Database fault tolerance

Technology

ASSIGN BUSTER

Fault-tolerance or graceful degradation is the property that enables a system to continue operating properly in the event of the failure of some of its components (Wikipedia). If its operating quality decreases at all, the decrease is proportional to the severity of the failure, as compared to a naively-designed system in which even a small failure can cause total breakdown. Fault-tolerance is particularly sought-after in high-availability or life-critical systems. A lockstep fault-tolerant machine uses replicated elements operating in parallel. At any time, all the replications of each element should be in the same state.

The same inputs are provided to each replication, and the same outputs are expected. The outputs of the replications are compared using a voting circuit. A machine with two replications of each element is termed dual modular redundant (DMR). The voting circuit can then only detect a mismatch and recovery relies on other methods. A machine with three replications of each element is termed triple modular redundant (TMR). The voting circuit can determine which replication is in error when a two-to-one vote is observed. In this case, the voting circuit can output the correct result, and discard the erroneous version.

After this, the internal state of the erroneous replication is assumed to be different from that of the other two, and the voting circuit can switch to a DMR mode. This model can be applied to any larger number of replications. Fault Tolerance is the ability of the system to cope with and to recover from various faults.

The faults that are covered:

• Fatal Server Process Fault: A server process fault is usually caused by a serious programming error, which causes the Operating System to terminate the process.

• Server Hardware Fault: A server hardware fault causes the complete machine unexpectedly to shutdown. All running processes will be lost as well. Key Vendors

• K2 Component Server (K2 Fault Tolerant Architecture) - The K2 Component Server supports detection of the above mention faults and implements support for recovery. However a Component programmer is still requested to use the offered functionality in case he wants to implement a complete recoverable application. That means a recovery from a severe fault must also be considered at the application design and implementation level.

• Versant Fault Tolerant Server (FTS) - The Versant Fault Tolerant Server (FTS) is an add-on software module for the Versant Object Database enabling automatic fail-over and recovery in the case of hardware or software failure. FTS uses synchronous replication between two database instances and supports transparent re-synchronization in the event of a failure. Synchronous database replication mirrors the contents of one database to another in a predictable and orderly manner. Versant's FTS is driven through configuration by the application process to support in-flight transaction recovery on failure. This approach offer a distinct advantage over the log based replication commonly found in database products.

• Real-Time Mantra - Fault Handling and Fault Tolerance

Architecture for fault tolerance in a database management system is based upon the concepts of careful replacement and differential files on multiple media with backup copies available. An algorithm for transaction execution that preserves the highest degree of consistency is presented along with an algorithm for the reorganization of the database. The reorganization algorithm merges the differential file with the original database in an on-line fashion.

Thus the database is available continuously, eliminating one of the drawbacks of differential file processing. A detailed simulation model has been implemented as a step toward the verification of the determinacy of the algorithms. In addition, certain performance aspects of the system have been analyzed statistically in order to estimate the overhead in time and space due to the redundancy in the system. After a failed node has been brought back into the cluster and its instance has been started, RAC's Cluster Ready Services (CRS) automatically manages the virtual IP address used for the node and the services supported by that instance automatically.

A particular service may or may not be started for the restored instance. The decision by CRS to start a service on the restored instance depends on how the service is configured and whether the proper number of instances is currently providing access for the service. A service is not relocated back to a preferred instance if the service is still being provided by an available instance to which it was moved by CRS when the initial failure occurred. CRS restarts services on the restored instance if the number of instances that are providing access to a service across the cluster is less than the number of preferred instances defined for the service. After CRS restarts a service on a restored instance, CRS notifies registered applications of the service change.

For example, suppose the HR service is defined with instances A and B as preferred and instances C and D as available in case of a failure. If instance B fails and CRS starts up the HR service on C automatically, then when instance B is restarted, the HR service remains at instance C. CRS does not automatically relocate a service back to a preferred instance. Suppose a different scenario in which the HR service is defined with instances A, B, C, and D as preferred and no instances defined as available, spreading the service across all nodes in the cluster.

If instance B fails, then the HR service remains available on the remaining three nodes. CRS automatically starts the HR service on instance B when it rejoins the cluster because it is running on fewer instances than configured. CRS notifies the applications that the HR service is again available on instance B. Ensuring that application services fail over quickly and automatically within a RAC cluster, or between primary and secondary sites, is important when planning for both scheduled and unscheduled outages. It is also important to understand the steps and processes for restoring failed instances or nodes within a RAC cluster or databases between sites, to ensure that the environment is restored to full fault tolerance after any errors or issues are corrected.