the Availability Digest

www.availabilitydigest.com

Beyond Redundancy

How Geographic Redundancy Can Improve Service Availability and Reliability of Computer-Based Systems

May 2012

The book Beyond Redundancy[1] provides an in-depth analysis of various approaches to geographical redundancy of IT systems to improve service availability. Among several recommendations, it concludes that the superior approach is the use of active/active systems with client-initiated failure detection.

The book is authored by Eric Bauer, Randee Adams, and Daniel Eustace, senior staff members of Alcatel-Lucent, all of whom specialize in system reliability.

Enterprises commonly make significant investments duplicating critical systems in geographically dispersed sites so their IT services can survive any disaster. There are many additional advantages that can be derived from two or more processing sites beyond disaster recovery. In particular, hardware faults, software bugs, and human errors that may render a site inoperable for a period of time can be mitigated by the processing capacity of an alternate site. Response times can be improved by locating sites close to communities of users. Geopolitical issues such as language, currency, and regulatory differences of various countries can be resolved by positioning processing sites in each area.

Beyond Redundancy focuses on the theoretical and practical aspects of the benefits of georedundancy on service availability and reliability. It is organized into three sections:

Part 1: Basics provides background and definition of terms for georedundant systems and service availability.

Part 2: Modeling and Analysis of Redundancy uses Markov chain modeling to analyze a variety of redundant architectures intended to increase service availability.

Part 3: Recommendations uses the results of *Part 2* to offer recommendations on architecture, requirements, design, and testing of georedundant configurations.

Part 1: Basics

Service-Criticality Levels

The authors define four levels of service criticality that they later use to select from a variety of georedundant architectures:

[1] Beyond Redundancy: How Geographic Redundancy Can Improve Service Availability and Reliability of Computer-Based Systems, Eric Bauer, Randee Adams, and Daniel Eustace, *John Wiley and Sons*; 2012.

Criticality	Impact	Availability
Safety Critical	Unacceptable safety hazard	seven 9s
Efficiency Critical	Local or system-wide economic impact	five 9s
Essential	Loss can be accommodated by reducing capacity	three 9s
Routine	Minor impact	two 9s

Georedundancy

Failure events can be caused by hardware failures, software bugs, network overloads and outages, environmental faults (power, air conditioning), people, and policies. Catastrophic events that impact critical infrastructure and that cannot be rapidly recovered locally are typically mitigated by having a redundant system that is physically separate from the primary facility at a far enough distance so that it is unlikely that both sites will be adversely affected by a common event. The sites are each configured to provide sufficient capacity to handle the full processing load for normal operations. They may be organized as

- A primary site with a cold standby site that is provisioned with the proper equipment but is otherwise not in operation.
- A primary site with a warm standby site in which all equipment is powered up and applications are loaded and ready to operate once the database has been loaded.
- A primary site with a hot standby site in which all equipment is powered up, applications are loaded, and a current copy of the database is maintained.
- An active/active pair of sites in which each site is active and is sharing the work load.

These configurations provide, in order, decreasing (better) recovery time to meet a specified Recovery Time Objective (RTO). RTOs can range from days for a cold standby system to hours for a warm or hot standby system to minutes or seconds for an active/active system.

The amount of data that may be lost – the Recovery Point Objective, or RPO – is a function of the time between backups. Cold and warm standby sites that depend upon periodic backups can experience hours or days of data loss. If data replication is used to keep the databases of a hot standby site synchronized with the primary site, data loss can be reduced to hours or even minutes. Data loss in active/active systems can be zero or measured in seconds or minutes.

The rapidity of recovery to a remote site depends strongly upon facilities that are incorporated into the systems for fault detection, fault isolation, rapid reporting of fault conditions, and automated recovery. The effectiveness of site recovery also has a distinct human aspect. It is affected by training, documentation, complexity (manual procedures versus automated procedures), and staff coordination.

Service Quality

In their book, the authors consider two measures of service quality – *service availability* and *service reliability*. Service availability is the readiness of the system to provide service to a user. It is the probability that the system is up and running and is usually expressed in *nines*. If the system is up 99.9% of the time, it has an availability of three 9s. The definition of service availability is complicated by several considerations:

- What is the minimum lack-of-service time that will be considered an outage?
- How are outages for only a proportion of users to be counted?
- How are outages of only certain functions to be counted?
- Following a failover, is fall back time to a good primary system to be counted?

- Is planned downtime to be included?

All of these and other factors must be specified in a Service Level Agreement (SLA) between the IT organization and the users of the services.

Service reliability is the probability that service requests are fulfilled correctly with acceptable response time. Reliability has two parts:

- Accessibility – the probability that a session will be successfully initiated with a service.
- Retainability – the probability that the service is delivered with acceptable quality during the session.

Reliability may be measured via logs, service probes sending test traffic, or by tracking the ratio of error responses to normal responses. Reliability is usually reported as failed operations per million attempted operations. Reliability is often not calculated while the system is down.

Part 2: Modeling and Analysis of Redundancy

The authors consider two types of redundancy – internal redundancy and external redundancy. Internal redundancy is used to recover from faults within a system. External redundancy incorporates two or more systems in a pool that can offer higher availability or capacity to users. The systems may either be collocated or geographically dispersed.

Markov Modeling

The various redundancy architectures are analyzed via reliability block diagrams and Markov modeling. Most of the analysis in the book is done via Markov models. A Markov chain is a directed graph comprising nodes and links (edges) between the nodes. The nodes represent system states (such as UP or DOWN). The links represent the rate at which the system transitions from one state to another. Since the probability of entering a state must be equal to the probability of exiting the state, a series of linear algebraic equations can be easily generated that can be solved for the probability that the system will be in each state.

Though the generation of the linear equations is straightforward, their solution is tedious. A system with eight states will have eight independent linear equations. To solve for the state probabilities, it is best to use a software utility for solving the equations. In the book, the authors give several examples of system availability solved via Markov models. However, general result equations are not given. If the availability of a particular system architecture is desired using different parameters than those used by the authors, its Markov model will have to be solved with those parameters.[2]

System States

For internal redundancy, the authors assume five states for a system (five nodes in the Markov model):

- Up – the system is up and is running in either simplex or full redundancy mode. Simplex mode is operation with a component failure but with its redundant backup component. The system has a single point of failure, but users are unaware of the fault.
- Down Uncovered – the system is down, but no one knows it. The operators have no indication that the system is down. The first indication might be frantic calls coming in to the help desk. When the outage is discovered, the system enters the Down Covered state.

[2] For examples of general solutions to availability Markov models, see Appendix 3: Failover Fault Models, *Breaking the Availability Barrier: Survivable Systems for Enterprise Computing*, by Dr. Bill Highleyman, Paul J. Holenstein, and Dr. Bruce Holenstein, AuthorHouse; 2004.

- Down Covered – the system is down and is undergoing active recovery, either automatically or by the operations staff. Upon successful failover to the redundant backup component, the system is recovered and enters the Up state.
- Failed Automatic Failover – the system attempted to fail over to the redundant backup component, but the failover failed. The failover must be accomplished manually before the system can enter the Up state.
- Duplex Failed – both components of a redundant pair have failed. One must be repaired before the system can return to the Up state.

For georedundant systems, the internal redundancy model above describes the availability states of each of the geographically dispersed systems. The authors overlay this model with three additional states to account for failover to a remote system:

- Up on Georedundant – the service has been recovered to the georedundant remote backup system.
- All Failed – both the primary system and the georedundant backup system have failed.
- Service Migration – the switchback from the backup georedundant system to the primary system is underway, during which time service is interrupted.

Many parameters are involved in these models, but the primary ones include:

- the component failure rate.
- failure coverage – the ability to rapidly detect and isolate a fault.
- failover latency – the time it takes to fail over to a redundant component and to restore service.
- failover success – the probability that the failover will be successful

Markov models are used to analyze simplex systems (those with no redundancy), active/standby (warm or hot) configurations, and active/active configurations.

The high availability strategy to be adopted must meet the RTO and RPO specifications given in the SLA and fall within the capital expense and operational expense budgets of the user organization. (In practice, the conflicting requirements of availability and cost will often need to be negotiated to arrive at an acceptable solution.)

Fault Detection

The authors point out that georedundancy is inherently more difficult than internal redundancy. Within a single system, there are many internal checks to detect failures. In a georedundant system, failure detection depends upon communication over network links to monitor the health of other components. There are several architectures for monitoring system health:

- Systems can monitor themselves and report problems.
- Heartbeats can be interchanged between systems to monitor the health of remote systems.
- An independent system can monitor the health of systems and determine recovery procedures.
- Clients can detect failures by the lack of responses to their requests.

The time from when a fault interrupts service to the time that the fault is detected is the *Uncovered State* described above and determines the *failure coverage* parameter in the model. Once detected, the time to recover service to the users is the *failover latency* parameter described above. The probability that the failover will be successful and not require manual intervention is the *failover success* parameter.

Analysis of Georedundant Approaches

Given the above definitions, the authors use Markov models to calculate the availabilities of several georedundant architectures:

- Simplex system (no georedundant backup site)
- Manual switchover to a hot backup site – once a failure is reported by the operational system, the operators use manual procedures to switch to the backup site.
- System-driven switchover to a backup site – the system detects an outage, perhaps through heartbeats or an independent monitoring system, and automatically switches over to the backup site.
- Client-initiated failover to a backup site – the clients independently detect a system failure based on return error codes or lack of a response and redirect their requests to the backup site.

Manual switchovers and system-driven switchovers are to hot standbys. Client-initiated failovers are used in active/active configurations.

Using the parameters specified by the authors, they calculate that a single site will have an availability of 99.9973% (almost five nines). Configuring this site for georedundancy, they calculate the following availabilities for different architectures:

Simplex system (no backup site)	99.9973%	(almost five 9s)
Manual switchover to hot backup site	99.9974%	(almost five 9s)
System-driven switchover to hot backup site	99.9975%	(almost five 9s)
Client-initiated switchover to active site	99.9996%	(almost six 9s)

At first glance, these results seem disturbing. As the authors conclude, having a backup site with manual or system-driven recovery is no better from an availability viewpoint than a single simplex system. So why even bother with the cost and complexity of a backup site? The reason, of course, is for disaster recovery.

Uncovered Outages

How do they come to this non-intuitive conclusion? The answer is in their assumptions about uncovered outages. These are the "silent" outages that go unnoticed until some fault detection mechanism – manual or automatic – alerts the operation staff to the fault so that recovery actions can begin. During the uncovered outage, the system is down and users are not being serviced.

In their parameters, the authors assumed that the primary system would fail four times per year. 10% of these failures were silent failures that took an average of 30 minutes to be detected. Thus, the system was down 4 x 0.1 x 30 = 12 minutes per year due to silent failures. This in itself led to an availability of 99.9977%. For simplex systems and manually and system-driven switchover configurations, uncovered outages represented by far the bulk of downtime. Having the ability to switch over to another system had no impact on this failure mode, and because it represented by far the bulk of downtime (total downtime per year from all causes was 13 to 14 seconds), switchover had minimum benefits from an availability viewpoint.

Conversely, client-initiated failover does not depend upon an external fault detection mechanism. If a client cannot receive a legitimate response from a site after a small number of fast timeouts and retries, it simply submits its request to the other site. Failover time is measured in seconds, and uncovered outage time is minimal. With the authors' parameters, uncovered outage time for client-initiated switchover led to 0.4 minutes of downtime per year rather than 12 minutes. This is more than an order of magnitude less, leading to an additional nine of availability.

Having said that, the authors' calculations show a marked difference in recovery times. Using typical parameters, manual switchover times are measured in hours. System-driven switchover, which is generally accomplished automatically, is measured in minutes. Client-initiated switchover is measured in seconds.

If one wants to evaluate these models for other values of uncovered downtime, the Markov models will have to be solved for these cases.

Client-Initiated Recovery

Having made their point about the advantages of client-initiated recovery, the authors next focus on the primary parameters governing the effectiveness of this technique. The key to this mode of georedundancy is that the client can quickly detect a site failure and rapidly switch over to the alternate site. Three parameters are important:

- Fault Detection – This is the amount of time that it takes for a client to decide that it is not going to get a valid response from its current site. Typically, a client will make a request and will then wait for the response. If the response is not received within a certain period of time, then the client will retry the request. After a certain number of retries, it will move to the alternate site and submit its request to that site. The fault detection time is then the request timeout times the number of retries. The time for a client to decide to switchover to a backup site should be fast enough to meet the SLA, but not so fast that its current site does not have enough time to attempt its own automatic recovery.

- Site Monitoring – Using just request timeouts to determine a site failure means that the client will not know of a site failure until it makes its next request. This can be improved by having the client proactively monitor the health of the site by sending it periodic heartbeats. If it fails to receive responses to a certain number of heartbeats, it can assume that the site is down; and it can reconnect to the other site. When it desires to send its next request, it has already gone through the recovery process and there is no downtime attributable to the primary site failure. Successful responses can be considered successful heartbeats. The more frequent the heartbeats, the less likely that the client will try to make a request on an inoperable site.

- Recovery – In order to recover, the client must reconnect to the alternate site. This will typically require that the connection to the alternate site be established and a new session initiated – a process that can be time consuming. There are several techniques for minimizing this time. The client could maintain simultaneous sessions with both sites. Session parameters could be replicated to the backup site so that it is ready to continue the session transparently. Session parameters could be maintained on the client, such as via cookies in a browser. If recovery is fast enough, the fault may not be counted as an availability failure according to the SLA. However, if the session is lost, the outage will be counted as a reliability failure since the request was not successfully processed.

Case Studies

Using the techniques presented in Part 2, the authors present two detailed case studies. One analyses the availability of a pool of DNS servers using client-initiated recovery.

The other is a complex of applications representative of a typical site. In this study, a client interfaces the system via a front-end server, which uses a database server to satisfy client requests. Several systems support these elements including an authentication server; a provisioning system; a backup system; a usage database for functions such as billing, compliance, and performance monitoring; and a system management facility. The authors introduce the concept of failing over single functions rather than the entire site for certain facilities. The criticality associated with each function in this example is as follows:

End user	efficiency critical
Other enterprise systems or software	efficiency critical
Legal/regulatory compliance support systems	efficiency critical
Maintenance engineers	essential
Network management	essential
Provisioning	essential
Accounting and billing	routine
Performance management	routine
Data backup	routine

This system is analyzed in detail in both Chapters 9 and 14 to demonstrate that the guidance provided in the book can be used in a real-life example.

Part 3: Recommendations

The authors conclude the book with a series of recommendations covering site location choices, recovery strategies, the use of cloud computing for site backup, and the issues that should be covered by an SLA.

They focus heavily on testing. They break redundancy testing down into four phases:

- Element testing, in which the recovery mechanism for each element of the final solution is tested to ensure that the element can recover successfully.
- End-to-end testing, in which site failover of the entire system is tested in a laboratory environment.
- Deployment testing, in which the redundant sites are tested in the production network environment.
- Operational testing, in which the production system is periodically tested to ensure that the documentation is kept up to date and that the staff is trained in fault detection and failover procedures.

Summary

This book is not a casual read. It is intended for the serious student who needs to understand the various aspects of georedundant systems from an analytical as well as a practical viewpoint. As such, it serves two purposes:

- For those who are mathematically challenged, it is an excellent treatise on the concepts behind georedundancy and the issues that must be considered in designing such systems. These concepts are easily digested by bypassing the Markov model discussions.

- For those who want to be able to specifically analyze various configurations, it provides the analytical tools to do so via its Markov models.

The book is biased on the importance of uncovered outages, concluding that manual and system-driven switchovers to standby sites provide no availability improvement of consequence beyond disaster recovery. While it is true that active/active systems with client-initiated failover will provide the best availability by far of any of the georedundant configurations, the provision of good fault detection mechanisms in the systems will certainly improve the availability of active/standby sites. The impact of improved fault coverage can be determined by solving the book's Markov models with modified parameters.