Linux High Availability for IBM xSeries

Bob Williamson
Vice President, Research & Development
SteelEye Technology, Inc.

bob.williamson@steeleye.com
www.steeleye.com
Session Agenda

• Overview of High Availability Clustering

• SteelEye LifeKeeper for Linux
  – How LifeKeeper Operates
  – Architecture
  – Configurations
  – Application Recovery Kits

• Building a Linux High Availability Cluster using LifeKeeper
High Availability. Easy & Affordable

Availability Primer

• Clustering provides higher levels of availability for business critical services by removing potential single points of failure.

• Primary causes of service being unavailable:
  - Software failure
  - Administrator error
  - Network outage
  - Planned maintenance
  - Server hardware failure
Availability Measures

- Availability Measures
  - 90% >1 month outage time per year
  - 99% just under 4 days outage time per year
  - 99.9% just under 9 hours outage time per year
  - 99.99% 53 minutes outage time per year
  - 99.999% 8 minutes outage time per year
  - 99.9999% 30 seconds outage time per year
  - 99.99999% about 3 seconds outage time per year

- Fault Resilient vs Fault Tolerant
  - Much less expensive to implement
    - Can be built from commodity components
  - Best solution for vast majority of implementations
    - Exceptions include in-flight aircraft systems, embedded medical devices, etc.
The “Availability Equation”

Time to Restore = Time to Detect + Time to Recover

• Detection
  - Are all cluster nodes available?
  - Are all application resources available?
  - What are the dependencies on the unavailable resource?
  - What is the impact of the unavailability?

• Recovery
  - What is the recovery policy?
  - What is the recovery procedure?
  - What dependent resources need to be recovered?
  - How will clients access recovered application?
  - What data does the application require?
  - How is the data accessed?
Monitor health of compute node and of individual services

• Heartbeats between cluster members
  – Multiple to protect against false failovers

• Agents to monitor individual resources
  – Applications, databases, file systems, IP address, ......

• Constant health monitoring with configurable check intervals

• Dependencies dictate monitoring order and recovery
Create Virtual Server consisting of all resources required to run business application
- IP Address for client access
- Database and/or File System for access to critical data
- Application services

On detection of problem, attempt to restart on local system. If this does not correct problem, migrate service to standby server. Cascade down server list until resource comes into service.
Linux HA Clustering Checklist
12 items to consider

✓ Support for standard Linux kernels and distributions
  - No requirement on Enterprise versions or patched kernels

✓ Support for variety of data storage configurations
  - Data Replication, Shared Fiber or SCSI, NAS, WAN

✓ Support for heterogeneous components
  - Flexible configuration, improved asset management

✓ Support for >2 node clusters
  - Opens up several possible configurations: One-to-Many, Many-to-One, Cascading
Linux HA Clustering Checklist
12 items to consider

- ✔ Active/Active and Active/Standby
  - Maximizes utilization of compute resources

- ✔ Detection of problem(s) at node and individual service level
  - Improved control, reduced “false failovers”, higher availability

- ✔ Recovery in-node and across nodes
  - Minimizes time to restore service

- ✔ Transparency to client connections of server-side recovery
  - IP Address and application resource virtualization
High Availability. Easy & Affordable

Linux HA Clustering Checklist
12 items to consider

✓ Protection for planned and unplanned downtime
  - Manual switchover enables systems maintenance with no service interruption

✓ Off-the-shelf protection for common business functions
  - Common agents for common tasks

✓ Ability to easily incorporate protection for custom business app
  - No modification to application
  - Software Developer’s Kit to assist

✓ Ease of cluster deployment and management
  - Creation, Administration, Monitoring
  - Cluster meta data management - distributed versus single repository
Choice of protection based on availability needs

- Different between organizations
- Different between applications
- Requirement will evolve over time

Level of Protection

Breadth of Capability

Protection Schemas

Two Node Data Replication
LAN

2 to 4 Node Failover
SCSI, LAN

n Node Failover
Fiber Channel, SAN

Stretch Cluster Replication & Failover
WAN
Entry-Level Configuration
- Lowest cost of entry
- No shared storage required
- Local dedicated access to application data
- Automated data replication
  - Synchronous or asynchronous
  - Disk-to-disk back-up
Shared storage configuration
- Shared SCSI, FC or NAS allows building of larger clusters
- Single write of data speeds processing versus replication
- Multi-path access to shared storage removes potential single points of failure
Summary

• HA clustering should be a key component of your overall Linux server system availability plan.

• By employing HA clustering, Linux systems built from commodity components provide levels of availability required for business critical functions.

• There are several criteria against which you should measure any Linux clustering solution, primary among them is the ability of the solution to scale as your availability needs change.

• Data replication can play an integral role in your availability deployment, providing a low-cost entry point.
Introducing SteelEye
LifeKeeper for Linux
How does LifeKeeper operate?

- Monitor health of system and of individual resources
- On detection of problem, attempt local recovery on same server
- If local recovery fails, perform failover to next priority stand-by node
- Alert administrator via SNMP
- When primary comes back online, failback (if configured to do so)
• **Failover**
  - Process of restoring a resource when the original server or resource has failed. Normally, a failover is *unplanned* and occurs when failure is detected by a backup system.

• **Switchover**
  - Process of shutting a resource or server down in an orderly manner and then recovering the resource or server on a backup system. This is normally performed in maintenance or test mode and is *planned*.
LifeKeeper Architecture
Major Components

LifeKeeper is composed of 3 major components:

• LifeKeeper Core
  • LK Configuration Database (LCD)
  • LK Communication Manager (LCM)
  • Lkcheck daemon
  • Alarm and Recovery APIs
  • SNMP Trap Generator

• LifeKeeper GUI

• Optional Application Recovery Kit(s)
LifeKeeper Architecture
Block Diagram

LifeKeeper Node

Application Recovery Kit

Resource Monitoring
Recovery Direction
Recovery Actions

LifeKeeper Core

Alarm API
Configuration API
Recovery API

Configuration Database
Communications Manager
Communication API

LifeKeeper GUI

Administration & Status Display

To LifeKeeper CM on another node
LifeKeeper Administration GUI used to

- Define Cluster membership
- Create/Delete communication paths
- Define resource hierarchical relationships
- “Extend” hierarchies to other cluster members
- Put resources “In-service” and “Out-of-service”
- Monitor status of cluster communication paths
- View status of cluster members
- View status of individual resources
- Read log files

- Written in JAVA
- Runs as either a JAVA application on an attached console or in any JAVA-enabled browser
SteelEye LifeKeeper for Linux

- **Distributions**
  - redhat
  - UNITED LINUX
  - SuSE

- **Scalable Configurations**
  - 2 node LAN cluster using LifeKeeper Data Replication
  - Shared Storage: Direct Attach SCSI, N-node Fiber Channel SAN
  - Network Attached Storage
  - Stretch cluster for geographic dispersion

- **Resources Protected off-the-shelf**
  - Databases: Oracle 8i/9i, DB2, MySQL, PostgreSQL, Informix, SAPDB
  - Applications: mySAP, Generic App via SDK
  - Services: Apache, Sendmail, SAMS, NFS, Samba, LVM

- **Simple to build, deploy, administer cluster**
  - Wizard-driven cluster configuration
  - JAVA GUI allows remote administration
  - Full CLI available for use if needed
The Configurations Are Key
LifeKeeper Configurations
Active/Active and Active/Stand-by

System Grouping Arrangements

Active / Stand-by

Active / Active
Storage Configurations
Shared Storage

Shared Storage
• Direct Attached SCSI:
  • Number of servers in cluster based on limitations of the ports on the SCSI enclosure (typically = 2 though there are quad-ported SCSI RAID enclosures, no LifeKeeper limitation)
  • Avoid Host-based RAID adapters except ServeRAID

• Fibre Channel:
  • LifeKeeper uses SCSI protocol over a Fibre Channel interconnect
  • Support for 32 servers in a cluster

• Network Attached

LifeKeeper...
• Prevents more than one resource from accessing the data at one time.
• Locks shared resources at the LUN level.
• Locks each LUN as a single disk, regardless of the physical location of the LUN.
Storage Configurations
Shared SCSI

Server 1

Server 2

Disk Array Controller

RAID Controller

Disk Array

RAID Control

Is Single Point of Failure (SPOF)

Physical Arrangement of Disks
(4 Physical Disks)

Logical Arrangement of Disks
(8 Partitions or LUNs)

SCSI Host Adapter

SCSI Host Adapter

LUN 0
LUN 2
LUN 4
LUN 6
LUN 1
LUN 3
LUN 5
LUN 7

RAID 5 Set
Storage Configurations
ServeRAID with EXP300

Controller no longer Single Point of Failure (SPOF)

Physical Arrangement of Disks
(4 Physical Disks)
Storage Configurations
Shared Fibre

FC Hub and Disk Array Controller are Single Points of Failure (SPOF)
2 Node Single Path Configuration with IBM FASt Fiber
2 Node Multi-Path Configuration with IBM FASt FC HA
Storage Configurations
Using Data Replication

Key Considerations:
• Network Bandwidth
• Amount of Data
• Sync or Async?
### Replication

- Easier to add to existing system
- Easier to Configure
- Nodes can be farther apart
- Easier to implement Disaster Recovery
- Can use any I/O adaptors and controllers
- Can use simple storage units
  
  - One-to-many backup not feasible

- Requires another copy of the data
- Uses CPU and I/O bandwidth in normal operation
- Tight synchronization causes performance loss; loose synchronization can lose data in a failure
- Failback causes need for resynchronization

### Shared Storage

- Must alter existing system cabling, etc.
- Disk Configurations must be synchronized
- Nodes must be physically close
- Disks are in a single location; no hot backup
- Must use HW that conforms to specific req’ts
- Must use hardened storage such as RAID
  
  - 1-to-many backup is feasible up to limit of disk interconnect
  - Shares a single copy of storage
  - No overhead in normal operation
  
  - Constant complete synchronization without data loss
  - Failback issues no different from failover
Introducing, ....The Recovery Kits
Recovery Kits
IP Address Protection

• Provides protection for IP Address access
• Delivered with LifeKeeper Core
• Often the first kit to be installed (used by other kits)
• Recovers an IP address or resource either between NIC cards in a single box or from a failed primary server to backup server(s)
• Recovery is possible through the use of a switchable IP address

Switchable IP Address (IP resource)

• a virtual or floating IP address that can switch between servers
• separate from the IP address associated with the network interface card
• also called IP aliasing
• Provides fault resilient protection for DB2 database instances.

• Provides protection for both classes of DB2 enterprise level database servers; the Enterprise Edition (EE) and the Enterprise-Extended Edition (EEE) in v7.2 and the Enterprise Server Edition (ESE) v8.1

• LifeKeeper with the DB2 Universal Database product family increases availability of DB2 operating environments by recovering database server failures without significant down-time or human intervention. Protection of the Application on top of the Database further extends the overall solution availability.
DB2 Supported Configurations

In these simple two node configurations, the DB2 database may be running in active/stand-by or active/active.
DB2 Supported Configurations

One DB2 EEE or ESE instance with two database partition servers is protected on Server 1 with one LifeKeeper DB2 resource hierarchy. Server 2 will assume ownership of the DB2 resource hierarchy when a failure occurs.

DB2 Active/Standby Configuration (1 Cluster)
DB2 Supported Configurations

One DB2 EEE or ESE instance with one database partition server is protected on Server 1 and one database partition server protected on Server 2. There is one LifeKeeper DB2 resource hierarchy on Server 1 and another DB2 resource hierarchy on Server 2. When a failure occurs each server will assume the other’s resources.
Recovery Kits
Oracle Protection

- Provides protection for Oracle 8i and 9i database services
- Operates with either Standard Edition or Enterprise Edition
- Operates in a RAC environment
  RAC protects DBMS, LifeKeeper protects application and other system services
- Can operate in either Active/Stand-by or Active/Active configurations
- Can be used as component of integrated solution to provide protection for all tiers of application architecture
  mySAP as an example
• Provides fault resilient protection for SAP Central Instances (CI) and associated resources.

• Works in conjunction with Database Recovery Kits (Oracle, DB2, SAPDB) to provide for full fault-resiliency in SAP deployments.

• Supports both Active/Stand-by and Active/Active configurations.
In this configuration, server s1 acts as a Primary, s2 acts as a backup. No applications are hosted on the Backup Server. This configuration might appeal to customers who do not want to have any performance degradation after a failover.
This configuration has s1 acting as a primary database server and s2 acting as a primary SAP server. The two servers are backing each other up for these functions. This configuration might appeal to customers who don’t have hardware capable of continuously hosting both CI and DB on one server and want to get the most use out of the hardware that they have. With this configuration, performance will be degraded after a failover since both DB and CI will be running on the same server. Both servers will have to be configured to supply at least a minimum level of performance after a failover.
LifeKeeper monitors all components of the end-to-end solution and takes appropriate recovery action taking into account the dependencies between solution components.
Generic Application Recovery Kit

- Allows for protection of custom applications
  - Internally developed business critical app
  - Applications without officially released Recovery Kits

- Sample scripts available as guideline

- Process as easy as defining:
  - How to programmatically monitor health of application
  - What to do when problem is detected
  - Relationship/dependency among the resources required by the application

- Separate SDK provides detailed documentation to assist with development, testing and packaging.
Building a Linux HA Cluster
Operational Considerations

Cluster servers
- All servers must be running the same Operating System
- Servers may have different hardware configurations
- Each server is assigned a priority for a given resource

Communication paths
- Minimum of two heartbeat communication paths to avoid single point of failure

Data resources
- Shared or replicated?
- File systems

Shared communication resources
- IP addresses

Application resources
- Off-the-shelf kits or Need to develop for custom application?
Operational Considerations

What is the cluster system grouping arrangement?
- Active/Active
- Active/Stand-by

What is the switchback configuration?
- Automatic switchback
- Intelligent (manual) switchback

What action should LifeKeeper take on system shutdown?
- Switchover resources
- Do not switchover resources
Networking Considerations

- **IP addresses for each system**
- **Two network interfaces are recommended**
  - one private LAN for LifeKeeper communication
  - one public LAN
  - public LAN can also be a secondary communication path
  - do not use private LAN as Data Replication interface
- **Communication paths can not be on the same subnet**
- **Verify the network is functional before starting the communication path configuration**
- **Network address must be resolvable**
  - `/etc/hosts`
  - DNS
Building a Linux HA Cluster
Gather the components - Hardware

**Servers**

Intel processor-based servers

**Universal servers**

- xSeries 200
- xSeries 220
- xSeries 230
- xSeries 240
- xSeries 250

IBM @server xSeries

- xSeries 300
- xSeries 330
- xSeries 340
- xSeries 342
- xSeries 350
- xSeries 360
- xSeries 380 (ia64)
- xSeries 440

BladeCenter Servers

- **Rack-optimized servers**

High Availability. Easy & Affordable
Building a Linux HA Cluster
Gather the components - Hardware

Storage

SCSI
- Adaptec 294x w/ compatible storage enclosure
- ServeRAID Host Adapter w/ EXP 300 storage enclosure

Fibre
- Qlogic QLA2x00F or IBM FASfT Host Adapter w/ FASfT Fiber Channel Storage Enclosure

Many other options available
Building a Linux HA Cluster
Gather the components - Software

Operating System
- redhat
- UNITED LINUX
- SuSE

Application Software
- DB2 Data Management Software
- ORACLE
- Powered by APACHE
- SAP

Custom App
Building a Linux HA Cluster
Gather the components - Software

LifeKeeper

Core

Installation Support

Recovery Kits
The following steps are required before installing LifeKeeper:

1. Install Linux and associated packages
2. Connect servers and shared storage
3. Configure shared storage
4. Verify network configuration
5. Install and set up database application (if applicable)
6. Set up LifeKeeper environment using LifeKeeper Installation Support CD

Note: Consult the LifeKeeper Planning and Installation Guide and LifeKeeper Release Notes for additional installation details.
Installation Support CD provides an interactive set-up script

- Identifies Linux distribution
- Installs appropriate packages required for successful LifeKeeper installation
- Initiates installation of LifeKeeper license key

- Two types of License Keys:
  - Permanent
  - Evaluation
- Required for LifeKeeper Core only (not kits)
LifeKeeper Installation
Two types of License keys

• Permanent License Key
  - For purchased software
  - No expiration
  - Requires input of two items:
    • **hostid of server (obtained via lmhostid utility)**
    • **Authorization code (shipped with media)**
  - Obtained thru SteelEye website ([www.steeleye.com/support](http://www.steeleye.com/support))

• Evaluation License Key
  - For downloaded and evaluation software
  - 30 day timestamp
  - Requires hostid only (no authorization code)
  - Obtained thru SteelEye Sales or Support
LifeKeeper Installation
Configure Cluster

• Define and test communication paths. Optimize settings as needed.

• Define paths to critical data.
  – If using data replication, establish source and target systems and do initial data sync.
  – If using shared storage, ensure that all cluster nodes can see the shared storage.

• Define application hierarchies to be protected by LifeKeeper. LifeKeeper GUI Wizard will walk you thru this process.

• Test LifeKeeper operation using manual switchover.

• Test LifeKeeper operation by inducing error conditions

• Put LifeKeeper into Production usage.
Linux High Availability for IBM xSeries

Bob Williamson
Vice President, Research & Development
SteelEye Technology, Inc.

bob.williamson@steeleye.com

www.steeleye.com