Fault Tolerant Design in the Earth Observing System Archive

Alla Lake, Jonathan Crawford, Raymond Simanowith, Bradley Koenig
Lockheed Martin
1616 McCormick Drive, Upper Marlboro, MD 20774
alake@eos.hitc.com
jcrawfor@eos.hitc.com
rsimanow@eos.hitc.com
bkoenig@eos.hitc.com
tel: +1-301-925-0626
fax: +1-301-925-0651

Abstract

The Archive for the Earth Observing System (EOS) is one of the largest and highest data rate archives in the world. The EOS Archive is referred to as EOS Core System (ECS) and is a multi-site distributed data warehouse of Earth-oriented satellite images and science algorithms/reports. Its data holdings are projected to approach five petabytes by 2002. Each distributed site is referred to as a "Distributed Active Archive Center" or DAAC. The DAAC sites are being incrementally delivered with final deployment by the end of 2000. One of the sites, the EROS Data Center (EDC) in South Dakota, is receiving and archiving Landsat data in addition to the data generated by the instruments on the Terra satellite launched in December of 1999. Four of the DAACs will begin receiving Terra data in early 2000 [1].

The ECS archive architecture is based on a multi-site, distributed, client-server model. Its components are interdependent. As in any large and reasonably complex system robustness and ability of functional components to recover from faults is of great importance. In particular, ECS places heavy emphasis on data integrity and data capture robustness. This paper briefly describes the design of the hardware and software to insure the EOS data is captured and distributed in spite of faults. The description of hardware failover is confined to the Ingest Component design. The paper is intended as an introduction to the Poster Presentation material, and other components are discussed in the Poster Presentation itself.

1 Introduction

The overall fault recovery scheme in the ECS archive is designed to be a combination of the hardware server failover and software server recovery. The hardware server failover is operator initiated. It takes place in the event of a catastrophic failure of the hardware server itself or its associated network interface. Hardware failover to a secondary server can also be initiated as a planned maintenance or upgrade step. Failover is followed by the software server recovery for the ECS archive to continue operation. A software server recovery can also take place when the software fails, independently of any hardware faults. The hardware and software recovery designs are functionally independendent and are treated separately in this paper.

The hardware portion of a fault recovery design differs on a hardware subsystem by hardware subsystem basis using several different configurations, as appropriate, to meet the EOS mission objectives. In most cases fault handling requirements range from 2 hours to 24 hours depending on the subsystem. In practice, the down time must be minimized because of the impact on both the user community and processing of the data. The goal of the fault tolerant design is to reduce the down time to at most 15 –30 minutes per incident.

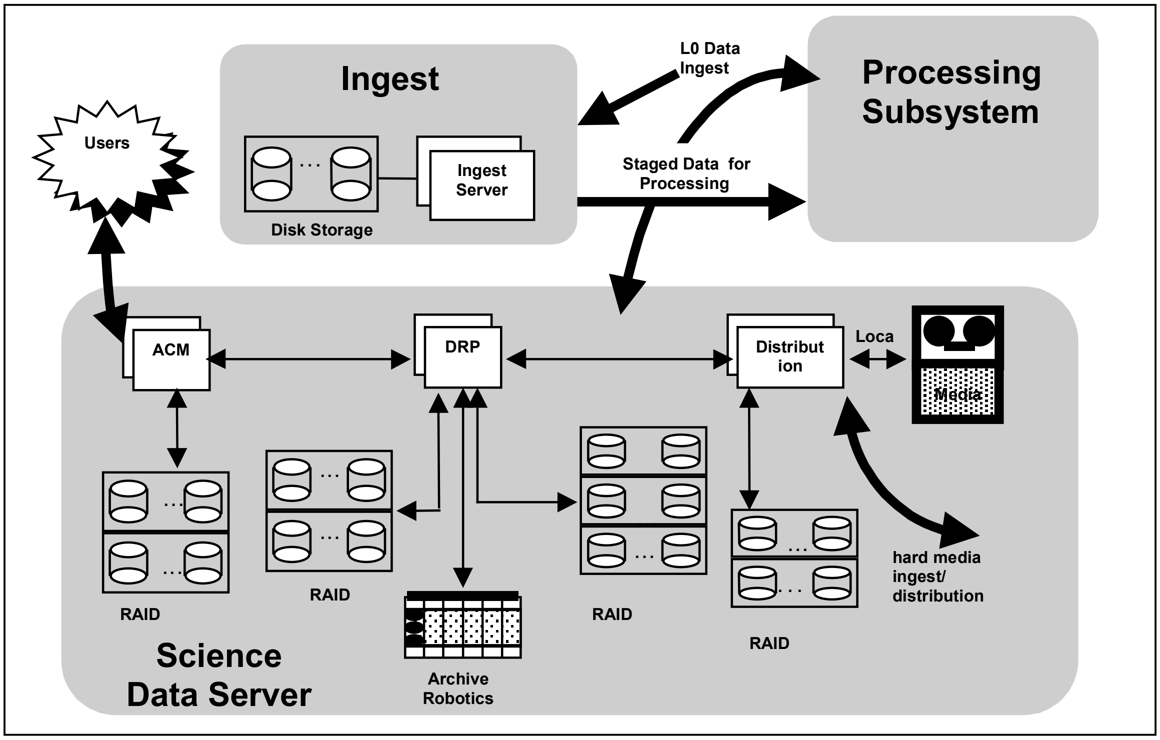

Figure 1. Hardware Configuration of the ECS Archive

Functionally, all external user electronic access to the system takes place via the Access Control Management (ACM) platforms, as illustrated in Figure 1, Hardware Configuration of the ECS Archive. The ACM is also where the ECS Data Server Metadata catalogue resides. In the ECS system a metadata catalogue indexes the total collection and points to files stored in silo-based archives.

The INGEST subsystem is responsible for data capture. Both the ACM and INGEST subsystems have the most stringent fault recovery requirements of 2 hours and a design goal of 15 minutes. Ingest hosts are used for Level 0 Instrument Data capture from the Front-End data capture facilities into the archive. A warm-standby pair configuration is used for the ACM and Ingest hosts. In this scheme failover to the secondary server is an operator-initiated event.

The Archive (DRP) hosts function as file servers connecting the rest of the system to the Nearline data holdings in the robotic silos. The hardware configuration of this component and aspects of its performance have been discussed at the March 1998 Sixth NASA Mass Storage Systems and Technologies Conference [2]. The DRP subsystem recovery requirement is 3 hours. A cluster "many-to-one" failover configuration is used for the DRP hosts. Once again, failover to the standby host is initiated manually. Once initiated, a portion of the process takes place automatically via execution of a series of scripts. Several steps within the failover procedure are manual, primarily the network router switchover. The ACM Data Base servers, the Ingest hosts and the DRP hosts platforms are at this time of Silicon Graphics Incorporated (SGI) Challenge[1] class servers. SGI Origin class servers will replace these as part of technology evolution during the life of the archive. The custom software for the Science Data Server in the ACM Hardware component resides on SUN platforms.

The Distribution component of the archive is responsible for the distribution of hard media to the users of ECS. The Distribution component has a recovery requirement of 2 hours. A load sharing configuration, allowing graceful throughput degradation in the event of failure, is designed for the Distribution hosts. The Distribution hosts are SUN Ultra servers.

More detailed descriptions of the software functions of the above system components can be found in other papers presented at this conference [1], [3].

2. Hardware Server Failover

Ingest is the only component which has been configured for hardware failover and tested in that configuration. All other components in the design have not been tested for the SGI Challenge class servers. The Ingest component consists of a pair of SGI Challenge servers. One of the servers is normally playing a primary role and the second one a secondary. Figure 1, Ingest Failover Pair, illustrates the hardware configuration.

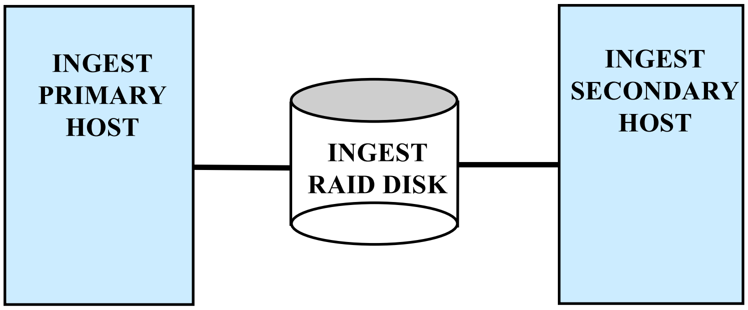

Figure 1. Ingest Failover Pair

Both hosts are physically connected to the Redundant Array of Independent Disk (RAID) in a "Dual-Bus/Dual-Initiator Configuration". Only one of the hosts is actively addressing the RAID at any one time. For simplifying the operations, the same host is always considered to be a "normally primary host". That is, in operation, that host is in the primary configuration at all times and is performing ECS ingest functions, except for the brief time periods for repair or upgrades. The secondary host may be used at the same time for other tasks or testing with two significant restrictions: 1) ECS functional configuration, both custom and Commercial Off The Shelf (COTS) must remain intact and in sync with the primary host, and 2) dual connected RAID is not available for use by the secondary host. Any attempt to address (read or write) dual connected RAID from the secondary machine may result in disk corruption. Switching control of the RAID from the normally primary to the normally secondary host is done through a failover procedure. Failback procedure is exercised when the RAID control is switched back from the normally secondary to the normally primary host. Both failover and failback involve 1) switching of ownership of RAID, 2) manual switching of all external network mounts and interfaces. Network switchover uses an *alias ip* mechanism. At the EDC DAAC, the only DAAC that currently uses HiPPI connection with Ingest, the HiPPI connection is also switched.

For the implementation of the Ingest Failover scheme, aside from the dual physical connection of the RAID, a number of specific changes must be made to the host system, network, and peripheral device configuration. Both the primary and the secondary hosts have an identical hardware complement, identically prepared RAID configuration, and their internal disks are loaded identically with the same complement of ECS COTS code.

3. Software Server Fault Recovery

The software for ECS is a C++ implementation using Distributed Computing Environment (DCE) [4] for process communications. The fault recovery software design relies on a combination of custom code supported with a relational database for request persistence and checkpointing, as well as COTS product features to allow network rebinding. Since all client-server interfaces are implemented using DCE RPC calls, it is crucial that lengthy processing operations not be repeated needlessly. At the first layer of software fault recovery, DCE rebinding is incorporated into the client interface classes. Rebinding permits automated detection and recovery of errors in DCE communications, including those introduced by network disruption. The second layer of fault recovery insulates against both client and server process failure ("crashes"). Long-running requests are checkpointed to the database, with all parameters and temporary data needed during processing. This checkpointing also provides a built-in queuing mechanism.

In the event of a client crash, the server, depending on the client type, takes one of the two possible actions. It abandons processing of the client's outstanding requests in favor of processing of requests from other client processes. Or, alternatively, it continues processing the client request and waits for the client to resynchronize to complete the transaction. In the event of a server crash, the client will attempt to rebind until the server is restarted, or until the client determines that an unacceptable period of time has elapsed. Resubmitted requests, whether through automatic rebinding or operator resubmission, are resumed from the last checkpointed state, thus eliminating redundant re-processing.

Upon restart, processes send a notification to the servers to which they are clients so that orphaned resources may be reclaimed.

Additional software fault recovery features provide for multiple server start "temperatures." Normal "warm start" processing permits resumption of request processing from the last checkpointed state upon client resubmission. "Cold start" mode terminates any in-progress requests and resets the persistence table to reflect an empty request queue. Resubmitted requests appear to the server as new requests. "Cold restart" mode provides a mechanism for "back-flushing" requests. Requests in progress are set to a failed state, and resubmission returns failure to the calling client.

4. Conclusion

Design of the failure recovery mechanisms in the ECS archive is an ongoing technical process as the system evolves following the computing technology trends. As an example, at the time of this writing, a replacement design for hardware failure recovery is being considered for implementation with the SGI's current generation Origin servers.

5. Acknowledgments

The authors thank Robert Howard of Raytheon Systems Corporation for the review of the text.

6. References

[1] Behnke, J., Lake, A., "EOSDIS: Archive and Distribution Systems in the Year 2000", Eighth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies, College Park, Maryland, March 2000

[2] Lake, A., "Performance Tuning of a High Capacity/High Performance Archive for the Earth Observing Systems Project", Sixth Goddard Conference on Mass Storage Systems and Technologies, College Park, Maryland, March 1998

[3] Crawford, J.M., "A Scalable Architecture for Maximizing Concurrency", Eighth NASA Goddard Space Flight Center Conference on Mass Storage Systems and Technologies, College Park, Maryland, March 2000

[4] OSF DCE User's Guide and Reference

[1] Here and subsequently, Silicon Graphics, Challenge and Origin are registered trademarks of Silicon Graphics, Inc., ULTRA is a trademark of SUN Microsystems, OSF is the trademark of the Open Software Foundation, Inc.