Realtime
publishers

The Shortcut Guide™ To

Untangling the Differences Between High Availability and Disaster Recovery

sponsored by

MARATHON
Run to Infinity

Richard Siddaway

Introduction to Realtime Publishers

by Don Jones, Series Editor

For several years now, Realtime has produced dozens and dozens of high-quality books that just happen to be delivered in electronic format—at no cost to you, the reader. We've made this unique publishing model work through the generous support and cooperation of our sponsors, who agree to bear each book's production expenses for the benefit of our readers.

Although we've always offered our publications to you for free, don't think for a moment that quality is anything less than our top priority. My job is to make sure that our books are as good as—and in most cases better than—any printed book that would cost you $40 or more. Our electronic publishing model offers several advantages over printed books: You receive chapters literally as fast as our authors produce them (hence the "realtime" aspect of our model), and we can update chapters to reflect the latest changes in technology.

I want to point out that our books are by no means paid advertisements or white papers. We're an independent publishing company, and an important aspect of my job is to make sure that our authors are free to voice their expertise and opinions without reservation or restriction. We maintain complete editorial control of our publications, and I'm proud that we've produced so many quality books over the past years.

I want to extend an invitation to visit us at http://nexus.realtimepublishers.com, especially if you've received this publication from a friend or colleague. We have a wide variety of additional books on a range of topics, and you're sure to find something that's of interest to you—and it won't cost you a thing. We hope you'll continue to come to Realtime for your educational needs far into the future.

Until then, enjoy.

Don Jones

Copyright Statement

© 2010 Realtime Publishers. All rights reserved. This site contains materials that have been created, developed, or commissioned by, and published with the permission of, Realtime Publishers (the “Materials”) and this site and any such Materials are protected by international copyright and trademark laws.

THE MATERIALS ARE PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE AND NON-INFRINGEMENT. The Materials are subject to change without notice and do not represent a commitment on the part of Realtime Publishers its web site sponsors. In no event shall Realtime Publishers or its web site sponsors be held liable for technical or editorial errors or omissions contained in the Materials, including without limitation, for any direct, indirect, incidental, special, exemplary or consequential damages whatsoever resulting from the use of any information contained in the Materials.

The Materials (including but not limited to the text, images, audio, and/or video) may not be copied, reproduced, republished, uploaded, posted, transmitted, or distributed in any way, in whole or in part, except that one copy may be downloaded for your personal, non-commercial use on a single computer. In connection with such use, you may not modify or obscure any copyright or other proprietary notice.

The Materials may contain trademarks, services marks and logos that are the property of third parties. You are not permitted to use these trademarks, services marks or logos without prior written consent of such third parties.

Realtime Publishers and the Realtime Publishers logo are registered in the US Patent & Trademark Office. All other product or service names are the property of their respective owners.

If you have any questions about these terms, or if you would like information about licensing materials from Realtime Publishers, please contact us via e-mail at info@realtimepublishers.com.

[**Editor's Note:** This book was downloaded from Realtime Nexus—The Digital Library for IT Professionals. All leading technology books from Realtime Publishers can be found at http://nexus.realtimepublishers.com.]

Chapter 1: What Is High Availability?

High availability and disaster recovery are two topics that are often tangled in thought and action. This guide will untangle the differences as well as explain the similarities and where the two areas converge. The guide starts with this chapter explaining high availability—what it is and what needs to be considered when implementing a highly available infrastructure.

Chapter 2 introduces disaster recovery, explaining the concept and comparing it with high availability—planning, implementation, and testing are discussed. The chapter closes with a look at how the technologies enabling high availability and disaster recovery are producing a convergence in how the two are implemented.

Chapter 3 returns to high availability and examines how you can configure your environment to be highly available. The chapter examines the reasons systems become unavailable and looks at traditional high-availability solutions such as clustering. We explore high availability from applications such as Microsoft Exchange Server and Microsoft SQL Server and discuss how virtualization brings its own availability challenges and solutions to the mix.

High availability is not created by technology alone. We also need to consider the people and the process they operate, which we explore in Chapter 4. This chapter considers the causes of down time and how you can eliminate the largest cause of unplanned downtime; it also discusses the impact of people and processes on high availability. A consideration of how you can apply the techniques and concepts of high availability into the disaster recovery arena closes the chapter. The first area we need to look at is high availability.

What Is High Availability?

High availability is often defined as systems that are designed, and implemented, to provide a designated level of operational availability during a given period. Availability is often defined as the ability of the users to access, and utilize, their systems.

This definition often results in a narrow view that doesn't cover the whole spectrum you need to consider. The first paragraph in this chapter closed on the phrase *highly available infrastructure*. This phrase gets to the heart of your requirements. A highly available infrastructure includes, among others:

- Server and storage hardware
- Data
- Network
- Maintenance
- Monitoring
- Configurations

We'll see how these areas relate as we proceed through the chapter.

What It Isn't

High availability is many things to many people, but there are a number of things that it isn't. The first thing to remember is that high availability is not a "silver bullet." Creating a cluster and using it for your email or database system will not solve your availability problems if they are due to network failures or poor administrative procedures.

The other important point to remember is that there isn't a single answer that will solve your availability problems. Assume for a moment that you have an application for which you need to ensure availability. There are a number of components that all have to be available:

- One or more servers
- Databases
- Applications
- Storage
- Network links

Each of these components will require their own availability solutions. The individual availability solutions will have to be integrated to ensure the whole system is configured for high availability.

High availability is not a technology solution. We will come back to this point later, but an environment can only be considered as having high availability when three things are in place:

- The correct technologies
- The correct people in terms of knowledge, skills, and mindset
- The correct processes to ensure that administrative actions contribute to high availability rather than negate it

These three items form the high-availability framework that is required in every situation. We will also see that this framework carries across into our disaster recovery discussion.

Defining Availability

Availability can be defined in a number of ways and by a number of people. A common measurement is to use the Mean Time Between Failures (MTBF) and the Mean Time to Repair (MTR) to define availability.

$$\text{Availability} = \frac{\text{Mean Time Between Failures}}{\text{Mean Time Between Failures} + \text{Mean Time To Repair}} * 100$$

When dealing with individual components such as a disk drive, this measure is acceptable, but the formula can be difficult to calculate when there are a number of items forming a chain of availability. All the links in the chain must be available for the service to be available. As a subjective measure, consider this chain:

1. In the data center, the server is up and running, the service hosting the application has loaded, and the storage systems are online.
2. The network administrators report that the network links between the data center and the user's location are up and have sufficient bandwidth.
3. The user's PC has started and loaded the client application.

Everything seems to be working but the user cannot access the application. As far as the user is concerned, the application has failed. That failure could be anywhere in the chain; for example, it could occur:

- Between the links and areas of responsibility outlined earlier
- At the firewall on the server because of a misconfiguration
- At the user's PC because of an error in the way it is set up or even that the network cable has been knocked out

The user just wants to access the application. She doesn't care that the server is up or the network links are not reporting a problem. As far as she is concerned, her application is down. If one user reports a problem, you might be OK, but if a lot of users cannot access the application, the downtime causes a significant problem for the organization.

In remote locations and branch offices, the problem becomes worse. There may not be anyone to help resolve the local parts of the problem.

Cloud computing is a rising trend in the supply of IT services. The application is provided by a third party hosted from their data center. In theory, all the user needs is an Internet connection. The number of links in the chain grows very rapidly in this scenario with the users, the provider, and one or more ISPs involved.

The final judge on availability is the user. If she cannot use her application to perform her working tasks, then it is not available. It does not matter how many statistics can be generated proving that this part or that part of the chain is available if the user cannot access and use the application.

Service Level Agreements (SLAs) are used to define an agreed level of service. That can be equated to availability when thinking about applications. SLAs are often talked about, but are they worth the paper they are written on? Unfortunately, as with so much in IT, the only possible answer is "it depends." It depends on a number of factors:

- Is availability being reported from the users' perspective?
- What penalties are in place for breaching the SLA, and are the users prepared to invoke them?
- Do the users trust the reports?

In these cases, a better measurement for availability is based on the service availability.

$$\text{Availability} = \frac{\text{Agreed Service Time} - \text{Down Time}}{\text{Agreed Service Time}} * 100$$

These issues are compounded by the fact that downtime is not just caused by failure.

Capacity

Every computer system has a finite capacity. The hardware capacity can be measured by:

- CPU cycles
- Network bandwidth
- Memory
- Disk space

This relationship is illustrated in Figure 1.1. If any of these components reach capacity (that is, they are fully consumed), the system can fail. Even approaching capacity can put enough strain on to the system that it appears to be unavailable because it has become unresponsive.

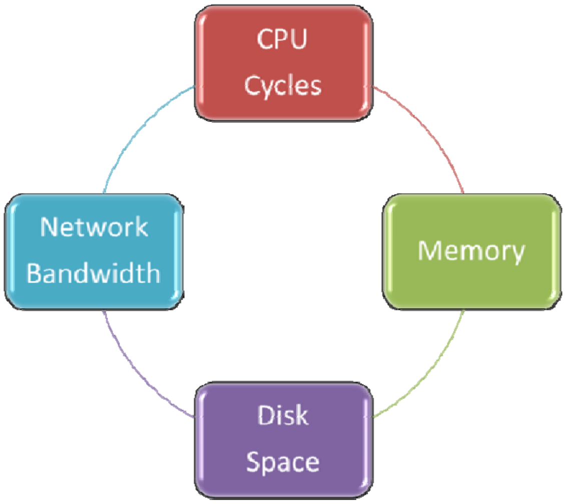

Figure 1.1 Hardware capacity constraints.

The correct design of the system at the outset should alleviate these issues. Design sufficient capacity into the system to manage the foreseeable capacity requirements. Specify the correct number and size of components. Proper planning at this stage can prevent major problems in the future.

Capacity issues are not just design issues. You can design and implement the perfect system, but over time, its capacity will degrade. The various components must be monitored to ensure that a capacity issue does not make your system unavailable. Trend analysis can identify when you will reach capacity for a given component. This data gives you time to arrange for additional capacity or other remedial work to ensure your system remains available.

When we talk about availability, it is usually in terms of the application. However, we need to consider the application, or service, and the data to obtain a complete picture.

Service vs. Data

Many discussions about availability concentrate on the availability of the service. Failover clustering is a technology designed to protect the services running on the cluster. It does nothing to protect the data.

Data is important, as it is the data that drives the business processes. Consider a system to process purchase orders. The data will include:

- Customers—Who ordered something
- Orders—What the customers ordered
- Delivery instructions—Where and when the items are to be delivered
- Invoices—How much the customer has to pay and whether they have paid

If this data is lost or corrupted, the financial impact on the organization could be huge. The regulatory requirements surrounding data, especially personal data, are becoming more and more onerous with stiff financial penalties for data loss or even incorrect data.

Ask the business what they want you to protect and the immediate answer will be everything. You can apply technologies to protect the service and you can apply technologies to protect the data. You should protect both. The ideal high-availability solution would protect both, be simple to set up and administer, and be cost effective. In many cases, you have to use a mixture of techniques. One proven technology that you need to consider is the venerable backup.

Do You Still Need to Back Up

I will answer this question and then explain why. The answer is simple and consists of one word. YES.

Backups will be a fact of the administrator's life for a long time to come. High availability does not negate the need for a backup. Consider Active Directory (AD). It could be argued that because you have multiple domain controllers in your environment that all contain a copy of the data, you don't need to back up. That can be taken as correct for day-to-day operations but, and it is a huge but, what are you going to do if you if your AD domain is wrecked?

A corruption of the database on one domain controller then replicates to every domain controller in the domain. Your AD has gone and so has access to every other aspect of your Windows environment.

The only way you can recover from this situation is from a backup. A slightly less traumatic occurrence is if a significant number of objects are deleted and you need to perform an authoritative restore. You need a backup to do that.

Single Domain Controller

A question that comes up on various forums with distressing regularity takes the form "I have one domain controller in my production domain and don't do backups. The machine has failed. What can I do?" The only real advice that can be given is to start applying for a new job.

In this case, the technology has not been used properly; the people did not have the skills and knowledge to understand the technology, and the correct processes were not applied. A business-critical technology that can be configured in a way that supplies high availability was allowed to fail because of a complete breakdown of all three parts of the high-availability framework.

Backup is the last resort for recovery. In some cases, it may be the only way a system can be recovered. It may not be financially viable to have standby systems for all the servers in your organization. In that case, recovery from backup is the only option.

The other purpose for backup is to provide a long-term archiving solution. In many business sectors, there are regulatory requirements regarding data retention. Backups may be one method of satisfying these requirements. I know of organizations that have to keep data for as long as 100 years. Backup tape is not going to solve that issue, but it should still be viable for the 3-to-7-year time span.

Backups will be with us for a long time and have a recognized place in high-availability and disaster recovery strategies. Those strategies must be based on solid business requirements. The next task is to look at how you can discover those requirements.

Business Requirements

A prime reason for the failure of IT-related projects is that the requirements were not captured correctly. This usually comes back to the business requirements—that is, what is the business trying to achieve? High-availability and disaster recovery projects are expensive propositions where it is imperative that the requirements are correctly understood.

Ask, Ask, and Ask Again

The only way to get this correct is to ask and keep asking until the full set of requirements have been captured. This will involve talking to multiple levels of the organization. The requirements at various managerial levels may well be different to those expressed by people actually using the systems. They all need to be captured and analyzed.

Think carefully about the answers you receive. They will all reflect the actual need but may not capture the whole requirement. Extrapolate failure scenarios. If you have developers, talk to them about use cases that describe how an application will be used. Apply the concept to failures to determine all the ways the system can fail, then work out how to stop it happening or how to recover from the failure.

When there is a failure of a critical system that hasn't been adequately protected, there is often a knee jerk reaction that "something must be done." If at all possible, avoid rushing into a solution. Follow the correct thought process:

- Analyze
- Design and plan
- Implement

Throughout this process, ensure that you

- Define the problem
- Refine the answer
- Double check you are solving the right problem

Above all else, ensure that the business processes are protected.

Business Processes

There are a number of drivers that will affect all organizations. The main two that affect our discussion of high availability and disaster recovery are financial and legal.

All organizations have financial drivers. A commercial organization will be interested in boosting revenues and reducing costs so that profits are maximized. Non-profit making organizations will also be looking to reduce costs so that their donations can be used to maximum benefit. Changes in the financial environment force the introduction or change of business processes.

The regulatory requirements organizations must satisfy form a constantly evolving landscape. These changes can impact business processes to drive change in the applications and infrastructure an organization requires.

The requirements of the business to satisfy their business drivers lead us to the hierarchy that Figure 1.2 shows. At the top is the business process. This is what the business does—how it generates revenue or makes efficiencies to cut costs. In many cases, this will require an application. It may be created by the organization or purchased from a software vendor. The application will usually have a requirement to store data. The store may be a database or an unstructured store such as the file system.

The application and data both require infrastructure for support. High availability is concerned with protecting the application, data, and infrastructure. Business continuity, of which disaster recovery is a part, is concerned with protecting the whole hierarchy.

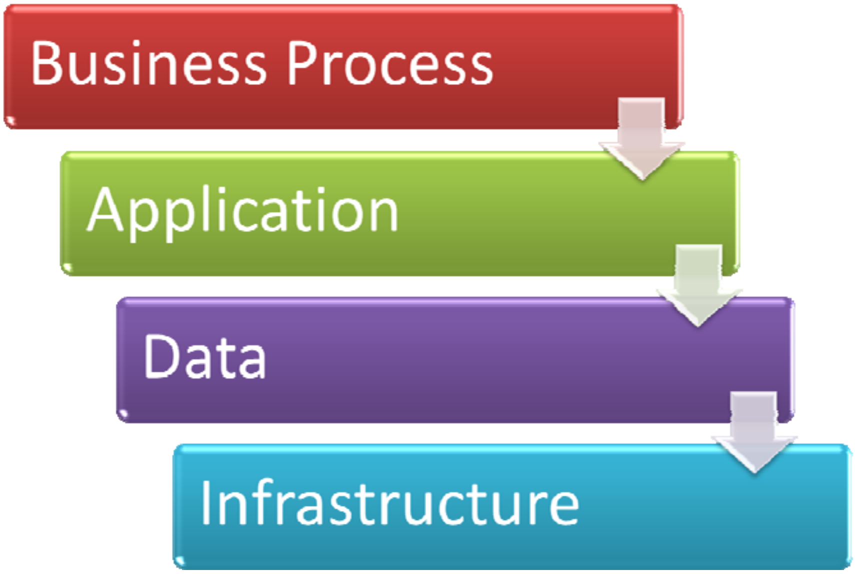

Figure 1.2 Relationship between business process and infrastructure.

Peak Periods

Every organization has peak periods; for instance, retail organizations have peaks at Christmas and financial services organizations have peaks at the end of the financial year. The applications supporting the business processes suffer an extra level of load at these times. This load can overpower the applications, causing downtime or the applications become so unresponsive that the users assume they are down. In either case, the users do not have access to their applications.

There are reports of organizations suffering downtime during peak periods every year. Why should this happen? Why is it allowed to happen? Peak periods are predictable. They are part of the business cycle and need to be treated as such. Even an event that falls outside the normal business cycle, such as a special offer or the release of new software for download, can be predicted to cause extra demand. The planning for these events should include extra computing capacity to handle the load. Notice I do not say expected load. In many cases, the load is predicted but the predictions prove to be wrong.

Virtualization can help in this situation. Need to bring another set of Web servers online? It is a quick and easy proposition to create and start up more virtual machines. The load on the Web farm should be monitored, and when it reaches a predefined level, one, or more, extra machines are brought online to spread the load.

Plan to prevent downtime during peak periods. A similar situation occurs with critical periods.

Critical Periods

Critical periods are where the organization has to perform a task with a tightly constrained timeframe. I spent a number of years working in a financial services company, and every year we had to produce a report for the regulatory authorities proving our ability to meet our liabilities incurred by the sale of financial products. If this report was not delivered by the specified time, we could be stopped from selling further products. That is about as critical as it gets.

These reporting deadlines may be legislative or imposed by a parent company. In either case, the deadlines must be met. The business processes required to meet these deadlines must remain available during these periods.

The cost of not having high availability can be quite fearsome but what is the cost of providing that availability?

The Cost of High Availability

Providing and running IT systems creates associated costs. The more sophisticated the systems, the higher those costs rise. High availability adds sophistication and therefore cost to the system, although the costs have been dropping in recent years as the technology has advanced.

The Cost of Doing Nothing

One option is doing nothing. You can assume that failures will never happen to you and that if you have a failure, the system can be brought back online very quickly. This assumes that the people, processes, and technology are in place to bring a system back online.

Is that assumption correct? Relying on this approach can, and will, fail for several reasons:

- The right people are unavailable. What are you going to do if the only person that really understands the system is on vacation and not reachable?
- The processes are unavailable. Have you tried to restore this system in a test environment? Is the recovery process documented?
- The technology is unavailable. Have you tested your backups? This is not a good time to discover you cannot read the tapes. Do you have a spare server or capacity on your virtualization hosts?

How much will it cost your organization while that system is unavailable? That cost could come from a number of sources:

- Lost orders
- Paying staff to do nothing
- Revenue collection delays

That is just the short-term financial cost. What about the long-term impact to your business reputation because you could not deliver as agreed?

Challenge these assumptions. The cost of failure is high and ultimately the price may be the collapse of the organization.

Does Your Business Depend on IT?

There are businesses that obviously depend on IT. Internet retailers are an obvious example. However, most businesses rely on IT to a much greater extent than they realize.

Do you use email at work? Research in Europe has shown that more than 60% of person-to-person communication is performed by email. This is both within the organization and outside the organization to customers, suppliers, regulators, and so on.

What happens if your email system fails? It does not have to be the whole system just the mailboxes for those people who need to communicate externally. If you cannot reach a company, do you assume that they have gone out of business or just don’t care about doing business with you? A prolonged outage leads to the situation that Figure 1.3 shows.

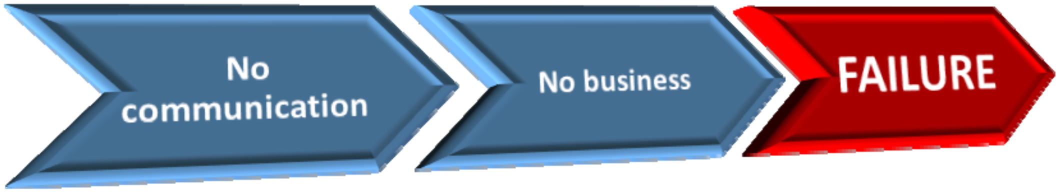

Figure 1.3: Impact of communication failure.

Every organization has data. It makes the organization work. Figure 1.2 showed the relationship between data, applications, and infrastructure. If you lose the data, you lose the application and the business process. This can lead to a loss of revenue, which is not a healthy position for the enterprise. In some cases, that revenue may not be recoverable. If the record of an invoice is lost, that revenue may never be received.

A Web presence is a must for most organizations. Some organizations derive all of their business from the Internet. If their Web presence fails, competitors are just a click away. Brand loyalty may be a disappearing concept, but the converse is alive and well. “I’m not going to use XXX again because their Web site is never available.”

The Web presence does not have to provide commercial services. If an information-only site is not available, it affects your reputation. A reputation for unavailable systems is a definite way to lose customers.

An organization as a whole may not depend on certain systems but those systems can still be critical. In some cases, these systems may be critical to people’s lives:

- Health care has specific needs—Patient monitoring systems need to be constantly available, down time on X-ray storage and processing systems can affect patient care, access to patient records can be required on a 24×7 basis
- Manufacturing organizations have critical systems—Chemical plant monitoring and control, print system control, and automated manufacturing systems

In addition, you can consider systems as diverse as

- Stock exchange trading systems
- Air traffic control
- Traffic management
- Automated railway systems
- Door access and other security systems

In all these examples, you need your applications and data to be available. The cost of it not being available can scale from the purely financial to the loss of human life. Organizations need systems to be available, but do they need all their systems all the time?

Degrees of High Availability

Talk to the business, and they will tell you that all their applications are business critical and have to be available 24×7. In many cases, this is simply not true. Many, if not most, organizations have a number of applications for which they require high availability, but it is rare that an organization requires everything to fall into this category.

There are several actions you can perform to boost the availability of your systems even if you are not supplying full high availability. Starting with the hardware you use for your servers, you can increase the availability by ensuring you build resiliency into the server by specifying:

- Fault-tolerant memory
- Redundant power supplies and fans
- Disk resiliency by using RAID or a SAN
- Multiple disk controllers
- Multiple network cards configured for fault tolerance

You can ensure that operating systems (OSs) are installed and configured to best practice. Using an automated build system ensures a repeatable and consistent configuration. Applications should be installed and configured correctly.

Basic good practice will deliver a system that will perform well and provide a reasonable level of availability. How can you measure that availability? One measure that is often used is the percentage of time available as Table 1.1 shows. This can also be expressed as a number of "nines" (for example, four nines means a system is 99.99 percent available).

%Availability	Downtime per year
90.0	36.5 days
95.0	18.25 days
99.0	3.65 days
99.9	8.76 hours
99.95	4.38 hours
99.99	52.6 minutes
99.999	5.26 minutes

Table 1.1: Percentage of time available.

The one thing that really leaps out of this table is how little downtime is allowed at the higher end of availability. Five and a quarter minutes is hardly enough time to reboot a server!

These figures should apply to unplanned downtime. A well-written SLA should include maintenance windows, otherwise the system will become unstable. If you have an availability target of 99.99% that allows 52.6 minutes downtime per year. Can you perform the monthly patching cycle in that time? Not if you have to reboot once a month. Applying service packs is an even bigger outage, especially on applications such as SQL Server or Exchange. Arrange the downtime to ensure that the system is maintained. Make sure that planned downtime is communicated so that the user population is aware of what is happening and when.

Pay for What You Get

High availability implies quality in a number of areas. This is especially true in the quality of the equipment that is purchased. I once worked in an organization where the network equipment was so old that our network administrators were scouring computer fairs for second (third or fourth) hand parts to keep it working. Did we have a high-availability environment? In a word, No!

Buy from recognized vendors who are known to build good servers, network switches, or whatever it is you need. Some high-availability options such as clustering in Windows Server 2003 require that hardware comes from a restricted list where the combinations of server and storage have been tested together.

Make sure that the quality aspects extend to the small parts such as cables when specifying, and buying, equipment. Spending thousands of dollars on servers and storage only to see the system fail because an attempt was made to save a few dollars on cabling doesn't make sense. One area to avoid skimping on quality is backup media. I mentioned earlier that a good backup is your ultimate recovery option. This option will not be reliable with poor-quality media that cannot be relied upon when attempting to restore.

Having considered why you need high availability, you must determine how you are actually going to deliver it.

Delivering High Availability

This topic is considered in more detail in Chapter 3. At this stage, we need an overview of the options before we turn our discussion to disaster recovery.

Consider Whole Infrastructure

High availability is not a bolt-on piece of technology. You cannot take your existing infrastructure, add a cluster to support an application, and claim that your environment provides high availability. What you have created is a clustered application. If you want true high availability, you need to consider the whole environment: network, applications, data, infrastructure, and servers.

You need to design high availability into the environment. As an example, consider the configuration that Figure 1.4 shows for a Web-based application. The Internet-based user connects through a firewall to the Web server in the DMZ. A connection through the internal firewall enables the Web server to talk to the database server.

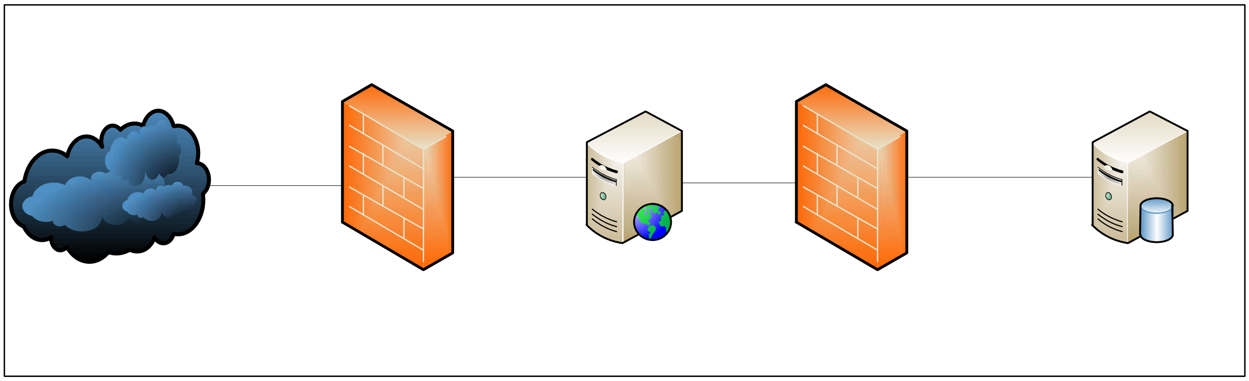

Figure 1.4: Standard infrastructure. Network links removed for clarity.

This configuration contains a number of single points of failure: Internet links, firewalls, network routers and switches, and a database server. If you remove all the single points of failure, you end up with an infrastructure that looks like Figure 1.5.

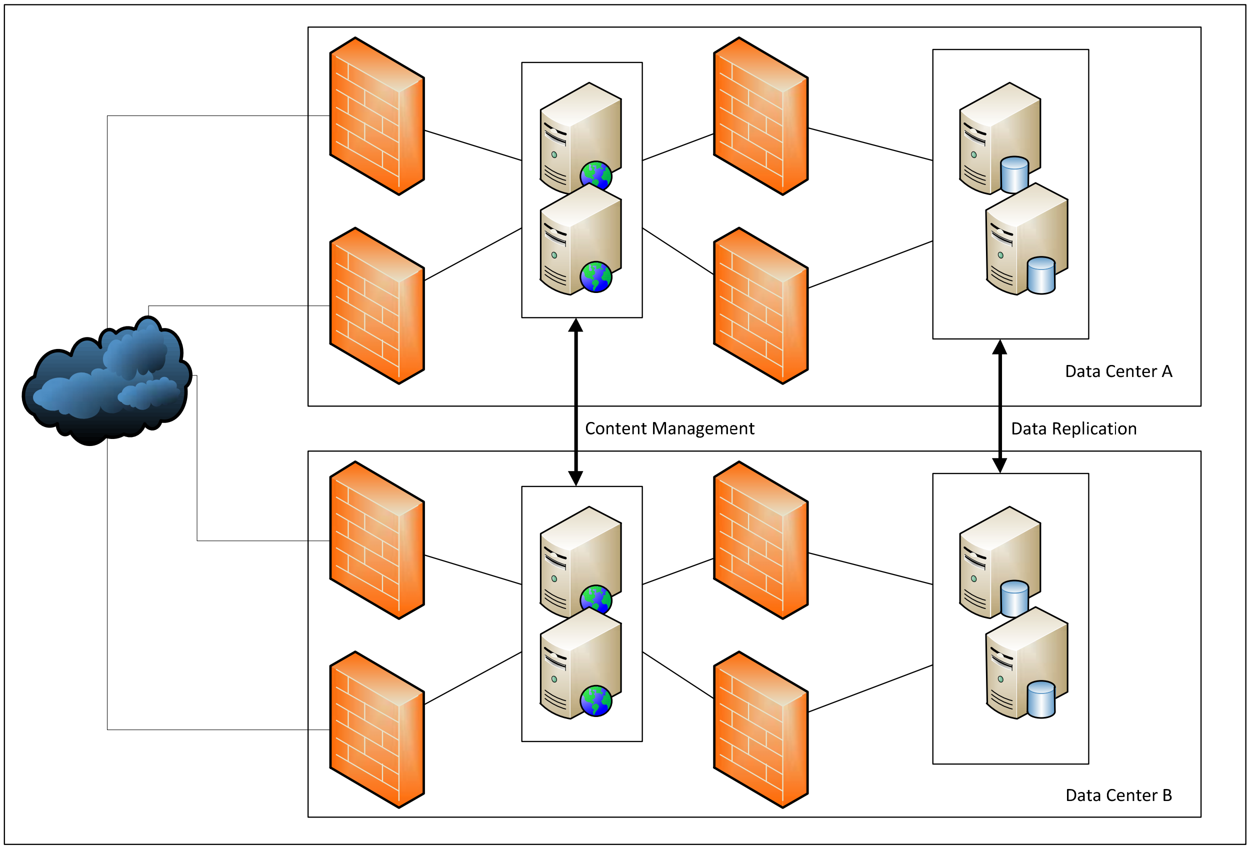

Figure 1.5: Environment designed for high availability. Network links removed for clarity.

There is redundancy at all levels of the environment including:

- Dual data centers
- Dual ISP links
- Redundant firewalls
- Redundant network routes and equipment
- Clusters and application-based high-availability techniques to protect the application and the data

Creating an environment configured in this way will cost a lot of money. It is also easier to implement in a green field site. However, if an application is important enough to the business, an existing infrastructure can be converted to this level of high availability. This is not the type of project to which a “big bang” approach should be applied. Approach it a step at a time and test each change thoroughly.

What sort of technologies can you use to meet your high-availability needs? We will look at these areas in greater detail in Chapter 3, but a quick overview is in order before we move on to disaster recovery.

Native Solutions

The two high-availability solutions that are native to Windows are Clustering and Network Load Balancing (NLB). Clustering provides high availability only while NLB also provides a method of balancing the workload across multiple machines. The obvious question is “Why can’t we use NLB all the time and forget clustering?” NLB only balances workloads that are TCP based, so, for instance, you can balance HTTP traffic to a Web farm but you cannot balance traffic to a database on SQL Server.

Clustering involves two or more machines configured with access to shared storage. A work load such as SQL Server, Exchange, or even file and print is installed on the cluster. The workload is hosted on one node of the cluster and in the event of a failure, it will fail over to the alternative node.

Most organizations configure two-node clusters. However, this is an expensive option, as one of the nodes is always in a passive mode waiting for the failover to occur. A better option is to use multi-node clusters with only one or two nodes available for failover. One system I worked on implemented four-node clusters for Exchange. Three were active with the fourth being passive. We implemented two of those. Instead of 12 servers for a “traditional” set of two-node clusters, we only used eight. That’s a significant cost saving in hardware, licenses, and administration.

NLB is often used for Web-based applications. It makes a number of servers look like one as the NLB cluster has a single IP address to which traffic is addressed. The cluster then manages the distribution of workload across its members. If one server fails, the others can redistribute the work. These systems protect the service but don’t necessarily protect the data.

Data-Based Solutions

Data is essential to business processes, as we saw earlier. There is no point in ensuring that your applications are available if the underlying data is unavailable. The basic methods to protect the data involve some type of replication or mirroring. You can replicate the data between storage systems. This can be performed by:

- The storage system, usually at the disk block level
- The application, usually at the database level

Storage-based replication will produce a second copy of the data on another set of disks. The replication target is usually in another data center so that high-availability and disaster recovery needs can be satisfied with a single solution. These techniques can add significant extra cost to the storage solution. If storage-based replication is used to protect databases, test the transaction consistency to ensure that the data doesn't end up being corrupted if failover occurs during a transaction.

It is possible to replicate data via the database. This involves configuring a system to copy the data at a transaction, or even database, level. The target database may not be available while replication is occurring. Failover is not automatic, but it is a simple process. Make sure the failover is practiced before it is needed.

Mirroring is similar in that there is a copy of the data on another system. The mirroring process can be configured to write to both copies of the data simultaneously, though this may affect the time taken to complete a transaction. Failover between the systems can be automatic with client applications being transparent to the change. Some applications supply their own level of high availability.

Application-Level High Availability

Some applications, such as AD and DNS, have an inherent level of high availability due to their distributed nature. Installing at least two domain controllers at each location protects the authentication, authorization, and name resolution services. If one of the on-site domain controllers should fail, the other will absorb the load. The services will still be available even if both fail because of the way AD will direct users to other domain controllers. Data is automatically protected because it is replicated between domain controllers.

The ultimate in protecting the application and data is to totally synchronize two systems. This technique involves two computers, the applications and data, being synchronized such that they are kept in lock step. The machines are presented to the clients as a single system and all changes occur on both. In the event of one system failing, the other still serves the applications to the clients. It is possible for the two systems to be in separate data centers to supply a disaster recovery solution as well as high availability.

Monitoring

We have quickly surveyed a number of techniques for supplying high availability for your systems. Whichever techniques are applied, it is not possible to implement the solution and forget about it. Systems have to be monitored. It is essential that faults are noted and rectified. Consider a two-node cluster as an example. If one node fails, the application will failover onto the second node. However, you have now lost your high availability. If the systems aren't monitored, you might not discover the failure until the second node fails. That is an embarrassment you can definitely live without.

The second use of monitoring is to determine capacity trends. Disks fill up, memory becomes a bottleneck, and CPUs become overloaded. All of these scenarios lead to a reduction in response time and unhappy users and give the system a reputation for unavailability. Monitor the trends in resource usage so that extra resources can be made available or upgrades can occur before the situation becomes critical.

Summary: Making High Availability Work for You

Creating environments with the correct high availability is hard work and can be expensive. There is a four-step process to creating and maintaining a high-availability environment:

- Identify needs. Make sure they are based on protecting business processes. The data and the service need to be protected.
- Design a solution. We have briefly covered the techniques available to you. Choose the most appropriate one to meet your organization's needs. Don't forget that virtualized servers need protecting as well.
- Implement the solution. This includes training for the people administering the system and the adoption of processes that augment the high-availability technologies.
- Monitor the systems. Early detection of failures or potential capacity issues means they can be dealt with before the system becomes unavailable.

High availability is not a luxury in the modern business world. It is a business imperative and the time to start is now.

Having gained an insight into high availability, it is now time to look at disaster recovery. The next chapter will concentrate on untangling the differences between high availability and disaster recovery as well as spend time considering where the activities are converging. In some cases, your high-availability solution also supplies a disaster recovery capability that it would be foolish to ignore.

Download Additional Books from Realtime Nexus!

Realtime Nexus—The Digital Library provides world-class expert resources that IT professionals depend on to learn about the newest technologies. If you found this book to be informative, we encourage you to download more of our industry-leading technology books and video guides at Realtime Nexus. Please visit http://nexus.realtimepublishers.com.