International Journal of Computer Engineering & Technology (IJCET)
Volume 7, Issue 1, Jan-Feb 2016, pp. 94-99, Article ID: IJCET_07_01_011
Available online at
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=7&IType=1
Journal Impact Factor (2016): 9.3590 (Calculated by GISI) www.jifactor.com
ISSN Print: 0976-6367 and ISSN Online: 0976–6375

DISASTER DETECTION PLANS IN DISTRIBUTED SYSTEMS

Dhanashri D. Dhokate
Assistant Prof, Information Tech,
PVPIT, Budhgaon (Sangli)

B. S. Patil
Associate Prof. Electronics Department
PVPIT, Budhgaon (Sangli)

ABSTRACT

The business continuity plan describes the steps an organization takes when it cannot operate normally because of natural or manmade disaster. It may be written for a specific business process or may address for mission critical business processes. Business continuity and disaster recovery are critical components used to ensure that the systems essential to the operation of organization are available when needed. This is important to control the MTD (Mean Tolerable Downtime).

A distributed disaster detection system that is based on agents. Disaster Recovery Plans is a crucial part of the life of an IT center in an organization; it contains policies and procedures to be applied before, during, and after a disaster of an IT system. However, an important part of the disaster recovery process comes when the disaster occurs up until the disaster recovery plan is activated. This is precious time to detect and declare a disaster especially in critical systems.

Here, is a distributed disaster detection system that is based on agents. Agents are to be located on different servers in a data center and the will communicate with a central management unit. The Disaster recovery plan is aimed to complement the work of an existing disaster recovery system, or to run stand-alone if no disaster recovery system exists and act as a warning tool aiding the system administrator.

Cite this Article: Dhanashri D. Dhokate and B. S. Patil. Disaster Detection Plans In Distributed Systems. *International Journal of Computer Engineering and Technology*, **7**(1), 2016, pp. 94-99.
http://www.iaeme.com/IJCET/issues.asp?JType=IJCET&VType=7&IType=1

INTRODUCTION

The common perception about disasters is that it can be caused by nature (volcanoes, floods, earthquakes, tsunamis…etc.) or by human action (wars, malicious activities…. etc.). Disaster Recovery can be defined as "(DR) Planning and implementation of procedures and facilities for use when essential systems are not available for a period long enough to have a significant impact on the business". Table 1 shows the percentage of disaster causes. Smaller disasters occur more frequently; thus, they have significant impact on the systems and would cause serious downtime and outages. Therefore, one can argue that a software detection system that can prove helpful in the 95% (by summing the top 3 causes shown it Table 1) of the total disasters is wise investment and can provide a valuable addition to any data center alongside the traditional disaster recovery systems.

Table 1 The percentage of disaster causes.

Rank	Disaster Cause	Percentage
1	Hardware/Infrastructure failure	55%
2	Human error	22%
3	Software failure	18%
4	Natural disaster	5%

This is especially true given that some disaster recovery systems have no detection and warning mechanism and are applied manually. We don't suggest that Distributed Disaster Detection plan given here is to replace existing DR solution but to complement it. This system can also run without ad DR solution and thus it is a low cost system to monitor and warn against possible failure of hardware, software or human errors.

DISASTER DETECTION OVERVIEW

The fig 1. Shows Disaster Timeline, showing MTD, RTO and Pre-recovery times.

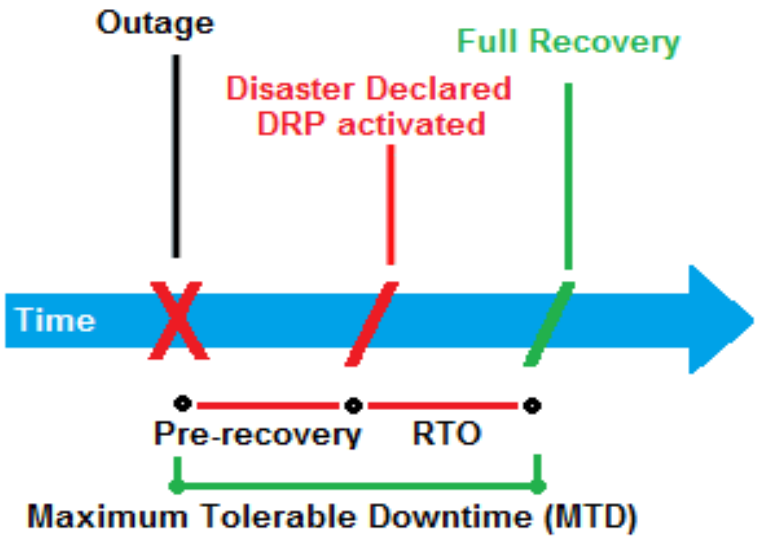

Figure 1 Disaster Timeline, showing MTD, RTO and Pre-recovery times

The disaster can be declared by:

1. **Manually**: the system administrator declares it and the recovery process is initiated
2. **Automatically**: system detects some abnormality and the absence of some services and declares a disaster and automatically starts the recovery process.
3. **System-assisted**: declaration, here the system has no privilege of declaring the disaster; however, the detection system will alert the system administrator to take action. Table 2 shows Disaster and declaration approaches.

Table 2 Disaster and declaration approaches

System	Advantages	Disadvantages
Manually	-Minimizes false negative -cheaper	Need human attention 24/7; therefore, can cause major delay
Automated	Fast response	High rate of false-positive
System-assisted	Combine both advantages	Cost of resource allocation and needs constant monitoring

One critical point as seen in Figure 1 above is the outage of some of the services which will result in a time where some data and services are disrupted; at this point the disaster is declared.

Both the advantages and disadvantages of each of the systems are previewed in Table 2 above. Besides, as with any detection system; a level of certainty is always factor in wither to declare a disaster or not; the typical four cases are shown in Table 3 below:

Table 3 Cases to decide to declare a disaster or not.

Case	Description	Result
False-negative	The system continue working	No alert
True-negative	The system declare a disaster by mistake	Alert
False-positive	The system fails to detect or uncertain to declare a disaster	No alert
True-positive	The system successfully detects a disaster and declare it	Alert

One important factor in detecting a disaster is the detection rate.

DISASTER DETECTION SYSTEM

The automated detection and warning system of disasters plays an important role in the disaster recovery. Indeed, it can improve maximum tolerable downtime (MTD) by minimizing the pre-recovery time by giving earlier signs of disaster. Thus, Disaster Detection plan will trigger the DR system if the detection returned a positive result. Fig.2 shows the Disaster detection system.

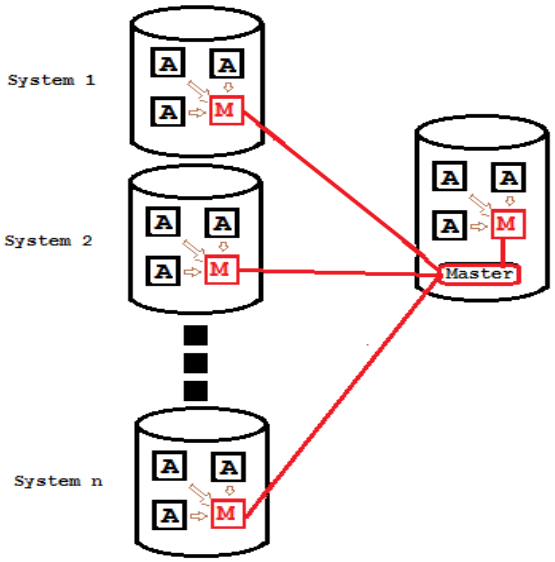

Figure 2 Diaster detection system

The disaster detection system consists of:

Agents:

An agent is basically a thread running on a system (an actual server) this thread monitors some files carefully placed in different location on the system. This thread will monitor those files for any change of contents, then, whenever a change occur will send a message to the local management system and alert of an "integrity" issue. Moreover, if the thread could not reach a certain file then it will send a different message as an alert of "availability" issue.

These threads will run in the background and should consume negligible amount of resources. The administrator should configure and set the threshold values. The agent will send a 3- tuple message containing:

- I – Agent ID
- L – Location of the issue
- S – issue "Availability" or "Integrity" or both.

Following is the algorithm of the agent and how messages are exchanged within the system with the management center.

Algorithm: Agent

S= {Availabilty_issue,Integrity_Issue,No_Issues}

Agent (I, L, S)

If! Read (location)

Return (id, loc_id, Availability_Issue);

If (location! = data)

Return (id, loc_id, Integrity_Issue);

Data =new_data;

Write (location; new_data);

Return (id, void, NO_ISSUES)

Management Centers

The management center is a thread which will be located at each system (server), they communicate with the virtual Agents and each management center will receive 2-tuple messages from the Agents and will compile a report message to be forwarded as 2-tuple message to the master management center as follows:

- S – System I.D.
- I – Type of issue detected.

Following is the basic algorithm for management and how the messages are exchanged with agents and with the master management center.

Algorithm: Management

Call the agent Agent (I, L, S)

If (S! =No_Issues)

Return (I,S)

Else call the next agent

Return (NO_ISSUES)

Master Management Center

This is the head of the system, the system receives messages from all management center and then determine wither to alert admin or administration system about a possible disaster based on a preset threshold. Hence, the administrator can have a sensitive system or a less sensitive system based on criticality of server. For example, the master management system can give more weight to critical systems and less weight to the non-essential systems.

Following is the basic algorithm for the master management center and how the messages are exchanged with management centers; and how the Disaster Detection plan passes messages to the system administrator.

Algorithm: Master_Management_Center.

Management (I, S);

Print_line (“Server:”, I,”has”, S)

A system administrator can have a dashboard to check on system issues and trigger a disaster recovery. On the other hand the master management center module given can be linked directly to existing DR system to have fully automated disaster detection and recovery system.

CONCLUSION

The system is basically based on passing messages about the status of each system and will help give an assessment about the system's health. a simple software system to give an early alert of disasters caused mostly by hardware failures, human errors and malicious attacks.

The system is completely a software system; the cost to deploy is negligible. However, the running cost is linked to the required configurations; in other words, if the system sends more status messages it will consume some bandwidth and processing resources, if the configuration is with less messages, these side-effects will also be negligible.

Given the advantage of not having to install special hardware for disaster recovery; on the other hand, it is obvious that the main limitation of the system is that if all servers got down. Thus, the system will also fail; the system is effective against security threats, limited hardware or software failures.

REFERENCES

[1] Denning, Dorothy E., An Intrusion Detection Model, Proceedings of the Seventh IEEE Symposium on Security and Privacy, May 1986, pages 119–131Fff

[2] Alghamdi, Hanaan and Alaama, Arwa, DRP-DRP: Data Replication Protocol for Disaster Recovery Planning, International Conference on Innovations in Information Technology, 2008. Pp 228-232.

[3] Ceballos, Juan DiPasquale, Richard; Feldman, Robert, Business continuity and security in datacenter interconnection, Bell Labs Technical Journal., Volume: 17 Issue: 3, 2012.

[4] http://www.latisys.com/blog-post/top-4-causes-of-it-disasters, April, 4th 2014. Jennifer Curry, Top four disaster causes,

[5] http://dictionary.reference.com/browse/disaster-recovery, October 2015.

[6] Colburn, Robert, Sound the Alarm: A History of Disaster Detection and Warning Technologies, The IEEE institute newsletter, September 9 2013.

[7] Flowers, April, NASA Builds GPS-Based System For Detecting Natural Disasters http://www.redorbit.com/news/science/1113025581/nasa-gps-system-detects-natural-disasters-121113/

[8] Y. Ren, M. Chuah, J. Yang, Y. Chen, MUTON: detecting malicious nodes in disruption-tolerant networks, IEEE WCNC 2010, April, 2010