A High Availability and Disaster Recovery System

Qin Zhang
The department of Computer Science
Chengdu University of Information Technology
Chengdu, China
acazhang@gmail.com

Hong Xu
The department of Computer Science
Chengdu University of Information Technology
Chengdu, China
xuhong@cuit.edu.cn

***Abstract*—In this paper, we present the design and implementation of cluster structure and disaster recovered system that can be used in the bank, industry and enterprise. It is capable of storing great capacity data. This toolkit is targeted at data security business which want to achieve high availability and disaster recovered structure and to compare their structure with others. The comparison functions can be used to find structure differences between two performance level, especially in data safeguard. The system is developed in Unix and the great capacity data is stored in a relational database.**

***Keywords*—high availability, recovery disaster, cluster, performance comparison**

I. INTRODUCTION

A cluster consists of two or more independent, but interconnected, servers. Several hardware vendors provided cluster capability over the years to meet a variety of needs.Some clusters were only intended to provided high availability by allowing work to be transferred to a secondary node if the active node fails. Others were designed to provide scalability by allowing user connections or work to be distributed across the nodes.

Another common feature of a cluster is that it should appear to an application as if it were a single server. Similarly, management of several servers should be as similar to the management of a single server as possible. The cluster management software provides this transparency. [1]

For the nodes to act as if they were a single server,files must be stored in such a way that they can be found by the specific node that needs them.There are several different cluster topologies that address the data access issue,each dependent on the primary goals of the cluster designer. The interconnect is physical network used as a means of communication between each node of the cluster.

Our designed system is being tested with data from a production environment. The system provides a safe disaster recovery for the client. The structures of the system prevents quantitative data from being lost.

The rest of the paper is organized as follows: Section 2 presents an overview of the system. Our database structure and disaster recovery features of the system are described in Section 3. Section 4 discusses some salient features of the system. Conclusion will be presented in Section 5.

II. OVERVIEW OF RAC AND DISASTER RECOVERY SYSTEM

Real Application Clusters (RAC) enables high utilization of a cluster of standard, low-cost modular servers such as blades. RAC offers automatic workload management for services. Services are groups or classifications of applications that comprise business components corresponding to applications and provide support for multiple services on multiple services on multiple services on multiple instances. If a primary instance fails, the system moves the services from the failed instance to a surviving alternate instance. Oracle also load balances connections across instances hosting a service. RAC, which is based on a shared-disk architecture, can grow and shrink on demand without the need to artificially partition data among the servers of the cluster. RAC also offers a single-button addition and removal of servers to a cluster. Thus, we can easily provide or remove a server to or from the database.

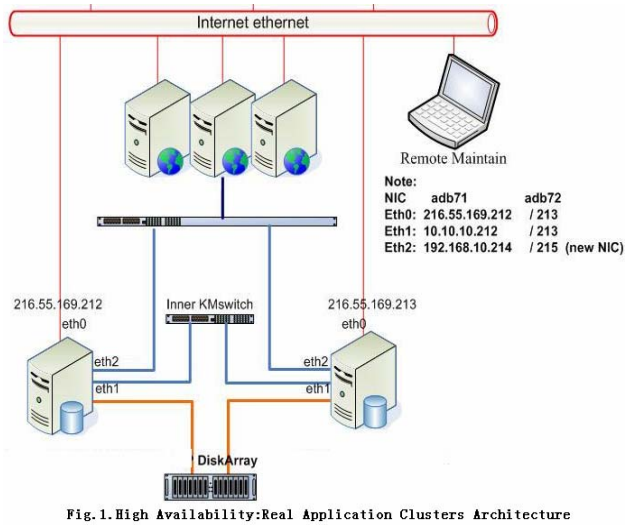

Fig.1.1. Real Application Clusters Architecture

Data Guard ensures data protection, and disaster recovery for enterprise data. Data Guard provides a comprehensive set of services that create, maintain, manage, and monitor one or more standby databases to enable production databases to survive disasters and data corruptions. Data Guard maintains these standby databases as transactionally consistent copies of the production database. Then, if the production database becomes unavailable because of a planned or an unplanned outage, Data Guard can switch any standby database to the production role, minimizing the downtime associated with the

outage. Data Guard can be used with traditional backup, restoration, and cluster techniques to provide a high level of data protection and data availability. With Data Guard, administrators can optionally improve production database performance by offloading resource-intensive backup and reporting operations to standby systems.

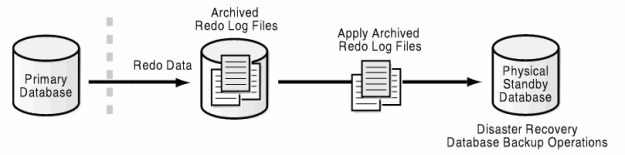

Fig.1.2. Data Guard Model

For our system, storage is a critical component of any grid solution. Traditionally, storage has been directly attached to each individual server(DAS). Over the past few years, more flexible storage, which is accessible over storage area networks or regular Ethernet networks, has become popular. These new storage options enable multiple servers to access the set of disks, simplifying provisioning of storage in any distributed environment. [3] AS we already saw, the choice of file system is critical for RAC deployment. Traditional file systems do not support simultaneous mounting by more than one system, or on a file system. Therefore, you must store files in either raw volumes without any file system, or on a file system that supports concurrent access by multiple systems. Thus, three major approaches exist for providing the shared storage needed by RAC:

- Raw volumes: These are directly attached raw devices that require storage that operates in block mode such as fiber channel or iSCSI.
- Cluster File System: One or more cluster file systems can be used to hold all RAC files. Cluster file systems require block mode storage such as fiber channel or iSCSI.
- Automatic Storage Management(ASM) is a portable, dedicated, and optimized cluster file system for database files.

III. THE SYSTEM STRUCTURE AND WORK MECHANISM

RAC(Real Application Clusters) is the successor to Oracle Parallel Server (OPS).RAC allows multiple instances to access the same database (storage) simultaneously. RAC provides fault tolerance, load balancing, and performance benefits by allowing the system to scale out, and at the same time since all nodes access the same database, the failure of one instance will not cause the loss of access to the database.

At the heart of RAC is a shared disk subsystem. All nodes in the cluster must be able to access all of the data, redo log files, control files and parameter files for all nodes in the cluster. The data disks must be globally available in order to allow all nodes to access the database. Each node has its own redo log files and UNDO tablespace, but the other nodes must be able to access them (and the shared control file) in order to recover that node in the event of a system failure.

The biggest difference between RAC and OPS is the addition of Cache Fusion. With OPS a request for data from one node to another required the data to be written to disk first, then the requesting node can read that data. With cache fusion, data is passed along a high-speed interconnect using a sophisticated locking algorithm. [2]

Not all clustering solutions use shared storage. Some vendors use an approach known as a Federated Cluster, in which data is spread across several machines rather than shared by all. With RAC, however, multiple nodes use the same set of disks for storing data. With RAC, the data files, redo log files, control files, and archived log files reside on shared storage on raw-disk devices, a NAS, ASM, or on a clustered file system. The database approach to clustering leverages the collective processing power of all the nodes in the cluster and at the same time provides failover security.

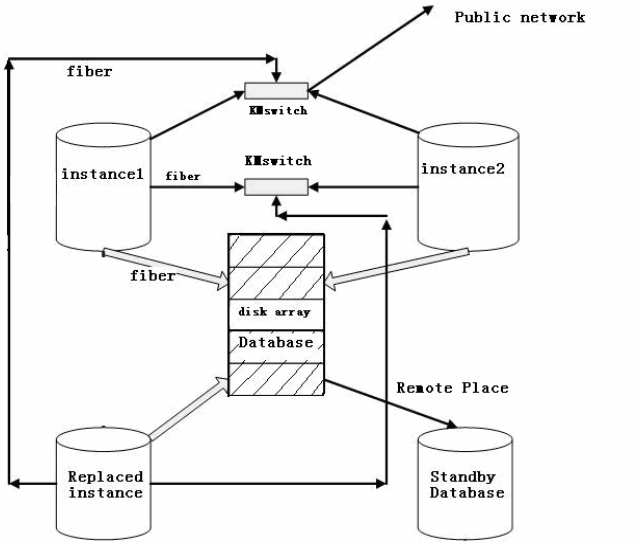

Fig.3.1. High Availability and Disaster Recovery System

When an instance fails and the failure is detected by another instance, the second instance performs the following recovery steps:

A. During the first phase of recovery, Global Enqueue Service(GES) remasters the enqueues.

B. Then the Global Cache Service (GCS) remasters its resources. The GCS processes remaster only those resources that lose their masters. During this time, all GCS resource requests and write requests are temporarily suspended. However, transactions can continue to modify data blocks as long as these transactions have already acquired the necessary resources.

C. After enqueues are reconfigured, one of the surviving instances can grab the Instance Recovery enqueue. Therefore, at the same time as GCS resources are remastered, SMON determines the set of blocks that need recovery. This set is called the recovery set. Because, with Cache Fusion, an instance ships the contents of its blocks to the requesting instance

without writing the blocks to the disk, the one-disk version of the blocks may not contain the changes that are made by either instance. This implies that SMON needs to merge the content of all the online redo logs of each failed instance to determine the recovery set. This is because one failed thread might contain a hole in the redo that needs to be applied to a particular block. So, redo threads of failed instances cannot be applied serially. Also, redo threads of surviving instances are not corresponding buffer caches.

D. Buffer space for recovery is allocated and the resources that were identified in the previous reading of the redo logs are claimed as recovery resources. This is done to avoid other instances to access those resources.

E. All resources required for subsequent processing have been acquired and the (Global Resource Directory)GRD is now unfrozen. Any data blocks that are not in recovery can now be accessed. Note that the system is already partially available. [5]

Then, assuming that there are past images or current images of blocks to be recovered in other caches in the cluster database, the most recent is the starting point of recovery for these particular blocks. If neither the past image buffers nor the current buffer for a data block is in any of the surviving instances' caches, then SMON performs a log merge of the failed instances. SMON recovers and writes each block identified in step C, releasing the recovery resources immediately after block recovery so that more blocks become available as recovery proceeds.

After all blocks have been recovered database or the recovered resources have been released, the system is again fully available.

In summary, the recovered database or the recovered portions of the database becomes available earlier, and before the completion of the entire recovery sequence. This makes the system available sooner and it makes recovery more scalable.

IV. SALIENT FEATURES OF THE SYSTEM

In this section, we describe some salient features of our system including replaced instance and data management, disaster recovery and extensibility.

In a Real Application Clusters environment, any standby instance can receive redo data from the primary database; this is a receiving instance. However, the archived redo log files must ultimately reside on disk devices accessible by the recovery instance. Transferring the standby database archived redo log files from the receiving instance to the recovery instance is achieved using the cross-instance archival operation.

The standby database cross-instance archival operation requires use of standby redo log files as the temporary repository of primary database archived redo log files. Using standby redo log files not only improves standby database performance and reliability, but also allows the cross-instance archival operation to be performed on clusters that do not have a cluster file system. However, because standby redo log files are required for the cross-instance archival operation, the primary database can use either the log writer process (LGWR) or archive processes (ARCn) to perform the archival operations on the primary database.

When both the primary and standby databases are in a Real Application Clusters configuration, then a single instance of the standby database applies all sets of log files transmitted by the primary instances. In this case, the standby instances that are not applying redo data cannot be in read-only mode while Redo Apply is in progress.

A.. Replaced Instance

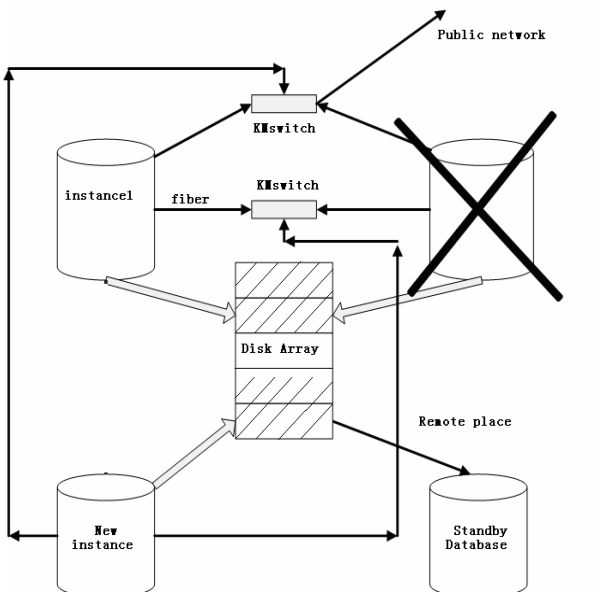

Fig.4.1. When an Server failed

How to manage and utilize different instance is much more difficult than the public database. In our system, when the instance 2 fails, we can use the new instance (replaced instance), in this way, the workload will be the same as the former performance. Importantly, we separate the different instance from public database, and apply the same schema but different constraints to them.

B. Data Management

Data Management is one of the most important considerations in any database. It is more important in our system because Real Application Clusters will contain some proprietary data that users normally do not want to share with other people or groups. Keeping the privacy of data is one of the major goals of our system. First, the whole system is designed as a standalone system that will be used only in the laboratory, to which there is no free access. Secondly, data are stored separately from the publicly data, and have an ownership property that allows only the owners to change their values and status. Access to the proprietary data is also under the control of the owners. Therefore, the proprietary data can be managed in a safe and flexible fashion in RAC.

C. Disaster Recover and Performance Testing

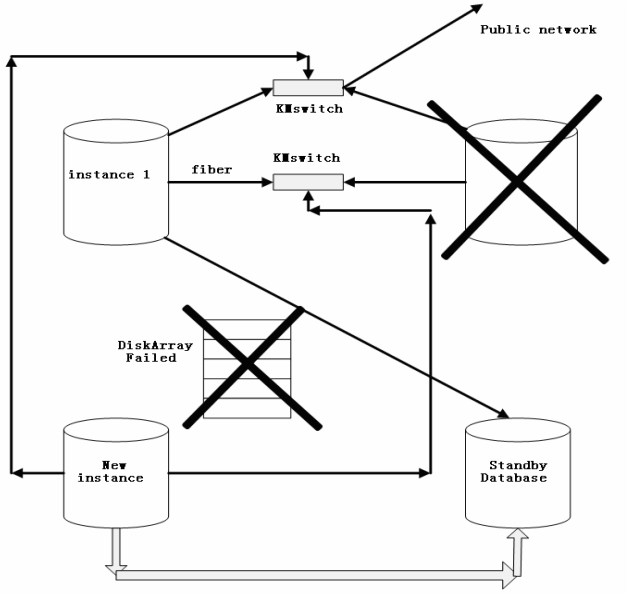

Fig.4.2. RAC and Data guard Mode

If the storage for textile machinery fired, especially, the diskarray which the database is stored will be failed. The remote standby database will play the role of diskarray and the system will not lose any data. Because the standby database receives archive log files from the diskarray everyday. Furthermore, if the instance 2 is failed, the new instance will replace it.

```
SQL>ALTER DATABASE COMMIT    TO
          SWITCHOVER TO STANDBY;
SQL>SELECT INSTANCE_NAME , HOST_NAME
          FROM V$INSTANCE
WHERE INST_ID <>
(SELECT    INSTANCE_NUMBER
FROM V$INSTANCE);
INSTANCE_NAME       HOST_NAME
------------------------    ---------------------
INST2                          standby2
SQL>CONNECT SYS/SYS@standby2 AS SYSDBA
CONNECT;
```

This provides for high availability and disaster recovery principles, and protects the bank, industry and enterprise production data.

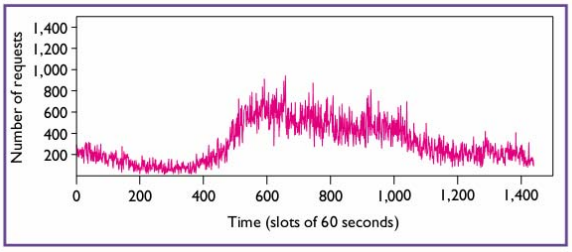

Fig.4.3. Arrival process of instance requests to an online transaction in one day. [4]

V. CONCLUSION

In this paper, we present an new high availability and disaster recovery architecture system. The problem is proved to protect data hard problem in general. We provide a safeguard data platform and can be high performance to work for database. The system is divided into two phases. The first phase is the cluster of many instances as the destination node. In the second phase, we detailed the data guard for the recovery database. Our cluster and disaster recovery structure is suitable for the great capacity data storage, management. In the future, we will develop a monitor tool to forecast disaster, and integrate more functions and tools to our system according to the practical requirements.

REFERENCES

[1] D.A. Menascé, R. Dodge, and D. Barbará, "Preserving QoSofE Commerce Sites through Self-Tuning: A Performance Model Approach," Proc. 2001 ACM Conf. E-Commerce, ACM Press, 2001, pp. 224–234.

[2] Agrawal R, Gehrke J, Gunopulos D. Raghavan P. Automatic SubSpace Clustering of High Dimensional Data for Data Mining Applications. In: Proceedings of the ACM SIGMOD International Conference on Knowledge Discovery in Database and Data Mining, Montreal, Canada, 1998, 94-105

[3] Rockart, John Fand Dvid W.De Long. Executive Support System. Dow Jones-Irwin, Homewood. IQ(1998)

[4] D.A. Menascé, "Load Testing of Web Sites," IEEE Internet Computing, vol. 6, no. 4, July/August 2002, pp. 70–74

[5] Rex Black, Managing the Testing Process, MA: Addison-Wesley, 2002:165-178.