White Paper

Providing Data and Application Availability Management for IBM BladeCenter

Introducing LifeKeeper™ from SteelEye Technology®

Authored by Bonni-Jo B. Salazar and Dwain Sims
October 2004

Abstract

By combining IBM @server® BladeCenter™ Servers and SteelEye LifeKeeper High Availability solutions, businesses can effectively manage infrastructures to maximize productivity and minimize costs.

This paper presents an overview of LifeKeeper solutions running on BladeCenter servers, including sample configurations.

Table of Contents

Introduction 3
Cost of Downtime 3
High Availability Clustering as a Solution 4
IBM eServer BladeCenter Servers 5
Protecting BladeCenter Servers with LifeKeeper 6
LifeKeeperArchitecture 6
Resource Driven Clusters 6
Virtualization 9
I/O Fencing 9
Cluster Configurations 10
Local Recovery 11
Custom Application Protection 11
Management 12
BladeCenters in HA Clusters 13
Communications 13
Storage Options 13
Sample Configurations 15
Conclusion 26
References 27

Introduction

Business data is the lifeblood of your company and assuring it is always available is the most important mission of your IT organization. System failures, human error, power outages and natural disasters put business critical information at risk of being inaccessible for extended periods of time or worse, permanently lost. You need solid protection against crippling data loss and application unavailability.

Continuous availability is no longer an ideal; it is a necessity. Longer work days, expansion into new markets and customers demand for more efficient services create an expanded requirement for increased system availability. Users are demanding a means of ensuring very high availability of their applications and access to data that permits them to accomplish their tasks and provide the highest levels of customer service. Interruption of workflow due to system failure is expensive and it can cause the loss of business. The need to increase computer system availability is becoming a key business concern. Increasingly there are solutions on the market today that address these concerns. A high availability software solution combined with the technology of Blade servers provides a reliable, scalable, and manageable solution that offers an alternative to traditional rack mount servers. The architecture of Blade servers are less complex, easier to install, need fewer people to keep up and cost less than a traditional server solution, which reduces the overall cost.

The Cost of Downtime

Today's businesses and customers require high-availability solutions (on demand computing) across the board and at an affordable price. A global business needs 24-hour access to information 365 days a year. In an Internet service model, organizations must anticipate customers arriving at their Web site and business partners interacting with their systems at any hour of any day. For many businesses, "regular business hours" have no meaning.

The cost of downtime, whether unplanned or scheduled can have substantial negative revenue impact both in terms of immediately lost business and productivity, as well as the subsequent effect of a potential loss of customer loyalty and confidence.

Business Operation	Average Cost per Hour of Downtime
Communications: Converged Services	>$10.0 million
Financial: Brokerage Operations	$6.45 million
Financial: Credit Card/Sales Authorization	$2.6 million
Media: Pay Per View	$150,000
Retail: Merchandise Sales	$140,000
Transportation: Airline Ticketing	$89,500
Media: Event Ticket Sales	$69,000

Source: Gartner, Dataquest, Contingency Planning Research and Others

High Availability Clustering as a Solution

There is a solution. By combining IBM eServer BladeCenter servers running Linux or Windows with SteelEye LifeKeeper, businesses can achieve between 99.99% and 99.999% uptime for business critical applications. Thus, you can plan on between just 8 to 55 minutes of downtime for both planned and unplanned outages for an entire year. And this is for everything - from your mail server to your business critical financial management or manufacturing systems.

SteelEye LifeKeeper is a software application that ensures the continuous availability of applications by maintaining complete system uptime. LifeKeeper maintains the high availability of clustered systems by monitoring system and application health, maintaining client connectivity and providing uninterrupted data.

To enable automatic system and application recovery if the system goes down, LifeKeeper allows applications to failover to other servers, up to 32, in the cluster. This helps LifeKeeper eliminate the risk of a single point of failure and allows systems to meet the stringent availability requirements of mission-critical operations by creating a fault resilient environment

With LifeKeeper, hardware component or application faults are detected in advance of a full system failure through multiple fault-detection mechanisms. LifeKeeper monitors clusters using intelligent processes and multiple heartbeats. By sending redundant signals between server nodes to determine system and application health, LifeKeeper confirms a system's status before taking action. This reduces the risk of a single point of failure and minimizes false failovers. LifeKeeper also limits unnecessary failovers by recovering failed applications, without a full failover to another server, if the hardware is still active.

If an event creates an interruption in a server's availability, LifeKeeper automatically moves the protected resources and applications to another server in the cluster. Because this switchover is transparent to clients, and assures that a system failure does not impact users' productivity. LifeKeeper migrates all application and transfer connectivity in such a way that clients have continuous access to applications and data. This ensures that all clients - from internal users to customers shopping online - are not affected by unanticipated system failures.

LifeKeeper provides a cluster framework to allow the number of users supported by an application to be increased by simply adding nodes into the cluster. To ensure protection from failures, LifeKeeper also supports scalability at the application level. When LifeKeeper is installed with a multi-directional configuration, applications that are running on one machine can be broken up and failed over to separate machines.

LifeKeeper provides protection for Linux and Windows environments to support disaster tolerance, multiple system failures or faster recovery.

SteelEye offers LifeKeeper Application Recovery Kits for packaged software, including databases, Web servers and application servers. These Application Recovery Kits include tools and utilities that allow LifeKeeper to manage and control a specific application. When an Application Recovery Kit is installed for a specific application, LifeKeeper is able to monitor the health of the application and automatically recover the application if it fails.

IBM eServer BladeCenter Servers

BladeCenter Servers are intended to address the most serious IT issues: space constraints, manageability, scalability, capacity, performance, cooling and power. Its design collects resources into high-density enclosures that support hot-swappable, high-performance 2-way and 4-way Intel processor-based servers. BladeCenter offers the high performance and manageability of IBM rack-optimized platforms.

BladeCenter collapses the data center by integrating functions such as Layer 2-7 Ethernet and your Storage Area Network (SAN) fabric into a 7U enclosure that simplifies deployment and management.

Your enterprise or network can benefit from simplified management, fast installation, modular scalability and high availability. And BladeCenter delivers improved space efficiency compared to most 1U solutions.

The Standby Capacity on Demand offering features a customizable BladeCenter system with a mix of active and standby capacity blades.

BladeCenter solutions make adding capacity simple and affordable. BladeCenter's technology features deliver an effective scale-out architecture that lets you add server modules quickly using a "pay as you grow" approach.

BladeCenter efficiently uses data center floor space with up to 84 2-way blades or up to 42 4-way blades in a 42U rack. The design features leading-edge cooling technology and Intel Xeon™, Intel Xeon Processor MP and PowerPC® processors.

BladeCenter and BladeCenter T chassis features, such as high-availability midplanes and redundant hot-swap cooling and power, help reduce single points of failure. This is part of the OnForever™ features – designed to deliver outstanding operation, helping to increase productivity. Tight integration of key components such as networking services, centralized management, and applications, help to enable high availability.

BladeCenter architecture is based on industry standards to support deployment of third-party software and hardware technologies. IBM works with industry-leading technology

companies to support innovative solutions running on Linux and Windows® and Novell operating systems.

Protecting BladeCenter Servers with LifeKeeper

LifeKeeper Architecture

LifeKeeper is one of the most mature high availability clustering systems available today. Originally created by AT&T Bell Labs in the early 1990's, LifeKeeper's first mission was to protect phone switches. LifeKeeper was the first commercially available high availability clustering system for Intel based computers.

In many ways LifeKeeper's original creators broke with the conventions of that day. Until LifeKeeper came along, most clustering systems were based on Quorum nodes or Quorum devices. These systems were based on the concept that clusters of machines were formed around a quorum device or node, and that all members of the cluster were agreeing to the actions to be taken by the cluster. LifeKeeper broke with this tradition with its Resource Driven Architecture.

Resource Driven Clusters

In a resource driven clustering system like LifeKeeper, there is no central control of the cluster. The actions taken on a resource in the cluster are determined by the cluster member who currently "owns" the resource. Cluster members who own a resource and determine a need to take action on a resource do not need permission by other members of the cluster. This type of clustering system tends to scale much better due to the fact that multiple resources are common, and that multiple owners in the cluster can take independent actions in the case of a failure. Also a resource driven cluster can form multiple sub clusters in the case of communications failures. Resource driven clusters can also contend more easily with new storage models. Data Replication and Network Attached Storage are much tougher concepts to fit into a Quorum cluster model. (i.e. it is hard to build a cluster when you have no quorum storage device.) And quorum based clusters have a significant single point of failure – the quorum device itself. If for some reason the quorum device is unavailable, a quorum based cluster is out of commission, even if some or all of its other resources are available. Resource driven clusters like LifeKeeper do not have such a limitation.

Resource driven clusters are by definition peer-to-peer clustering systems. There is no centralized control of the cluster and no particular cluster member is in charge. In LifeKeeper the cluster configuration is replicated to all the nodes in the cluster, and there is no central place where cluster members must go to figure out what to do next. The cluster configuration exists on all of the members of the cluster.

Resource Hierarchies

Another easy extension of resource driven clusters is the concept of resource hierarchies. Resources can be linked together to form a group of resources that will make up an entire

application service. The hierarchy structure defines for LifeKeeper the order in which resources must be put into service for the entire solution to function.

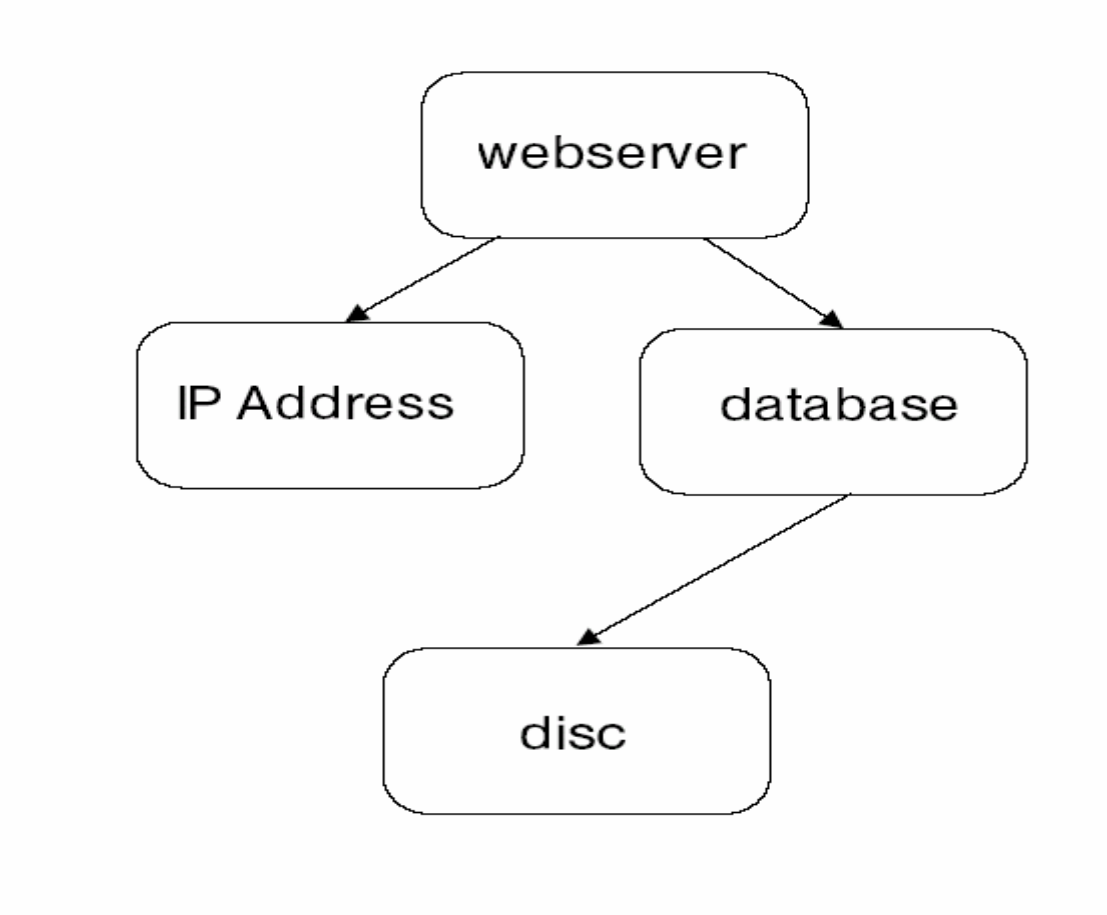


Diagram 1: An Application Resource Hierarchy

In diagram 1 there is a hierarchy of resources that define a web server application. The web server is dependent on the IP address (that network clients would attach to) and a database. Further, this database is dependent on a disk where its data is stored. Before this web server can be brought into active service, the disk has to be operational, the database system needs to be running, and an IP address has to be active. Once all of these dependent resources are operational, the web server resource can be started up.

Application Recovery Kits

In order to create resource hierarchies easily, SteelEye has made available a number of preconfigured Application Recovery Kits for common services and applications. These recovery kits are comprised of several parts. The first is the logic for starting and stopping the service or application. The second is the code for monitoring the resource, and also recovery code to be executed if the monitoring reports a failure. The recovery kit also defines the minimum set of dependent resources that service or application will need to operate properly. There are Java property files that are created to allow the recovery kit to be "plugged" into the LifeKeeper graphical user interface, also known as the LifeKeeper Availability Manager.

SteelEye supplies recovery kits for applications like DB2, Informix, Sybase, mySQL, SQL Server, Oracle, PostgreSQL, SAP, maxDB, ClearCase, Exchange, Sendmail and others. There are also recovery kits for common services like Samba, NFS, LVM, Apache, IIS, and LAMP.

There are basic building block recovery kits that are included with the LifeKeeper core product. These kits protect resources like IP addresses, File Systems, Raw disk partitions (for Database use), Volumes, LAN Manager Names, and Shares.

LifeKeeper also has the ability to allow custom applications to be protected by creating restore and remove scripts. This is the Generic Application Recovery Kit and it is part of the core product.

One of the very compelling benefits of the Application Recovery Kit technique is the ability to use "off the shelf" versions of the applications. The Application Recovery Kit provides a wrapper that deals with all of the logic needed for a particular application to exist in a LifeKeeper cluster. This also means that standard versions of the applications can be used. LifeKeeper recovery kits allow the workgroup versions of applications like DB2 and Oracle to be protected. The standard versions of Exchange and SQL Server can be protected by a LifeKeeper cluster. This can mean considerable savings when compared to the "Enterprise" or "Advanced" versions needed in other clustering systems.

Cluster Metadata

Since LifeKeeper is a resource driven clustering system, each node in a LifeKeeper cluster keeps a copy of the cluster's configuration locally. This configuration information is known as the cluster metadata. On a working cluster member, the cluster metadata is kept in a shared memory space for easy and fast access. LifeKeeper also has a utility available to make a backup copy of the cluster metadata, and to be able to restore it later in case of a problem.

Cluster Communications

In a resource driven clustering system it is important that cluster member remain in contact with one another. An important part of LifeKeeper's architecture is the LifeKeeper Communication Manager. This subsystem allows the administrator to create Communication Paths that allow the cluster members to remain in communication with each other. The Communication Paths can be created across TCP network links or through serial ports across a null modem cable. SteelEye strongly suggests that two Communication Paths exist between nodes in a production cluster.

*Heartbeat

The first of two major functions of Communication Paths are to pass Heartbeat information. The basic idea behind this is to confirm to other cluster members that a given node is up and still operational. If a node ceases to send heartbeats, the other nodes in the cluster will begin to recover any of the resources that the failed node has in service at that time. The number of missed heartbeats before the recovery process starts is configurable by the user.

*Cluster Configuration

The second function of a Communication Path is to exchange cluster metadata. During the process of creating a cluster, there is a lot of cluster metadata passed back and forth on the Communication Paths, and a relatively small amount once the cluster is in full operation.

Virtualization

Another concept that is important in a resource driven cluster is the idea of Virtualization. This really means that once a hierarchy of resources is created and extended through the cluster, it no longer belongs to a specific Blade server. The service can now be executed by any of the Blades in the cluster.

Application

At the top level is Application Virtualization. This means that the application is now under control of LifeKeeper. The application is no longer tied to a specific Blade server; it can run on any of the Blade servers that it has been extended to. Clients and Administrators will be able to work with the application as they always have, with the only exception being that it is no longer tied to a specific piece of hardware. This means that maintenance windows no longer have to occur at strange hours of the night. The application can be moved to a different blade (1 to 2 minute switchover time), and maintenance can be performed during normal business hours. Users can access their application as usual. Administrators and users will now find the application by its virtual IP address, and not necessarily on a particular Blade.

Networking

Closely related to the idea of Application Virtualization is Network Virtualization. Services under LifeKeeper protection are typically assigned a new, virtual IP address that can be asserted by any of the systems in the cluster where the application is put into service. Network based clients need to now attach to the service via the virtual IP resource that is part of the resource hierarchy for that application, not to a physical machine's IP address. These are regular IP addresses, and they normally have DNS names associated with them. But they are LifeKeeper resources, and they can be asserted by any Blade that is part of the cluster where that application might be in service.

Client Attachment

Network clients, either humans or other computer systems, now can attach to LifeKeeper protected services via the virtual IP address for that service. Though the service may be executed by any of the Blades in the cluster, they will always be able to find this particular service by its IP resource.

I/O Fencing

I/O Fencing is the concept of protecting disk resources from corruption. Resource driven clusters usually use different techniques than Quorum based clusters when providing this I/O Fencing. LifeKeeper uses slightly different techniques on its Linux and Windows releases.

Linux - SCSI Reservations

When Blade servers running Linux are attached to DS4000 based storage systems, individual LUNs are made available as useable storage. Each of these LUNs will show up in Linux as a SCSI disk. LifeKeeper's I/O fencing technique is to set a SCSI reservation in the DS4000's controller on a given LUN. Once that reservation has been set, the controller will no longer allow access by other systems. Other systems may be able to see that the LUN exists, but they will have no further access.

If LifeKeeper needs to move a disk resource from one Blade to another, the reservation can be dropped and then reasserted by the new Blade. In the case of a hard Blade failure, LifeKeeper will force the reservation to clear so that the new Blade where the resource will reside can bring it into service.

Windows- Filter Driver

When LifeKeeper was ported to Windows NT in the mid 1990s, SCSI Reservations were not accessible by the SCSI and Fibre Channel drivers built for Windows. A different type of I/O Fencing technique was needed. Windows has the capability to add "Filter Drivers" into the I/O stack of the operating system. The filter driver that the LifeKeeper team created provides for communication back to LifeKeeper to determine if the volume in question is being used by another node in the cluster. If LifeKeeper reports that it is being used, all access to that volume is blocked.

Using this technique, it is paramount that working Communication Paths be present between the blades in the LifeKeeper cluster. To this end, there are two additional Communication Path possibilities that should be explored when setting up a Windows based LifeKeeper Cluster. The first of these is a "disk communication path." The disk communication path is a tiny LUN that LifeKeeper running on two Blades can use to pass heartbeat and cluster metadata. This feature is only available in the Windows version of LifeKeeper, and it is ideal for small clusters.

Another, more scaleable technique is also available. When using the Fibre Channel Expansion cards in a Blade server to talk to DS4000 storage, a new communication possibility exists. TCP drivers exist for Windows 2000 and Windows Server 2003, thus allowing a normal LifeKeeper Communication Path to be created across the Fibre Channel fabric. Using this technique, all of the members of the LifeKeeper cluster can communicate across the same interfaces and paths that access to the critical data is occurring on. In combination with Communication Paths running on the normal network interfaces of the Blades, communication across the SAN provides an extra layer of security to insure data integrity in the cluster.

Cluster Configurations

LifeKeeper offers many different ways to configure Application Resource Hierarchies.

Active / Passive

This is the classic cluster configuration, with resources active on one Blade, and a second Blade waiting to take over in case of a failure. The Blades could be in the same

BladeCenter, or in different BladeCenters. Active / Passive will work with all storage options.

Active / Active

In this configuration, both Blades are doing useful work. One might be running SQL Server, the other running Exchange. Each Blade is being the backup for the other. In the event of a failure, both services will run on the same Blade. The only real caveat with this type of configuration is that you have enough capacity on each Blade to run both applications adequately. This configuration will work with any storage option.

N+1

This configuration is an adaptation of Active / Passive. N+1 allows there to be "N" number of Blades doing useful work, and a single Blade that is backing up all of the other Blades. This can be a very economical way to provide good failover capabilities. Alternate configurations like N+2 can also be constructed. Within limits, N+1 will work with all storage options.

Cascading Failover

Sometimes deeper levels of protection are needed. Clusters need to be created to allow for multiple failures. LifeKeeper allows resources to be extended to several different Blades, so that if more than one Blade were to fail, the resource would have some other place to be started. Shared or NAS based storage is usually required for Cascading Failover.

Local Recovery

The Application Recovery Kits available from SteelEye all include a feature known as "Local Recovery." LifeKeeper will monitor resources on the local Blade through the "QuickCheck" feature, which runs periodically, and if a failure is detected, a local restart of that resource will be attempted. It is almost always faster to do local restart rather than failover to another Blade. If the local recovery is unsuccessful, failover to another Blade will occur.

Custom Application Protection

It is often important for clustering systems to support new or custom applications; ones that need protection, but no existing Application Recovery Kit is available. LifeKeeper provides a couple of different solutions for these very common situations.

LifeKeeper SDK

SteelEye Technology provides a Software Developers Kit to customers who need to create their own recovery kits. The SDK provides documentation and sample code to developers to speed the creation of new recovery kits. The building of a new kit can be very useful to OEM and ISV customers who want to use LifeKeeper as the basis for

offering their highly available version of their applications. Also, having a full recovery kit available is useful for any larger scale deployments.

Generic Application Recovery Kit

In many situations, the Generic Application Recovery kit mechanism that is provided as part of the LifeKeeper core product is more than sufficient in providing support for new or custom applications. Using the "GenApp" feature, a developer need only specify a "Recover" script, a "Remove" script, and optionally a "QuickCheck" script. These scripts are typically written in either Kornshell or Perl scripting language. GenApp provides an easy way to extend the capabilities of a LifeKeeper cluster.

Management

Configuration, setup, and management are essential parts of any cluster deployment. LifeKeeper provides several tools to make these jobs easy.

LifeKeeper Availability Manager

The primary point of interaction with LifeKeeper for most people begins and ends with the LifeKeeper Availability Manger. This Java based graphical user interface allows easy system connection, Communication Path creation and monitoring, Resource creation and extension, and Resource Hierarchy management. The Availability Manager can be run as a standalone Java application on one of the clustered Blades, or it can be accessed via a web browser on another machine and run as a Java applet.

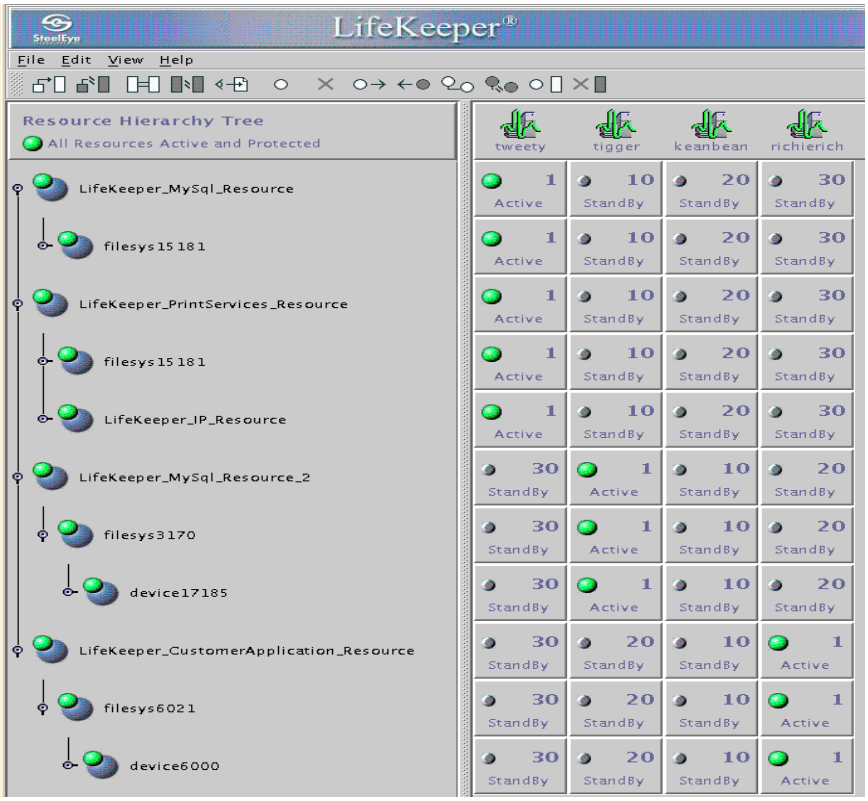


Diagram 2: The Availability Manager

The Availability Manager is wizard based. Most configurations are done as a series of easy questions, and default and dropdown lists are supplied wherever possible.

Once resources are created in the cluster, the Availability Manager allows these resources to be extended to other systems that belong to the cluster. Diagram 2 shows that a mySQL database resource hierarchy has been created on the server "tweety" and then extended to the servers "tigger," "keanbean," and "richierich."

SNMP Alerts

LifeKeeper can also be configured to send Simple Network Management Protocol traps anytime cluster state is changed. This allows systems like IBM Director or Tivoli to be alerted whenever there is activity with cluster resources.

BladeCenters in HA Clusters

The IBM BladeCenter and Blade Servers make an outstanding platform for deploying a LifeKeeper based High Availability cluster.

Communications

By using an internal bus style network configuration, individual Blades can be networked without the need for external switches or cabling. This means that all cluster communication can happen inside the BladeCenter. This is a very simple and clean configuration. The greatest enemy of a high availability cluster is complexity, so the internal networking inside BladeCenter provides a very stable platform for the creation of LifeKeeper clusters.

When clustering Blades in separate BladeCenters, the picture is nearly as rosy as the one with a single BladeCenter. Two Ethernet cables can be used to chain the BladeCenters together. The cables are used to tie together the internal Network switches of the BladeCenters. This allows cluster communication to occur through both sets of internal network switches, providing more redundancy.

Storage Options

There is a wide array of storage options available for the IBM eServer BladeCenter, and most all of these are ideal for LifeKeeper cluster storage.

SAN Based DS4000

These Fibre Channel based storage systems are perfect for LifeKeeper cluster configurations needing shared storage. Formerly know as the FAStT series of storage servers, they provide RAID style storage that is perfect for LifeKeeper.

*Single Path

Fibre Channel storage can be single or multipath configurations. For budget minded customers, a single path Fibre Channel configuration can be created with the BladeCenter. The BladeCenter can be configured with a single Fibre Channel switch, or the optical pass-through module can be installed for instances where an existing Fibre Channel switch is already in place. LifeKeeper will work with single path configurations without needing any additional software.

*Multipath

DS4000 storage utilizing a full multipath Fibre Channel configuration are among the most robust environments that can be created for a LifeKeeper high availability solution. All pieces of the storage solution have redundant backups. These configurations use two Fibre Channel adapters per Blade. There are either two BladeCenter based switches, or two external Fibre Channel switches with optical pass through modules in the BladeCenter, and two array controllers in the DS4000 unit. All of these components are cross cabled so that no single path through the system is critical for the solution to function. LifeKeeper will work fine in such an environment, but an additional piece of software is needed to manage the multiple paths. IBM Total Storage has made available a driver level software solution known as RDAC (Redundant Disk Array Controller) for both Linux and Windows that will manage all of the paths and insure that data flows freely in case of a failure of any of the Fibre Channel components.

NAS Based Storage for Linux

LifeKeeper for Linux and the BladeCenter servers work very nicely together in environments where Network Attached Storage is being utilized. Typically, storage will be attached to the Blades via their network interfaces using NFS (Network File System). SteelEye makes available optional software to use this type of storage, the Network Attached Storage Recovery Kit. The only caveat on Network Attached Storage is to make sure that the application that needs protection by LifeKeeper is compatible with Network Attached Storage (NFS). There are some applications that place special burdens on the file systems, and they will not work properly with NFS.

Data Replication

Another method for providing storage for a high availability cluster using BladeCenter servers is Data Replication. Using this technique, the data critical to the application being protected is confined to a particular disk partition or volume. Using SteelEye's LifeKeeper Data Replication product, the data is then replicated across a network to another partition or volume on a different Blade. The Blades could be in the same BladeCenter, or in different BladeCenters. Possibly even in a different BladeCenter in different data center, connected by a wide area network.

*LAN Based

When doing Data Replication between Blades in the same BladeCenter, or between Blades in different BladeCenters that are locally connected, LAN based data replication can be utilized. LAN based data replication is easy to configure and can give very high performance due to the high speed nature of the networking environment.

*Synchronous

LifeKeeper Data Replication can work in one of two modes, Synchronous or Asynchronous. When doing synchronous replication, as each block of the source partition (or volume in Windows) is written, the block is transported across the network interface to the remote Blade, and write occurs on the remote first. The acknowledgement of the write is then transmitted back to the source Blade, and then write is committed there. Synchronous replication insures a very high level of data integrity, but it can be relatively slow in performance. Synchronous replication is only suitable for a LAN environment.

*Asynchronous

LifeKeeper Data Replication can be set up in Asynchronous mode as well. When doing asynchronous data replication, as the source Blade writes a block to disk, the replicated block is transmitted across the net. But in this case the source goes ahead and commits the write before receiving the acknowledgement of the target Blade. Using a technique called intent logging, LifeKeeper Data Replication keeps track of blocks that have been committed, and insures that blocks on the target Blade are written in time order to insure file system integrity. Asynchronous data replication cannot guarantee that the data on the target Blade is exactly the same as that of the source Blade, however. For many situations, asynchronous replication is an acceptable tradeoff, when high performance of the source blade is paramount.

*WAN Based Asynchronous

When setting up a WAN connected cluster, often for Disaster Recovery purposes, the Asynchronous feature of LifeKeeper Data Replication is ideal. The higher latency of the WAN environment demands asynchronous replication.

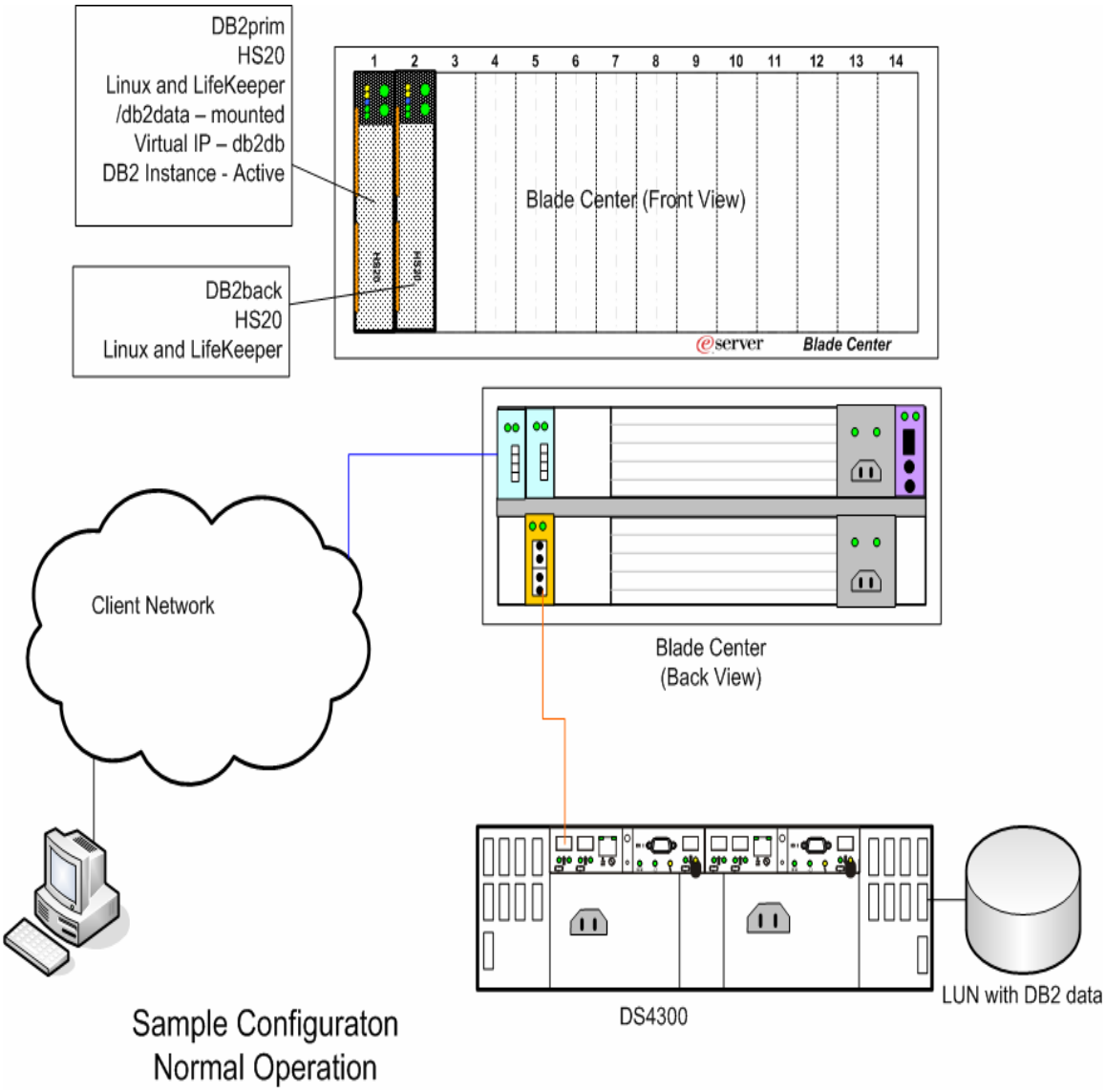
Sample Configurations

Let’s look at a sample configuration, and we will analyze what will happen in the case of a failure.

Our sample application will be a DB2 Database running on Linux. The primary server for the database will be a HS20 Blade in an eServer BladeCenter. The backup server will also be an HS20 Blade in the same BladeCenter. Local storage for both of these Blades will be ATA hard disks attached directly to the Blade. The storage for the DB2 database instance will be on a DS4300, configured in a single path mode, using a Fibre Channel switch that is internal to the BladeCenter.

Step by Step Failure Recovery

Normal Operations



Primary Blade
Name: DB2prim
IP Address: 192.168.45.10
Virtual IP Address: 192.168.45.20

Secondary Blade
Name: DB2back
IP Address: 192.168.45.15

The DB2 database is under LifeKeeper protection in this two node cluster. There has been a file system resource created, and this LUN (served up by the DS4300) is mounted on the mount point of /db2data. So in normal operations, this LUN is mounted on /db2data on the Blade DB2prim. There is also a LifeKeeper IP resource associated with the DB2 resource hierarchy, and it is given the DNS name db2db, and its address is 192.168.45.20. So in this example, all client connections to the DB2 database would take place on the interface called "db2db."

As you can see from the diagram, even though there is a lot going on here, it is a relatively simple configuration when using an IBM eServer BladeCenter. Only two cables (excluding power cords) are needed for this configuration; a Fibre Channel cable to link the DS4300 to the Fibre switch in the BladeCenter, and a network cable to attach the network switch in the BladeCenter to the public network.

Let's imagine a failure has taken place on DB2prim; its local ATA disk drive has died. Linux and the DB2 database running there come to a screeching halt!

1. LifeKeeper running on DB2back will notice that DB2prim has stopped sending heartbeats across the configured Communication paths. After missing a preconfigured number of heartbeats, LifeKeeper on DB2back will begin to take action.2. LifeKeeper will now begin to recover the resources that DB2prim was using. DB2back will clear the SCSI reservation that the DS4300 was holding on the LUN, and will proceed to mount the file system on the mount point /db2data.
3. LifeKeeper running on the Blade DB2back will now configure the virtual IP address on the 192.168.45 subnet. It will now respond to requests on the interface 192.168.45.20 by the name of "db2db."
4. LifeKeeper will begin to startup the DB2 database instance. Once it is up and running, it will respond to requests on the 192.168.45.20 ("db2db") IP address. Clients can begin reconnecting immediately.
5. Once the Blade DB2prim's disk is repaired (and data restored from backup), the Blade will reboot and will rejoin the cluster. DB2prim will now become the backup for the DB2 application, until an administrator moves the DB2 resource hierarchy back to the DB2prim Blade. This action can also be configured to happen automatically in LifeKeeper.

Other example configurations

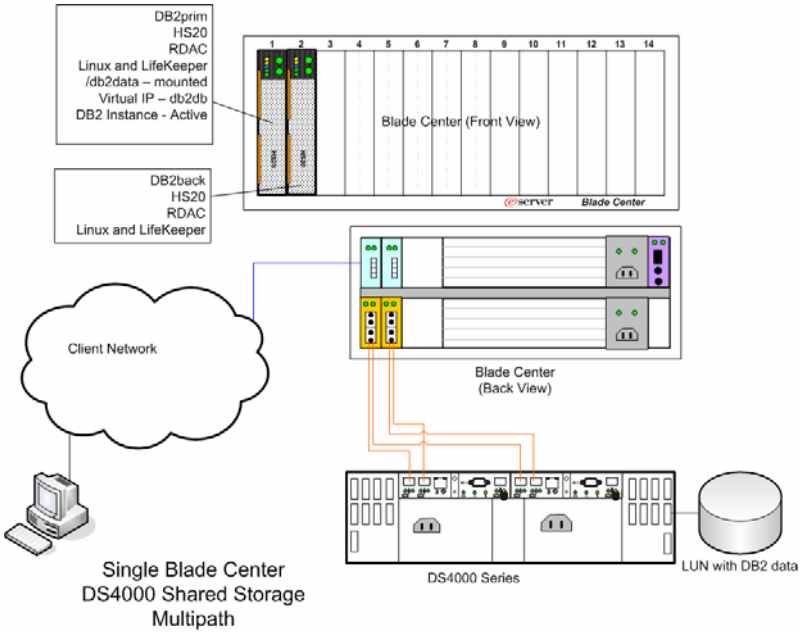
BladeCenter Scenarios

Here are some possible configurations using eServer BladeCenters and LifeKeeper. Most of these configurations can be used with Windows or Linux. These are just specific examples.

Within a single BladeCenter

These examples use a single BladeCenter. These configurations are very clean and simple, although perhaps not as intensely robust as a two BladeCenter configuration.

Using DS4000 based Storage

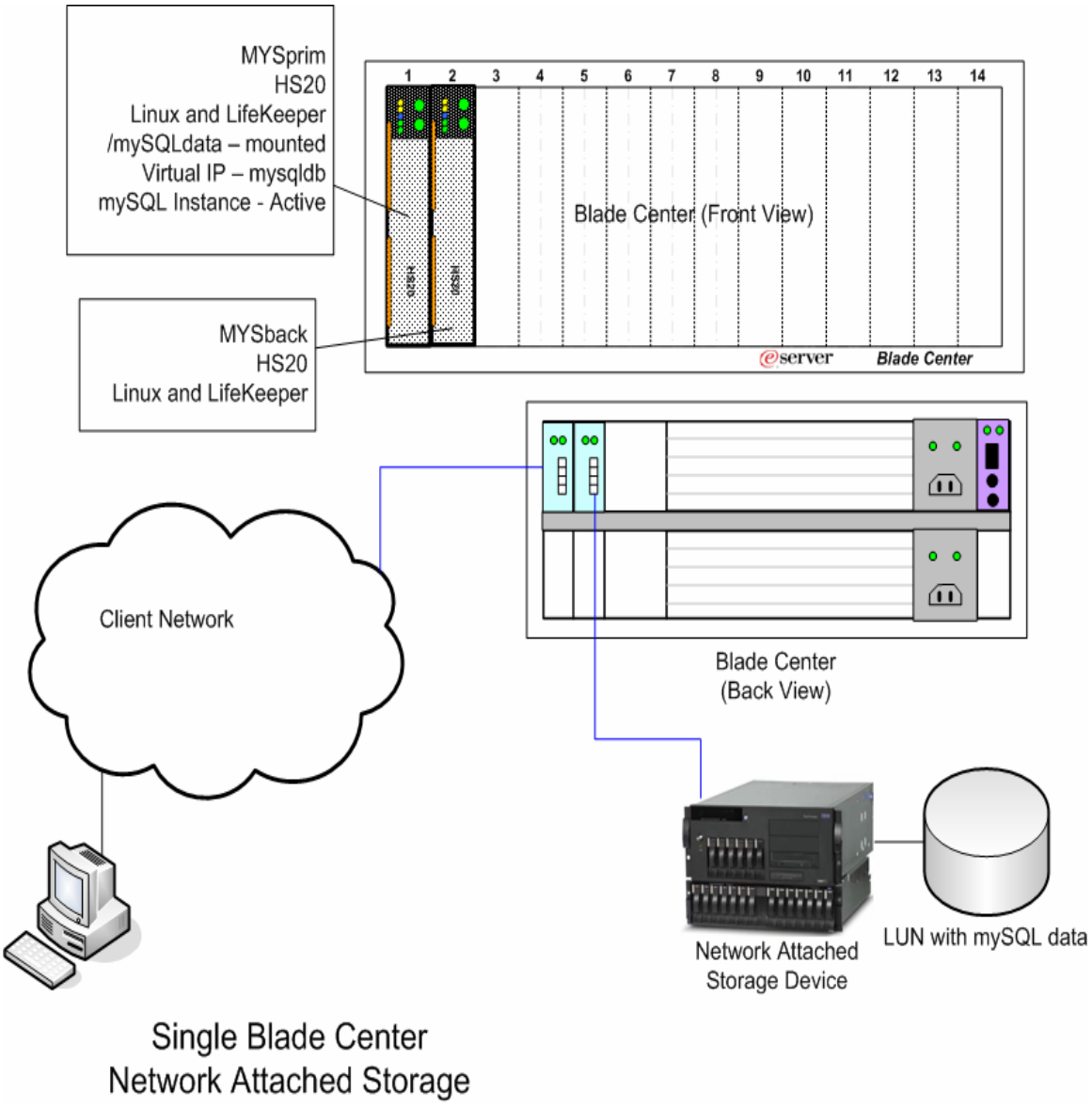


This configuration is an extension of our sample configuration that was discussed above. In this example a multipath configuration is used.

Even though this is a full multipath configuration, there are only five cables needed to set this up (excluding power cords and KVM cables). The primary Communication Path (heartbeat) would happen on the second Ethernet switch, and the second Communication Path would go across the Client Network, although all traffic would stay on the internal switch.

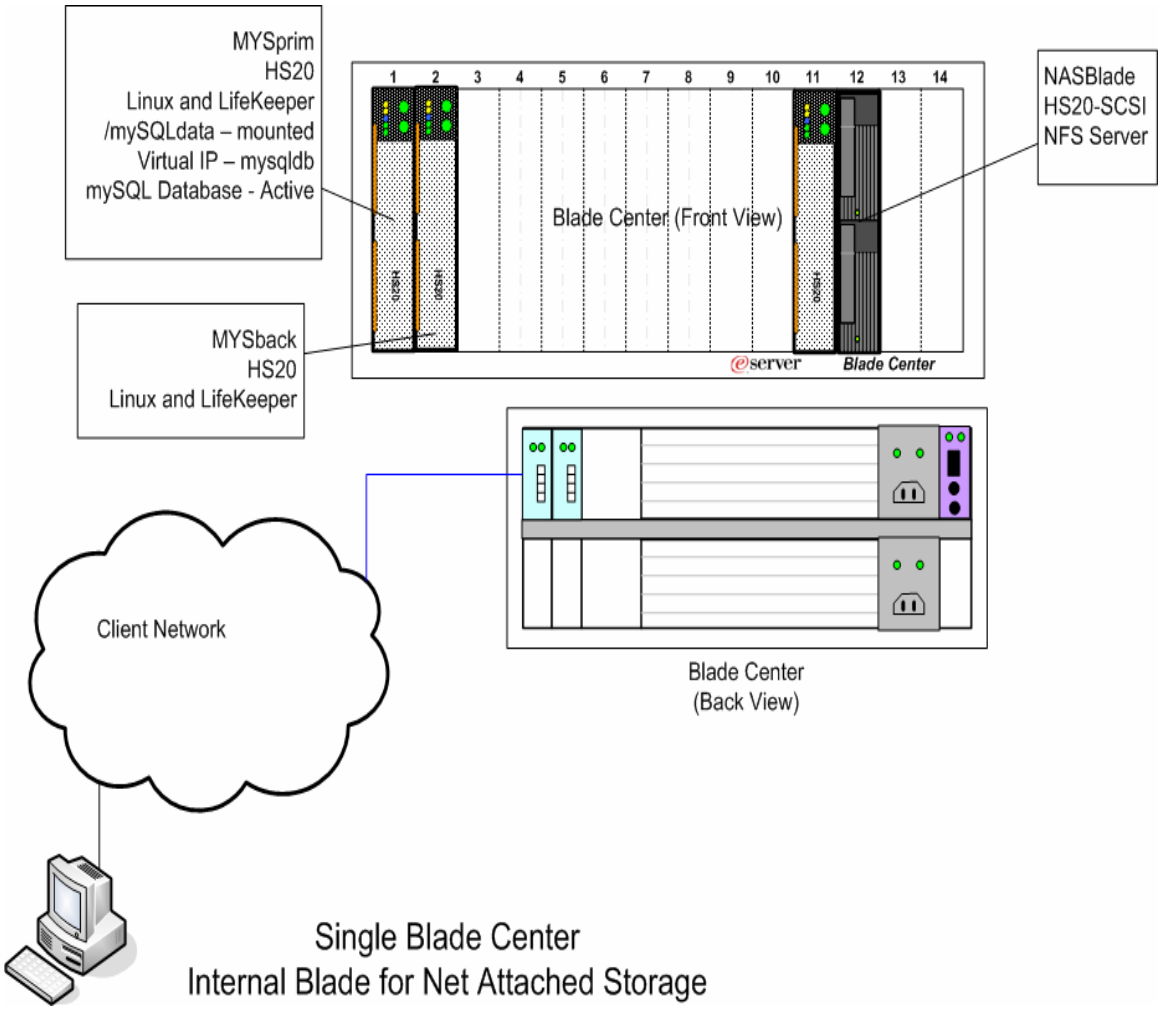
The Fibre Channel cables are cross connected between the internal switches of the BladeCenter and the dual ported controllers in the DS4000.

Using Network Attached Storage



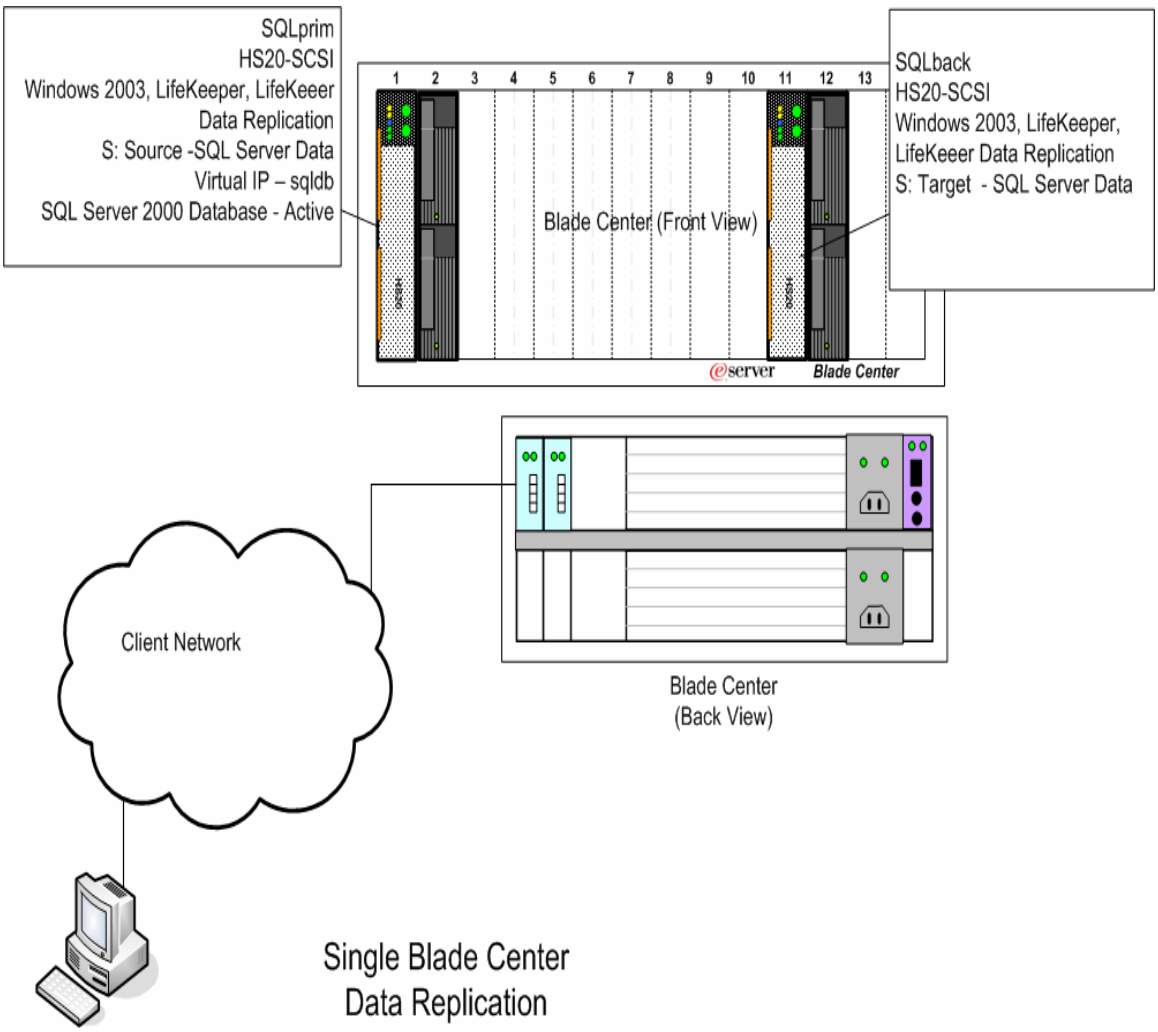
With Network Attached storage in a Linux configuration, the configuration is even cleaner. The primary LifeKeeper Communication Path occurs over the second switch in the BladeCenter, and the secondary LifeKeeper Communication Path happens on the client network. The Gigabit Ethernet connection through the second Ethernet switch is a very adequate connection using NFS to the NAS server. In this particular example, a mySQL database is being protected using the mySQL Application Recovery Kit.

Using Network Attached Storage – Alternate Configuration



Here also is an alternative NAS configuration. Using a HS-20 (SCSI version for higher performance) blade running Linux and configured as an NFS server, we can keep all of the traffic internal to the BladeCenter. The SCSI disks of the "NASblade" can be set up as RAID 1 to provide good data protection.

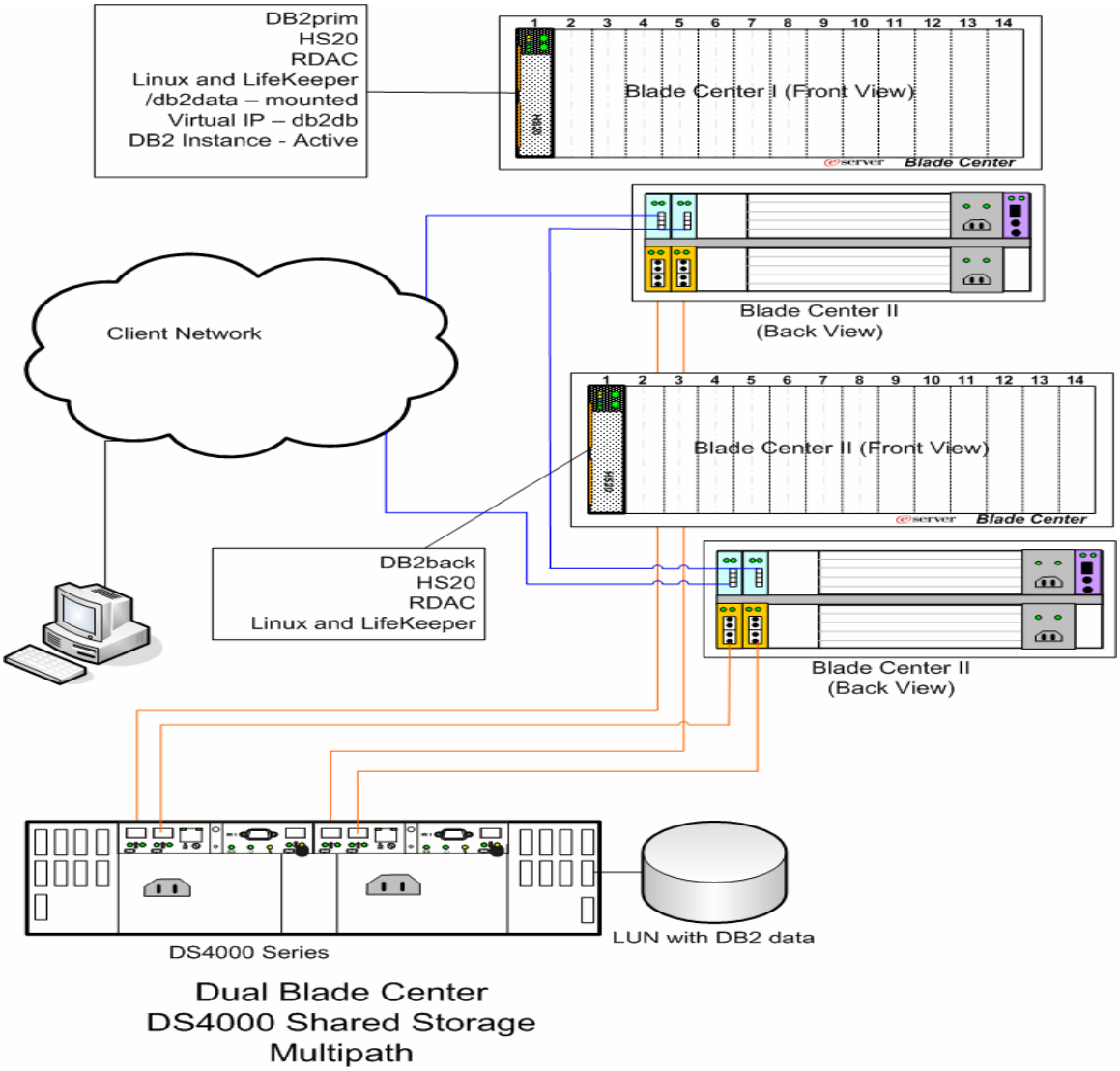
Using Data Replication



In this example Windows 2003 Server is the operating system. Two HS-20 (SCSI Version) Blades are used as the cluster servers, and LifeKeeper Data Replication is used to protect the critical data. The SCSI disks are not required, but they will give higher performance than the ATA drives that are in the normal HS-20 Blades. The primary LifeKeeper Communication Path occurs over the second switch in the BladeCenter, and the secondary LifeKeeper Communication Path happens on the client network. LifeKeeper Data Replication uses the second Ethernet switch as its replication path. This also is a very clean and simple configuration,

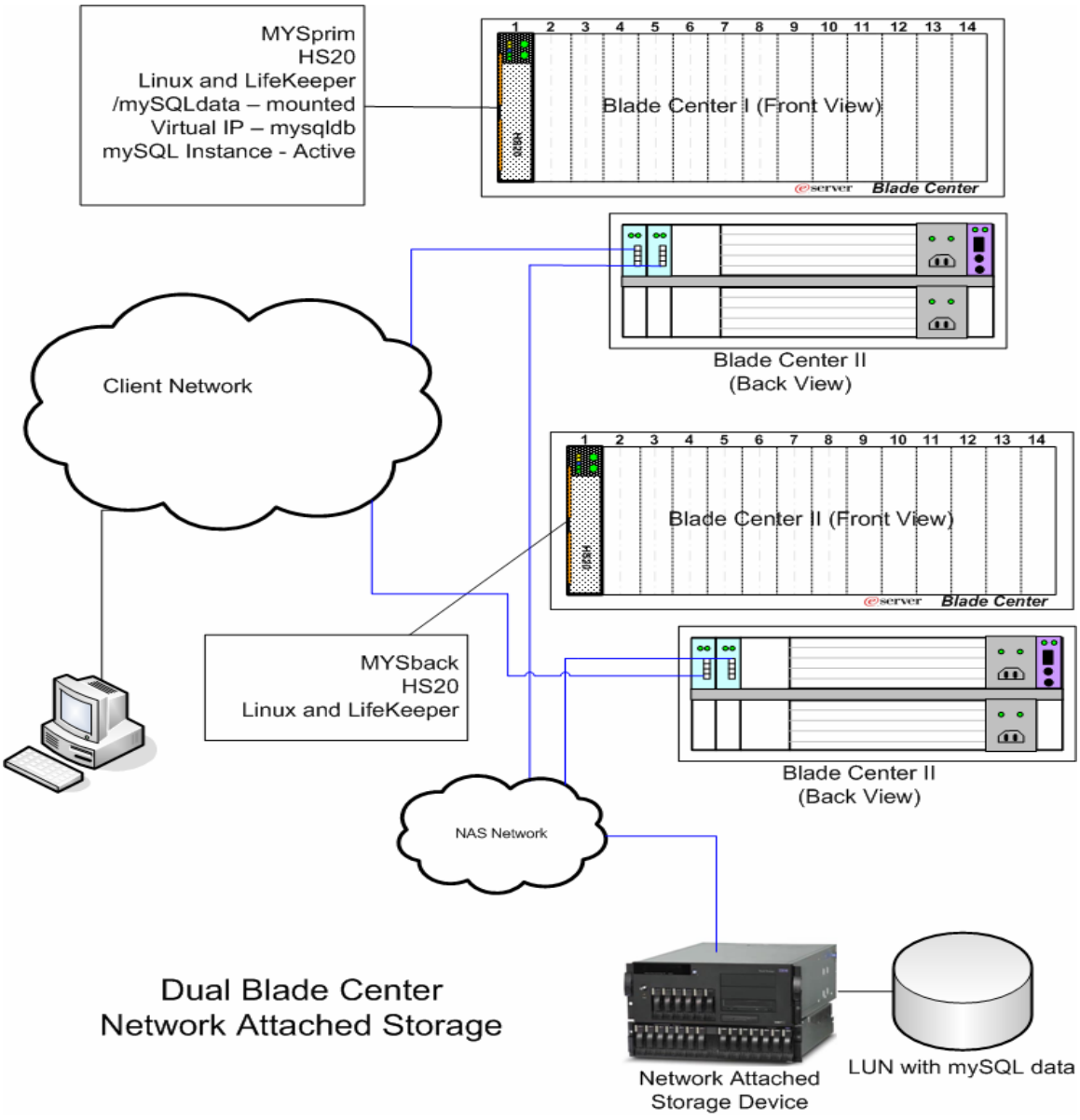
Between Blades in separate BladeCenters

Using DS4000 based Storage



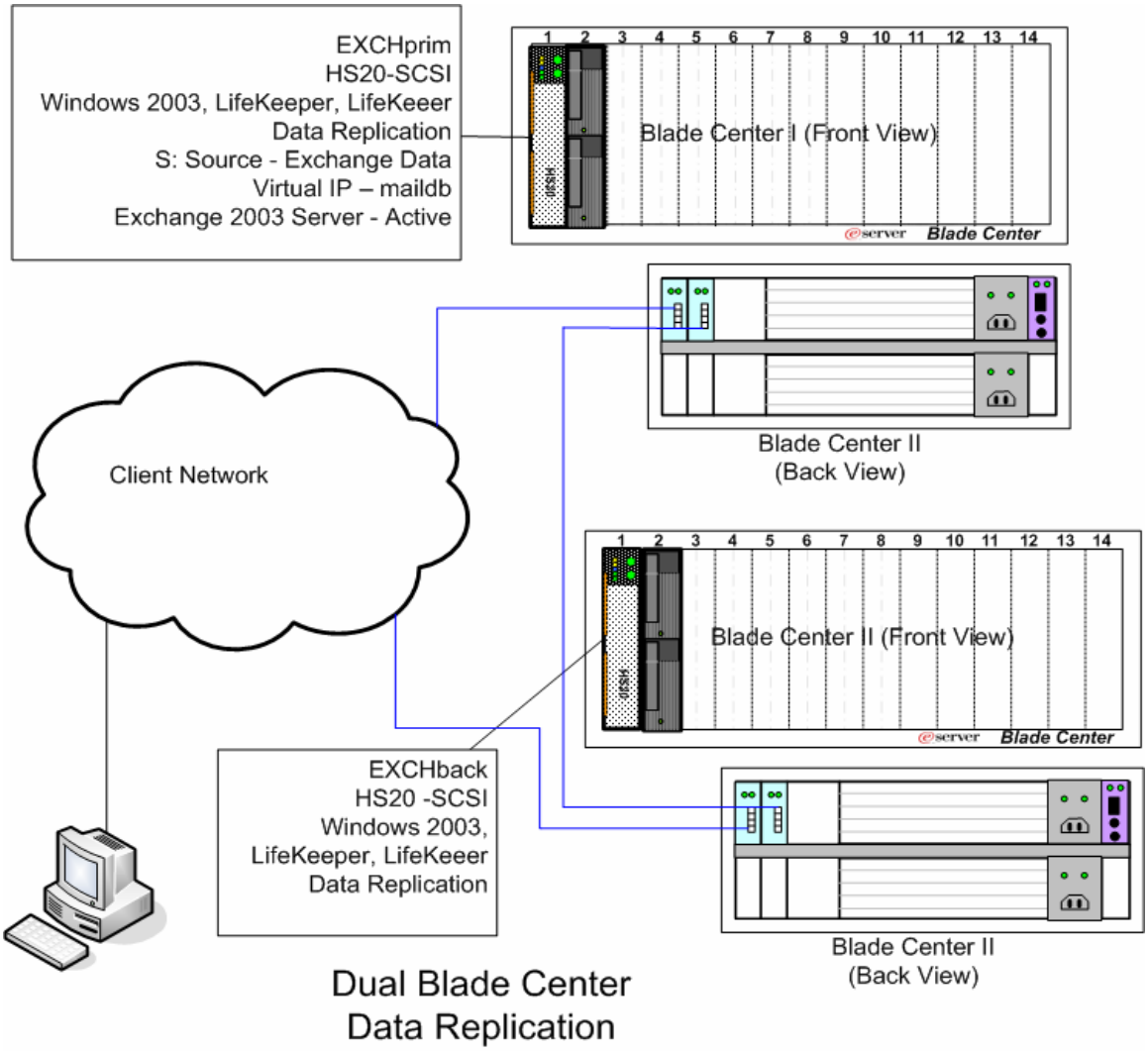
This configuration uses a single blade in two separate BladeCenters for maximum redundancy. The DS 4000 based storage is multipath connected to Blades through the internal Fibre Channel switches of the BladeCenter. This configuration is very expandable. More Blades can be added to serve other applications, or for building partitioned DB2 databases, all protected by LifeKeeper.

Using Network Attached Storage



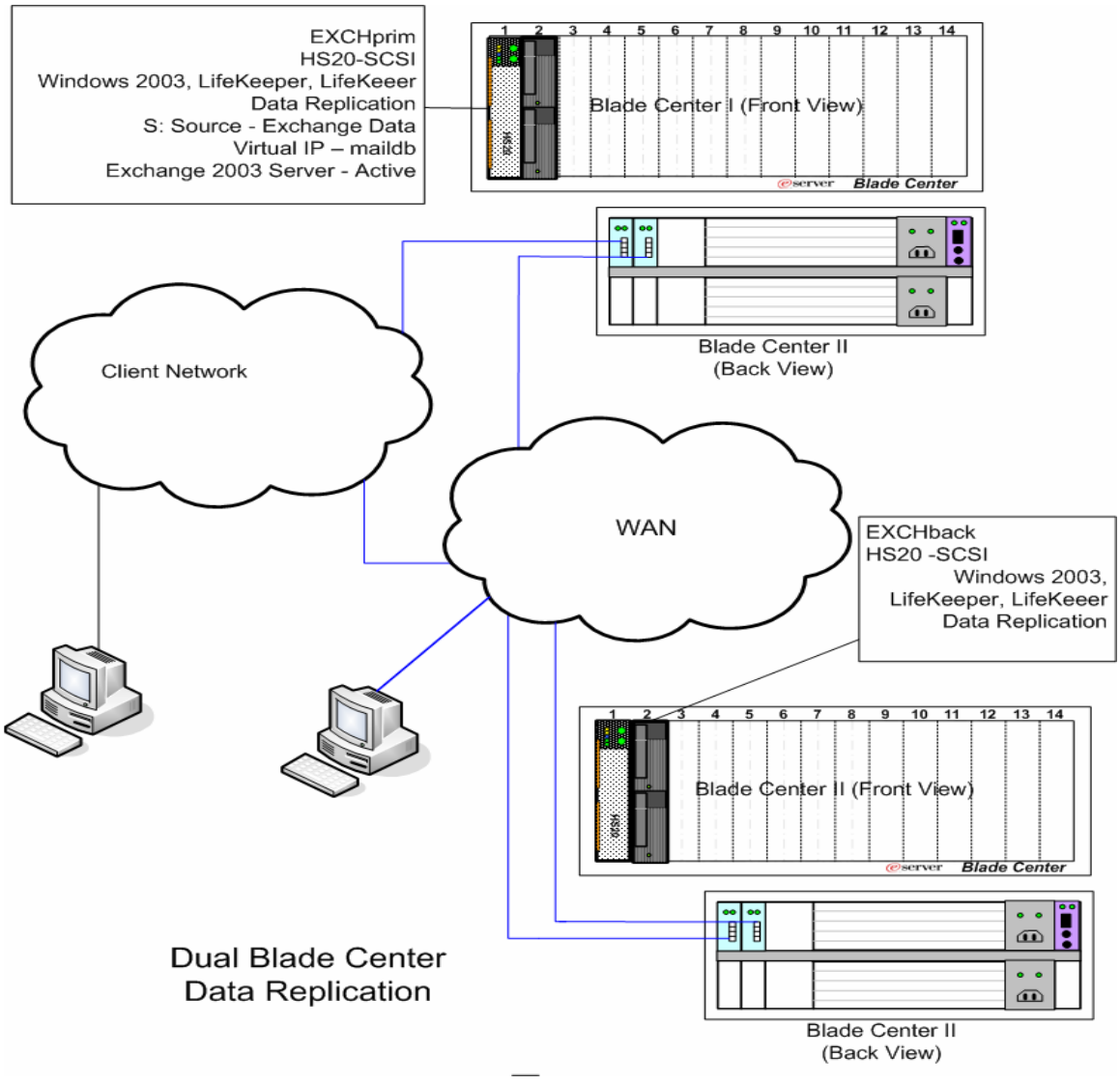
Using the mySQL example as above, this dual BladeCenter cluster configuration adds another layer of protection for Network Attached storage configurations. In a failover scenario, the LifeKeeper cluster will move the mySQL application from the Blade MYSprim to MYSback in the other BladeCenter. So in the rare event that the entire BladeCenter were to go down, the application would be protected. As above, this configuration is also very expandable.

Using Data Replication



In this example, Exchange 2003 is the protected Application. Using LifeKeeper Data Replication between Blades in two different BladeCenters, and Life Keeper for application high availability, the email system will be will protected. This again is a very clean configuration, with only three Ethernet cables tying the cluster system together. Also note that the Active Directory servers for this configuration are not shown, but they are necessary for Exchange to work properly. HS-20 Blades in each BladeCenter could serve as primary and backup AD controllers.

Between Blades in geographically dispersed BladeCenters



A slight, but significant, variation of the example above, this configuration separates the BladeCenters across a WAN. In this case it is imperative to use LifeKeeper Data Replication in asynchronous mode to make performance acceptable. As above, the Active Directory controllers are not shown, but the solution would require at least one on each side of the WAN

Conclusion

The combination of IBM eServer BladeCenter servers and SteelEye LifeKeeper provides a powerful solution which meets the needs of ‘On Demand computing’.
In today’s business world, customers, suppliers, employees and management expect continuous availability of critical systems. IBM eServer BladeCenter and SteelEye LifeKeeper can help make this expectation a reality, in a most economical manner. BladeCenter servers make it easy to deploy and expand critical business systems as part of a LifeKeeper high availability cluster.

References

Gartner, Dataquest, Contingency Planning Research and Others
SteelEye Technology™ LifeKeeper Product Briefs
IBM BladeCenter Brochure. Copyright IBM Corporation 2004