50116 Arsalan Qaiser

A High Availability Clustering Protocol

Abstract

Clustering is a technique in which physical connections and software programs link two or more servers or nodes. When a failure occurs on one node, its workload can failover to the surviving node. The goal of the cluster is to establish a high-availability environment that minimizes application downtime. Using high-availability server clustering in storage environment gives some special characteristics such as data integrity and file clustering.

Currently the main focus for preserving data integrity and consistency in cluster architecture is the use of power switches. This device enables cluster system to power-cycle the other cluster system before restarting its services during failover.

This research proposes a protocol that avoids the threat of data corruption. Its main advantage is that it does not require forcible removal of power from server.

This inclusion would solve many of the problems associated with power switches techniques such as a forced cut-off corrupts any data resident in non-shared directly attached storage, under certain scenarios (System Hang) interrupts are blocked, the kernel thread not called which causes data corruption, it halts any application running on server even though that might not requires access to shared storage.

My proposed protocol work on hot-standby configuration in which a primary cluster system run all services and a backup cluster system takes over only if the primary system fails and support all generic services.

Data integrity and availability is assured by preventing a node of a distributed, clustered system from accessing shared data in the case of a failure of a node or communication links with the node.

While this new approach offers significantly better flexibility and performance at lower Costs and it does not raise the new set of technical challenges. This approach provides highly available access to data while maintaining data integrity and maximum scalability.

The proposed solution is analyzed on a real time environment, the different IP addresses are assigned to primary and secondary server and on different scenarios the ICMP packets are sent towards primary server. The result shows that when the primary server gets responsive and changes its parity bit, then the proposed protocol stops the pre defined services on secondary server and starts the services on primary server.