Designing Fault-Tolerant Distributed Archives for Picture Archiving and Communication Systems

Rebecca Mendenhall, Matt Dewey, and Ian Soutar

Purpose: Distributed archives in a picture archiving and communication system (PACS) environment can provide added fault tolerance and fail-over capability, as well as increased load capacity at a more economical price than traditional "high-availability" systems. Systems can be configured with varying levels of fault tolerance, depending on the amount of redundancy desired. There is, however, a direct correlation between the level of hardware redundancy and cost to implement. This presentation details the system design for fault-tolerant distributed archives as well as several options for redundancy, referencing implementation of a fault-tolerant archive system at the University of Utah. **Methods:** The distributed archive system described here is based on Image Devices' image archive software, which can be implemented on multiple individual archive servers in order to distribute archive functionality and operational load. The configuration and implementation of the individual servers together make up the distributed archive system and does not impact the ability of the system to be scaled to meet future requirements. Several implementation and configuration options exist, including the ability for servers to maintain replicated databases containing patient and image information. Thus, each archive can be aware of all information and the location of this information within the distributed archive system. **Results:** The goal is to produce systems that will still be operational in the event of any single point of failure, ie, a network connection failure between facilities or the failure of a single archive server within the distributed system. During normal operation, workload for image acquisition, image routing and image query requests will be distributed between the archive servers. If the system is deployed in a multifacility environment, each archive server can be configured to be responsible for the acquisition and image distribution management within that server's local facility. If the system is deployed in a single facility environment, load can be distributed evenly between the archive servers based on an understanding of the workload requirements generated by each acquisition and display device in the system. In the event that an archive server fails, other archive servers within the system will have the ability to provide

some or all of the failed server's functionality. The degree of fail-over capability is dependent on the archive server's configuration as well as hardware redundancy employed. Three levels of fault-tolerant design can be achieved with this system architecture: (1) duplicate work capability only; (2) duplicate work capability and short-term image cache; (3) duplicate work capability, short-term image cache, and long-term image archival. Using the basic fault-tolerant design above, we have implemented a multifacility distributed archive system at the University of Utah. This system was implemented at a fraction of the cost of true "high-availability" archive architectures yet provides constant up time for the PACS system. If the network connection between the two locations goes down, each site is still fully functional for soft-copy read, as well as image acquisition and distribution. If either of the archive servers goes down, the image sources are redirected to the other archive server. The operational server then handles image distribution for both locations. Access to images in the short-term image cache is available to both archive servers and is not affected by loss of the network connection or remote server. Because there is only one long-term archive device, the ability to retrieve images from long-term storage is the only function compromised by a network or server failure. **Conclusion:** By implementing distributed archives in a PACS environment, it is possible to achieve a highly fault-tolerant system without the expense of high-availability hardware and software. The design concepts outlined here can be applied to any PACS system that supports distributed archive functionality.

Copyright © 2001 by W.B. Saunders Company

THE USE OF redundant server hardware and data storage methods, such as redundant array of inexpensive disks (RAID) is ubiquitous in today's market. While these methods provide a reasonable level of assurance that certain failures will not severely affect the operation of the archive system, they do not guard against the fact that the a single archive server represents a single point of failure. If we consider that the loss of an archive's functionality or a network failure could severely impact the operation of an imaging facility, or more importantly, the treatment of a patient, then a more fault-tolerant solution must be contemplated.

Distributed archives in a picture archiving and communication system (PACS) environment can provide added fault tolerance and fail-over capability,

From Marconi Medical Systems, Cleveland, OH; and the University of Utah Medical Center, Salt Lake City, UT.

Address reprint requests to Rebecca Mendenhall, MS, Marconi Medical Systems, Radiology Information Services, 595 Minor Rd, Cleveland, OH 44143. E-mail: rebecca.mendenhall@marconi.com.

Copyright © 2001 by W.B. Saunders Company

0897-1889/01/1402-1020\$35.00/0

doi:10.1053/jdim.2001.23824

as well as increased load capacity at a more economical price than traditional "high-availability" systems, ie, multiple servers synchronized at the operating system (O/S) level. Systems can be configured with varying levels of fault tolerance, depending on the amount of redundancy desired. There is, however, a direct correlation between the level of hardware redundancy and the cost of implementation.

This report details the system design for fault-tolerant distributed archives as well as several options for redundancy. Implementation of a fault-tolerant archive system at the University of Utah is described.

METHODS

The distributed archive systems described here are based on Marconi's Intellistore image archive software (Marconi Medical Systems, Cleveland, OH). This software can be implemented on multiple individual archive servers in order to distribute archive functionality and operational load. The configuration and implementation of the individual servers together makes up the distributed archive system, which could be scaled to meet future requirements. Several implementation and configuration options exist, including the ability for servers to maintain duplicated databases containing patient and image information. With this configuration, each archive can be aware of all information and the location of this information within the distributed archive system. Not only does this create a more fault-tolerant, redundant archive system, but it has the added benefit of distributing system load across multiple systems. Each archive server will still be equipped with redundant hardware components in order to maintain data and operational integrity at the server level.

The implementation of a distributed archive system involves the definition of system and business requirements and a distributed system design that will best meet these requirements. The following is a list of objectives of a distributed system that should be evaluated when designing a distributed solution.

Data availability. Image data must remain available in the event of an archive server or network failure. This requirement is usually achieved through data redundancy across multiple archive servers, as well as the redundancy of the servers themselves. There are varying degrees of data redundancy possible in any distributed system and part of the design work is determining what level will satisfy the financial and operational requirements for the current system.

Application availability. The failure of an individual archive server should not impact the main functionality of an archive system, namely, the storage and distribution of image data. This can be achieved by replicating the archive's functionality across all participating archives, or by implementing specific archive functionality or components, such as storage or image distribution management, on individual archive servers. Either way, the functionality is available in multiple locations in the event of a failure.

System serviceability. Planned outages required to perform regular maintenance and upgrades should not adversely impact

the ongoing operation of image data storage and distribution. By having multiple archive servers in the system, maintenance can be performed on each archive server individually without affecting operation of the system as a whole.

Load distribution. The distribution of archive functionality allows for the load associated with different functionality and components to be distributed evenly across several archive servers and network devices.

The goal is to produce systems that will still be operational in the event of any single point of failure, ie, a network connection failure between facilities or the failure of a single archive server within the distributed system. During normal operation, workload for image acquisition, image routing and image query requests will be distributed between the archive servers. If the system is deployed in a multifacility environment, each archive server can be configured to be responsible for the acquisition and image distribution management within that server's local facility. If the system is deployed in a single-facility environment, load can be distributed evenly between the archive servers.

In the event that an archive server does fail, other archive servers within the system will have the ability to provide some or all of the failed server's functionality. The degree of fail-over capability is dependant on the archive servers configuration as well as hardware redundancy employed. Three levels of fault-tolerant design can be achieved with this system architecture. These are listed in increasing order of fault tolerance provided, as well as cost to implement:

- Duplicate work capability only
- Duplicate work capability and short-term image cache
- Duplicate work capability, short-term image cache, and long-term image archival

It is the task of the design team to determine which level of fault tolerance is appropriate for the current system.

The primary requirement for a distributed archive system that can duplicate work capability in the event of any single failure in the system is a duplicated or shared database. Patient demographics, image locations, image sources, image destinations, routing rules, and prefetch logic for the entire system must be known to any archive server that is designed to provide fail-over capability. During normal operation, some of the failover logic, ie, intelligent routing and prefetch rules, may need to be disabled but they must exist on more than one server for there to be redundancy in the system. However, these servers can be configured so the short-term image cache is not redundant from one system to the next. This allows each server in the system to have a smaller RAID array, thereby reducing the total system cost. The disadvantage to this is in the event of a single archive failure, the rest of the system will not have access to the images in the failed servers image cache and will have to retrieve any such images from long-term storage.

To add redundancy in the short-term image cache is a very simple configuration change on the archives, but it requires every server in the system to have a RAID array equal in size to the sum of each individual server's requirements. This can get quite costly. For the implementation at the University of Utah this would have required about 1.2 TB of RAID on each archive server. This does, however, provide higher availability of images in the short-term image cache in the event of a single failure within the system, eliminating unnecessary retrievals from long-term image storage.

The highest level of redundancy can be achieved by adding a duplicate tape unit to the distributed system design. Again, this is a fairly simple change to configure on the archive servers. In this scenario, the two tape libraries will be exact mirrors. Thus if either tape unit fails, all images are still available from the second unit. The cost implications for adding a second tape system are even greater, however, than adding redundant short term image cache. The tape library is also the highest maintenance component of the archive system. Adding a second unit doubles that requirement, another factor to consider when determining what level of redundancy is acceptable. Storage area network (SAN) or networked area storage (NAS) technology may provide some alternative solutions here but is still a maturing technology and can be cost-prohibitive.

Although the following section describes specific hardware components, the design concepts in this report can be implemented with any server, image cache, and long-term archival media. The key is a software architecture and system design that allows the distributed archive servers to replicate databases and share or duplicate various components to produce fault-tolerant performance.

RESULTS

Using the most basic fault-tolerant design described in the previous section, we have implemented a multifacility distributed archive system at the University of Utah. The University of Utah PACS system handles up to 200,000 examinations per year between the main hospital system and remote clinics included in the network. The system must be operational 7 days a week, 24 hours a day, 365 days a year. Down time is not an option.

The distributed system incorporates two independent archive servers in two separate buildings. Each archive has a Sun server running Solaris 7.0 (Sun Microsystems, Palo Alto, CA) with a short-term image cache on external disk arrays in a RAID 5 configuration. One archive also has an attached StorageTek (Denver, CO) 9710 tape library with 9840 tape technology for long term image storage managed by hierarchical storage management (HSM) software. The systems communicate via an OC12 backbone that is part of the University of Utah network system. Both systems have a synchronized Oracle (Redwood Shores,

CA) database that stores patient and study level information, as well as the locations of all images within the system (Fig 1).

If all systems are functioning as normal, load is shared between the two archives based on the image source. The hospital archive (Archive 1) will receive all images acquired at the hospital, while the outpatient archive (Archive 2) will receive all images acquired from the University of Utah outpatient imaging systems, including a dozen remote clinics. Note that the outpatient archive (Archive 2) has the tape system attached. Thus it is the only archive that can directly write or retrieve images from tape. Each archive is responsible for routing received images to proper workstations for review, generating prefetch requests to support reading of these examinations, and ensuring long-term archival is completed for all images it receives (Table 1). Note that under this system design although the database between the two archives is redundant, ie, the same database exists on both servers, the short-term image cache is not. Each archive server would need to double its RAID capacity to reach this level of redundancy. As discussed previously, this level of redundancy was deemed too costly for the benefit it provided.

Should there be a failure in the network communication between the two facilities, both archives will continue to operate as usual servicing their local system demands. The only functionality lost in this scenario is that images received by the hospital will not actually be archived to tape until the communication link is restored. For added protection, this data is stored on mirrored RAID until its storage on long-term media has been confirmed. Likewise, prefetch requests at the hospital for examinations that are stored on tape will not be able to be serviced until the communication link is restored. A failure of the tape library itself would have a similar effect (Table 2).

Should there be a failure of either archive server,

Fig 1. University of Utah distributed archive. Configuration

Table 1. Distributed Archive Responsibilities Under Normal Conditions

Normal Operation	Archive 1—Hospital	Archive 2—Outpatient
Image acquisition	Local only	Local only
Image routing	Local only	Local only
Image archival	Request remote archival for local images	Archival for local images
Image prefetch	Request remote prefetch for local images	Prefetch for local images

the opposite server will be configured to immediately take over image acquisition and routing for all nodes in the system. If it is the hospital archive (Archive 1) that failed, full tape system capabilities will be maintained. This means image archival and prior exam prefetch will not be affected. If the outpatient archive fails (Archive 2), there will be no image archival or retrieval from long-term storage until functionality is returned (Table 3).

CONCLUSIONS

The distribution or separation of system functionality and components that results from a distributed system implementation creates several benefits:

Scalability. Adding additional capacity to one of the archives, or the addition of a new archive, does not require a total redesign of the distributed system or the rebuilding of any servers.

Fault tolerance. The failure of a single component or server has less affect in a distributed system than in a single-server implementation, since the data and functionality has been made available on other servers.

Application-specific configurations. The distribution of archive functionality and components can

Table 3. Distributed Archive Responsibilities When an Archive Server Fails

Failed Archive Scenarios	Archive 2 Fails, Archive 1—Hospital Takes Over	Archive 1 Fails, Archive 2—Outpatient Takes Over
Image acquisition	All images in system	All images in system
Image routing	All images in system	All images in system
Image archival	Queue requests for all archival	All images in system archived
Image prefetch	Queue requests for all prefetch	All images in system prefetched

allow hardware configurations to be tailored to fit specific requirements. For example, a server can be implemented that provides optimal performance as a web server where a high number of small transactions are the norm. Alternatively, a server that houses the database can be configured to provide optimal performance for a small number of large transactions. The ability to “tune” the server’s configuration for the intended workload is not usually possible on single systems that house multiple applications.

The distributed archive system deployed at the University of Utah was implemented at a fraction of the cost of true “high -availability” archive architectures yet provides the University with constant up time for their PACS system. If the network connection between the two locations goes down, each site is still fully functional for soft-copy read, as well as image acquisition and distribution. If either of the archive servers goes down, the image sources are redirected to the other archive server. The operational server then handles image distribution for both locations. Access to images in the short term image cache is available to both archive servers and is not affected by loss of the network connection or remote server. Due to the fact that there is only one long-term archive device in this system implementation, the ability to retrieve images from long term storage is the only function compromised by a network or server failure.

By implementing distributed archives in a PACS environment it is possible to achieve a highly fault-tolerant PACS system without the expense of high-availability hardware and software. The design concepts outlined here can be applied to any PACS system that supports distributed archive functionality.

Table 2. Distributed Archive Responsibilities under Failed Network or Tape System

Failed Network or Tape System	Archive 1—Hospital	Archive 2—Outpatient
Image acquisition	Local only	Local only
Image routing	Local only	Local only
Image archival	Queue requests for remote archival	Archival for local images
Image prefetch	Queue request for remote prefetch	Prefetch for local images