igor: A System for Program Debugging via Reversible Execution",ACM SIGPLAN,Work shop on Parallel and Distributed Debugging,Vol.24,No.1,pp.112-123,1989.
[4] N.H.Vaidya,"Impact on checkpoint latency on overhead ratio of Checkpointing Scheme", IEEE Transactions on Computers,pp.942-947,Vol.46,No.8,1997.
[5] J.S.Plank and K.Li, "Ickp-A Consistent Checkpointer for Multicomputer", IEEE Parallel and Distributed Technology,vol.2,No.2,pp.62-67,1994.
[6] P.SunilGavaskar and Dr. Ch D.V.Subbarao,"Fault Tolerance Agents In Grid Environment", International journal of Advanced Research in Computer and Communication Engineering(IJARCCE),Vol.2,Issue 10,pp.3980-3982,October 2013.
[7] A.Agbaria, H.Attiya, R.Friedman, and R.Vitenberg,"Quantifying Rollback Propagation in Distributed Checkpointing",In Proc.of SRDS2001,pp.36-45,Oct.,2001.
[8] L.M.Silva,"Checkpointing Mechanisms for Scientific Parallel Applications", Ph.D.Thesis, University of Coimbra, 1997.
[9] Ian Foster and Carl Kesselman,S.T.,(2001) "The Anatomy of the Grid: Enabling Scalable Virtual Organizations",Intl J. Supercomputer Applications, 15:200-222.
[10] E.N. Elnozahy,D.B.Johnson and W.Zwaenepoel,"The Performance of Consistent Checkpointing",In Proc.of SRDS'92,pp.39-47,Oct.1992.
[11] Mohammad Tanvir Huda,Heinz and W.Schmidt,Ian D.Peake, (2005) "An Agent Oriented proactive Fault-tolerant framework for Grid Computing",In proceedings of the First IEEE International Conference on e-Science and Grid computing,PP.304-311.
[12] Nazir,B. and Khan, T.,(2006)"Fault Tolerant job Scheduling in Computational Grid.Emerging Technologies,In IEEE International Conference on Emerging Technologies,Volume,Issue,13-14.
[13] Eric roman, (2002) "A Survey of checkpoint / restart Implementations", Lawrence Berkley National Laboratory, CA.
[14] J.S.Plank,"Improving the Performance of Coordinated Checkpoints on Networks of Workstations using RAID Techniques",In Proc.of SRDS'96,pp.76-85,1996.
[15] N.H.Vaidya," A Case for Two-Level Recovery Schemes",IEEE Transactions on Computers,Vol.47,No.6,pp.656-666,June.1998.
[16] K.Hwang, H.Jin, E.Chow, C.Wang and Z.Xu,"Designing SSI Clusters with Hierarchical Checkpointing and Single I/O Space",IEEE Concurrency,Vol.7(1),pp.60-69,Jan-March1999.
[17] Partha Sarathi Mandal, Krishnendu Mukhopadhyaya, "Performance analysis of different checkpointing and recovery schemes using stochastic model",Journal of Parallel and Distributed Computing,66,99-107,2006..
[18] Israel Koren and C.Mani Krishna, (2007) "Fault Tolerant Systems", Elsevier Inc., ISBN:978-81-312-1530-2.
[19] Leyli Mohammad Khanli, (2010) "Reliable Job Scheduler using RFOH in Grid Computing", Journal of Emerging Trends in Computing and Information Sciences,Vol.1,No.1.
[20] Leyli Mohammad Khanli, Maryam Etminan and Far Amir Masoud Rahmani, (2010) "RFOH: A New Fault Tolerant Job Scheduler in Grid Computing", In Second International Conference on Computer Engineering and Applications.
[21] Fabio Favarim, joni da silva Fraga, lau Cheuk Lung and Miguel Correia, (2007) "GRID TS:A New Approach for Fault-Tolerant Scheduling in Grid Computing", In International Symposium on Network Computing and applications,pages:187-194.
[22] Ranganathan K, Foster I, "Identifying dynamic Replica strategies for a high performance data grid[C]".In Proceeding of the International Grid Computing workshop.Berlin:Springer-Verlag,2001: 75-86.
[23] M.Garmehi and Y.Mansouri, "Optimal placement replication on data grid environments", in 10th International Conference on Information Technology (ICIT 2007), pp.190-195.
[24] P.SunilGavaskar and Dr.Ch D.V.Subbarao,"Metadata Control Agent approach for Replication in Grid Environments"International Journal of Electronics and Computer Science Engineering,ISSN-2277-1956,Volume2,Number4,2013.pp.1156-1161.
[25] S.Hwang and C.Kesselman, (2003) "Grid Workflow: A Flexible failure handling framework for the Grid",In Proceedings of the 12th IEEE International Symposium on High Performance Distributed Computing,Seattle,Washington,USA.

BIOGRAPHIES

P. SunilGavaskar is a PhD candidate at the Department of Computer Science and Engineering, Sri Venkateswara University; Tirupathi.His research interests include distributed systems, Grid Computing. Contact him at sunil079@gmail.com.

Dr. Ch D.V.Subbarao is a professor and Head of the Department of Computer Science and Engineering, Sri Venkateswara University; Tirupathi. His research interests include distributed systems, Grid Computing. Contact him at Subbarao_Chdv@hotmail.com.