International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 02 | Feb -2017 www.irjet.net p-ISSN: 2395-0072

A Brief Review Of Approaches For Fault Tolerance In Distributed Systems

Shwethashree A[1], Swathi D V[2]

[1]*Asst.Prof., Ballari Institute of Technology and Management, Ballari, Karnataka, India*
[2] *Asst.Prof., Ballari Institute of Technology and Management, Ballari, Karnataka, India*

Abstract - *Distributed information processing systems have evolved over the years and are in the main stream of computing systems. The major concern in distributed systems is ensuring the predefined level of reliability and availability. These systems are prone to failure because of their high complexity. Hence fault tolerance becomes the major issue to be addressed in designing these systems. This paper provides the study of various approaches for fault tolerance.*

Key Words: **Distributed system, Fault tolerance ,Redundancy, Replication, Dependability**

1.INTRODUCTION

Distributed systems consists of group of autonomous computer systems brought together to provide a set of complex functionalities or services. The computer systems are geographically distributed and are heterogeneous in nature. Distributed systems appear as one local machine to the users. These systems are advantageous as they provide scalability of software and resources dynamically. Distributed systems are required to be dependable having following characteristics.

- Systems must be available, must not fail.
- Must fulfill timing and requirements.
- Systems output is required to be accurate.
- System must be secure
- System must provide safe mode operations

Thus, the dependability refers to reliability, availability, survivability and safety. To achieve these system characteristics, system must be designed to have the ability to handle faults and failures dynamically.

In large and dynamic distributed system millions of computing devices are working together. Faults and failures are inevitable in such complex design. Failures can cause serious damage to the users. In systems such as online railway ticket booking systems, net banking systems failure may lead to loss of money. Hence implementation of fault tolerance techniques becomes the key factor of concern.

2. FAULT TOLERANCE APPROACHES

Fault tolerance approaches can be classified into two types: Proactive and Reactive. Proactive approaches predict errors, faults and failures and replace the suspected components where as reactive approaches reduce the effect of faults by taking necessary actions. Some fault treatment policies can also be used to prevent faults from being reactivated.

3. REDUNDANCY BASED FAULT TOLERANCE

Redundancy is having more than one functionally ready components of a system other than a component that actually provides the service. At two levels we can implement the redundancy: Process level and data or object level. This approach uses replication technique to create redundancy. Replication is the process of creating and maintaining multiple copies of data objects or processes.

3.1 OBJECT LEVEL REPLICATION

In this approach multiple replicas of data items or objects are created and maintained at different nodes in the system[1]. The incoming request is served using one of the replicas. In this way failure of any node will not affect the functionality of the system. There are few major issues to be addressed while using replication like maintaining consistency of replicas, degree of replica and on demand replication.

When user updates the information of an object, replication manager must update all replicas to ensure the consistency among replicas of same object[2]. It is important to have efficient strategy for managing consistency. In a passive strategy, a primary executes and updates state changes to all replicas where as in active strategy all replicas execute individually. Passive method avoids redundant execution and active method has comparatively low response time. These approaches have advantages and also disadvantages. Researchers have proposed various algorithms in this regard. Simple and adaptive algorithms are comparatively efficient.

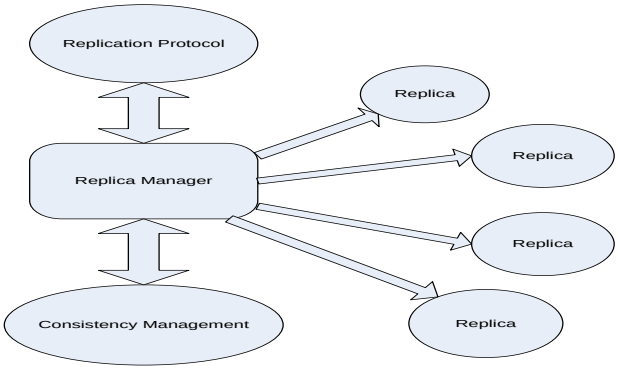

Fig 1. Replication technique

Replication protocol specifies the process of replication of objects. Number of replicas is also referred as degree of replica. The replication protocol has to create optimal number of replicas. Too many replicas are hard to manage consistency where as less number of replicas lead to low performance. Various replication protocols are proposed by the researchers. Voting, Primary backup replication, primary per partition are some of the protocols used[4].

3.2 PROCESS LEVEL REDUNDANCY

Process level redundancy is a method of creating redundant application processes. It's a software based technique to handle transient faults. Transient faults are short lived and less severe. They are caused by temporary malfunction of any component in the system. Some time it is hard to diagnose these faults. Process level redundancy handles these kind of faults by leveraging processes across multiple cores as shown in the figure.

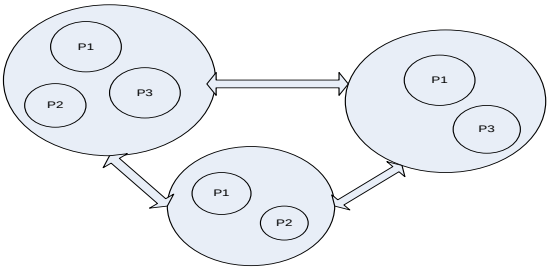

Fig. 3 Process redundancy

Process level redundancy allows the operating system to schedule processes across all available hardware resources. In this method we can ignore the faults that do not propagate to affect program correctness. This method provides improved performance with additional overhead of fault detection. Further there is research scope for adaptive algorithms using process level redundancy.

4. CHECK POINTING TECHNIQUE

In this technique current computation state is stored in a stable storage. Check points are established in the normal execution periodically. At the time of failure, the operation is rolled back to most recent state in the stable storage. This technique allows the system to recover from failure and continue the operation normally. There are two types of rollback recovery. Checkpoint based rollback recovery and log based rollback recovery. The former method uses only check pointing technique where as latter uses check pointing with logging of non deterministic events.

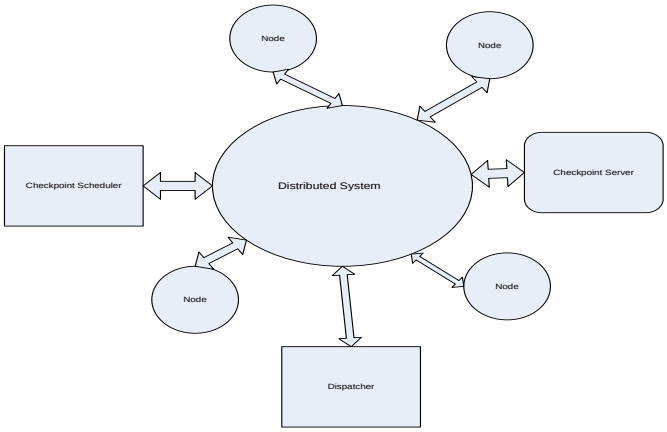

Fig 2. Check Pointing Technique

To establish checkpoints periodically there are two techniques. Coordinated check pointing and uncoordinated check pointing[3]. In coordinated check pointing processes coordinate their checkpoints to maintain a system wide consistent state. It involves rollback check point of all the processes from last snapshot, even when single process crashes. This is time consuming but required to ensure the system wide consistent state. Uncoordinated check pointing involves message logging. Here all the processes are made to execute check point independent of others. With logging we get all the description of execution states of processes in case of failure.

Check pointing technique is comparatively costly as it consumes various resources and recovery takes lot of time. There are various issues involved with establishing check points, their storage location, time required to run and frequency of check pointing. There is lot of scope to improve the overall performance of this technique.

5. FUSION BASED TECHNIQUE

Redundancy based fault tolerance is widely used but the major problem of this technique is managing the backups. In fusion based approach, it requires a few backup machines which can be managed easily[6]. Here the backup machines are fused corresponding to the given set of machines[7]. This technique has very high overhead during recovery process. Hence it is suitable for systems where probability of failures is low.

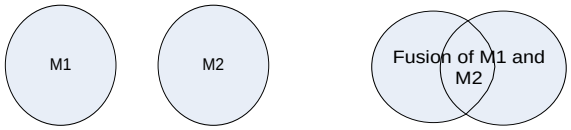

Fig. 4 Fusion Based Technique

6. DYNAMIC AND SELF ADAPTIVE FAULT TOLERANCE ALGORITHMS

Current trend of distributed applications demand dynamic and self adaptive fault tolerance techniques. The techniques must be capable of handling frequent and multiple faults at the run time and also adaptive to different run time conditions. Adaptive programming model can be used to develop such techniques[5]. There is lot of research scope for developing programming models for implementing adaptive techniques.

In large scale distributed system failure detection is fundamental task for ensuring fault tolerance. Failure detectors must be capable of working asynchronously and independent of application flow. The major issue with these is their ability to scale for large number of nodes. Handling multiple faults is becoming crucial as number of nodes scale in distributed system. Single point failure often cause serious problems, hence must given much attention.

7. COMPARISON

Choosing a fault tolerance technique appropriate for a system is an important task[8]. We can evaluate the techniques based on some major factors like consistency management, multiple faults handling, and efficiency of working procedure, multiple failure detection and performance.

Comparison shows these techniques are reliable and have the capability of detecting and handling multiple faults. The performance can be improved by working on the critical issues.

Table -1: Comparison of fault tolerance technique

Factors	Replication based	Process level redundancy	Check point based	Fusion based
Consistency management	Strategies like active or passive replication	Can be easily implemented	Can be handled by avoiding orphan messages	Has to be implemented among back up machines
Multiple faults handling	Affected by degree of replica	Affected by number of redundant processes	Affected by check point scheduling	Affected by number of back up machines
working	Requests are forwarded to replica/s	Redundant processes	Snapshots saved on stable storage for recovery	Back up machine
Multiple failure Detection	Accurate and adaptive	Reliable, accurate and adaptive	Reliable, accurate and adaptive	Reliable, accurate and adaptive

8.CONCLUSIONS

The fault tolerance of a distributed system is a characteristic that makes the system more reliable and dependable. The fault detection and fault recovery are the two stages in fault tolerance. The fault tolerance approaches discussed in this paper are reliable techniques. Further the performance of these can be improved towards achieving high reliability. There is lot of research scope in minimizing recovery time of existing techniques and implementing dynamic adaptable techniques.

REFERENCES

[1] M. Wiesmann, F. Pedone, A. Schiper, B. Kemme, G.Alonso," Understanding Replication in Databases and Distributed Systems," Research supported by EPFLETHZ DRAGON project and OFES).
[2] A. Kale, U. Bharambe, "Highly available fault tolerant distributed computing using reflection and replication," Proceedings of the International Conference on Advances in Computing, Communication and Control ,Mumbai, India Pages: 251-256 ,: 2009

[3] M. Elnozahy, L. Alvisi, Y. M. Wang, and D. B. Johnson. A survey of rollback-recovery protocols in message passing systems. Technical Report CMU-CS-96-81, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, October 1996.

[4] J. Walters and V. Chaudhary," Replication-Based Fault Tolerance for MPI Applications," Ieee Transactions On Parallel And Distributed Systems, Vol. 20, No. 7, July 2009

[5] Cao Huaihu, Zhu Jianming, "An Adaptive Replicas Creation Algorithm with Fault Tolerance in the Distributed Storage Network" 2008 IEEE..

[6] N. Budhiraja, K. Marzullo, F.B. Schneider, and S. Toueg. The Primary-Backup Approach. In Sape Mullender, editor, Distributed Systems, pages 199-216. ACM Press,1993.

[7] V.K Garg,. "Implementing fault-tolerant services using fused state machines," Tech-nical Report ECE-PDS-2010- 001, Parallel and Distributed Systems Laboratory,ECE Dept. University of Texas at Austin (2010).[8] N. Xiong, M. Cao, J. He and L. Shu, "A Survey on Fault tolerance in Distributed Network Systems," 2009 International Conference on Computational Science, 978-0-7695-3823-5/09