**The Design of a Practical System for Fault-Tolerant Machines**

**Summary:**

The main focus of this paper to build a fault tolerant enterprise grade system for virtual machines running on servers in cluster. This is achieved by created a backup virtual machine on another server for the primary virtual machine. System is designed in VMware vSphere 4.0. Servers are model as deterministic state machines. Fault tolerant servers are where a backup server is available to take over if the primary server fails and backup server have nearly identical state as that of primary server. Servers are on shared storage.

All changes of primary servers are transferred to backup server and backup serve executes exactly through log files. Only primary virtual machine is on network and all inputs come to primary virtual machine and then transferred to backup server through logging channel.

Failures in this system is detected using combination of heart beating messages between relevant servers and traffic is monitored on the logging channel between relevant channels and either of the primary or backup virtual machine takes over the execution. Challenges for replicating execution is correctly capturing all the input and non-determinism and applying these inputs and non-determinism without degrading performance.

Deterministic replay is used to produce log entries to record the execution of the primary

VM and then send them to the backup VM via the logging channel. The backup VM run all the log instruction and output of backup VM is discarded. Primary VM may not send an output to the external world, until the backup VM has received and acknowledged the log entry associated with the operation producing the output. Backup VM will have exactly the same state of primary VM if it has received all the log entries.

Starting a VM machine in same state as that of primary VM is provided. Traffic on logging channel is monitored and maintained. Control operations applied on primary VM are also applied on backup VM like if the primary VM is explicitly powered off, then backup VM should be stopped as well etc. For control operations special control entries are sent on the logging channel from the primary to the backup, in order to effect the appropriate operation on the backup. Issues related to disk IO and network operations and their solution are discussed.

The author have discussed design alternative like shared verses non-shared disk space in order to handle the failover situation. An alternate design is to have the backup VM execute disk reads and therefore eliminate the logging of disk read data. This greatly reduce the traffic on the logging channel for workloads that do a lot of disk reads however it may slow down the backup VM’s execution. Performance VMware FT for multiple applications is analyzed in detail. This system typically reduces performance of real applications by less than 10% and the data bandwidth needed to keep the primary and backup VM executing in lockstep is less than 20 Mbit/s for several real applications.

This paper provide a brief information about the structure and working of system. Author have also provided alternative scenarios for most common faced issues and have done detailed performance analysis of system with practical implementation on multiple machines however I have few questions that needed further clarification.

**Questions:**

* How system will handle the situation if both primary and backup server fails at same time?
* Having multiple copies of backup server and keeping all sync, isn’t is too much costly?
* How the system handles the lost instructions?
* What are alternative system for handling fault tolerance other than Virtual Machine based structure?