Novel Perspectives in Construction of Recovery Oriented Computing

Manas Kumar Yogi[1] Annemneedi Lakshmanarao[2]
[1,2]Assistant Professor
[1,2]Department of Computer Science and Engineering
[1,2]Pragati Engineering College Surampalem, Andhra Pradesh, India

Abstract— In this paper we present novel views in implementation of a recovery oriented computing system. We discuss the various factors considered while ROC system design and move on to present the existing technologies in this field. Based on this we propose a ROC system design which enhances the robustness of such a system. Finally we state the future directions which the researchers are currently working in this area.

Key words: Recovery, Fault, RAID, NAS, Restartability

I. Introduction

A. Factors Influencing ROC System Design:

1) Availability:

System must provide 100% service and high quality

a) Flexibility:

System must provides flexibility

b) Accessibility:

Accessibility of System must be considered.

c) Transparency:

System must provide transparency

2) Maintainbility:

Reducing complexity while upgrading software, hardware

3) Scalabilty:

Improvement &in extension of the ROC system.

II. Current Technologies

A. Raid/Nas

RAID (Redundant Array of Inexpensive Disks, now commonly redundant array of independent disks) is a data storage virtualization technology that combines multiple physical disk drive components into a single logical unit for the purposes of data redundancy, performance improvement.RAID is very basic model for Recovery Systems.RAID comprises of several levels which supports different types of backup storage. RAID 0 consists of striping, without mirroring or parity. RAID 1 consists of data mirroring, without parity or striping. RAID 2 consists of bit-level striping with dedicated Hamming-code parity. RAID 3 consists of byte-level striping with dedicated parity.RAID 4 consists of block-level striping with dedicated parity. RAID 5 consists of block-level striping with distributed parity. RAID 6 consists of block-level striping with double distributed parity.RAID can also be nested like RAID 0+1 or RAID 1+0. Software based RAID implementations also provided by some operating systems.

Network-attached storage (NAS) is a file-level computer data storage server connected to a computer network providing data access to a heterogeneous group of clients. NAS is specialized for serving files either by its hardware, software, or configuration. NAS is designed as an easy and self-contained solution for sharing files over the network. A NAS unit is a computer connected to a network that provides only file-based data storage services to other devices on the network.

B. .Recursive Restartability:

In this approach we assume faults in software functionality like memory leaks, deadlocks hinder the normal functionality of the software. This means restarting the system is only available way of restoring the system functionality. But the restart should be methodical. It should be announced before restart as unannounced restarts are expensive. The main cost of recursive restarts are cost incurred while saving and restoring the system state. The possibility of data loss always exist whenever there is restart.This is the reason why we advocate a strong fault containment while designing recursive restart systems. The main issue lies while designing is the restart dependencies of each component in a RR system.

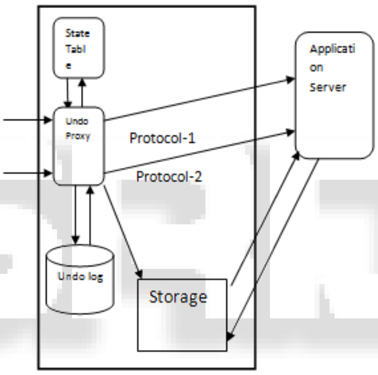


Fig. 1: Architecture of ROC

III. Proposed System

A. Independent Subsystem Design:

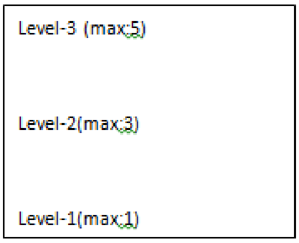
We propose a independent system design which can strengthen recovery procedure. When we design the system in form of sub components then backup defense increases thereby enhancing the reliability of the system.The cohesion of the ROC can be defined by one of components which are capable of autonomic existence.

B. Tolerance Of Unannounced Restarts:

We can model the system to allow maximum of RR per day depending on the computing nature. If its critical project then more than 1 RR should not be tolerated. This tolerance level is important for leveraging the performance of the system. We introduce a measure of counts of RR at each user level .If this count is restricted to a reasonable value then recovery cost will be reduced and time to stabilize the environment of the computing system will also be less.

C. Enhanced Recursive Restart Methodology:

As shown in the below figure users are maintained at 3 levels in this methodology. The users have the facility of recursively restarting a system at level i.e. the lowest level only one time but at level 2 this is possible for maximum of 3 times. Most users operate at this level so 3 times is the optimal count for this level. Level 3 operates at highest level ie, at administrator level so 5 times restart is allowed to tolerate different types of system failures. But practically the admin restores the system with at most two restarts.



Maximum users are designed to give access at level-2
max: maximum number of restarts allowed

IV. FUTURE DIRECTIONS:

IV.FUTURE DIRECTIONS:

1) Reducing the percentage of failure of operators by using new design principles of Recovery Oriented System.
2) Support for live recovering experiments in Operating System.
3) Creation of metrics to measure the cost of downtime & ownership.
4) Development of robust Virtual Machines for implementing/facilitating Recovery Oriented System methodology successfully.

V. CONCLUSION

Recovery oriented system are a aim towards building self diagnostic systems and self tolerant systems when failure occurs in the system. In this paper we have presented measures towards building such a smarter system which can recover by means of efficient strategies. As computing needs of human beings are increasing day by day the need for implementing ROC systems is not only a need but a achievement by the researchers working in this direction. We earnestly hope this paper acts a catalyst for design efficient ROC systems in future.

REFERENCES

[1] A. Brown, G. Kar, and A. Keller. An Active Approach to Characterizing Dynamic Dependencies for Problem.Determination in a Distributed Environment. To appear in Proceedings of the Seventh IFIP/IEEE International Symposium on Integrated Network Management (IM VII), Seattle, WA, May 2001.

[2] G. Candea and A. Fox. Recursive Restartability: Turning the Reboot Sledgehammer into a Scalpel. Submission to the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII).

[3] J. M. Christensen and J. M. Howard. Field Experience in Maintenance. Human Detection and Diagnosis of SystemFailures: Proceedings of the NATO Symposium on Human Detection and Diagnosis of System Failures, J.Rasmussen and W. Rouse (Eds.). New York: Plenum Press, 1981, 111–133.

[4] S. Fisher. E-business redefines infrastructure needs. InfoWorld, 7 January 2000. Available from www.inforworld.com.

[5] D. R. Kuhn. Sources of Failure in the Public Switched Telephone Network. IEEE Computer 30(4), April 1997.

[6] J. Menn. Prevention of Online Crashes is No Easy Fix. Los Angeles Times, 2 December 1999, C-1.

[7] B. Murphy and T. Gent. Measuring System and Software Reliability using an Automated Data Collection Process.Quality and Reliability Engineering International, 11:341–353, 1995.

[8] C. Perrow. Normal Accidents: Living with High-Risk Technologies. Princeton, NJ: Princeton Press, 1999.

[9] R. H. Pope. Human Performance: What Improvement from Human Reliability Assessment. Reliability Data Collection and Use in Risk and Availability Assessment: Proceedings of the 5th EureData Conference, H.-J.Wingender (Ed.). Berlin: Springer-Verlag, April 1986, 455–465.