How To Build A High Available Environment With Linux On System z

Steffen Thoss
System z Linux and Virtualization PDT Leader
IBM Systems & Technology Labs
stethoss@cn.ibm.com

Marcus Kraft
Product Manager
SUSE Linux Enterprise Server for System z
mkraft@suse.com
Abstract

Linux running on IBM System z inherits the characteristics of a reliable and highly scalable server.

If you want to build an High Available environment, on the technical side, you need to consider all levels in the IT environment from your disk infrastructure, virtualization layer to the applications and databases.

Come and see what you need to consider and how you can build it, with out of the box tools from SUSE and IBM to take advantage of the qualities of System z.

Experience increased uptime, servicability and resilience for your mission critical business services.
Definitions

• High Availability (HA)
  - Provide service during defined periods, at agreed levels, and masks *unplanned* outages from end-users. It employs Fault Tolerance; Automated Failure Detection, Recovery, Testing, Problem and Change Management

• Continuous Operations (CO)
  - Continuously operate and mask *planned* outages from end-users. It employs Non-disruptive hardware and software changes, non-disruptive configuration, software coexistence.

• Continuous Availability (CA)
  - Deliver non-disruptive service to users 7day/week, 24hs a day
  - There are no planned or unplanned outages

• The goal is to strive to provide Continuous Availability.
Business Continuity Issues

What are the reasons for system outages?

- **Planned** outages
  - Maintenance
  - Tests

- **Unplanned** outages
  - Operator errors
    - Lack of application skills
    - Lack of OS skills in heterogeneous environment
  - Application failures
    - SW exceptions
    - Environment / Configuration problems
  - Environmental failures
    - OS failures
    - HW failures
    - Disasters

Source: Gartner Group
**Identify RTO, RPO, NRO**

**Recovery Time Objective (RTO)**

What time difference can be between Failure and a total productional run level?

**Recovery Point Objective (RPO)**

What is the toleration for data loss?

RTO = “0” means, NULL data loss acceptable

RPO = “5” means, data loss in last 5 min acceptable

TREND: RPO = 0

**Network Recovery Objective (NRO)**

Time requirements for network availability.
Business Impact Analysis (BIA)

- IT Resource relation and priorities for DR
  - Consider all environments
  - Prioritize based on business importance

Figure 3-2  Example of the Business Impact Analysis process
High Availability And Disaster Recovery

Differences

• High Availability (HA)
  - Failover is typically realized via duplication and clustering
  - Failover times measured in seconds and minutes
  - Reliable inter-node communication

• Disaster Recovery (DR)
  - Failover is typically realized with 2 or more sites in case of disasters
  - Failover times often measured in minutes and hours
  - Unreliable inter-node communication assumed
Components Of High Availability

Software Provided HA
- Web Srv.
- Apps.
- Database

Operating System HA
- Linux
- z/VM

Network HA
- Network

Hardware Provided HA
- System z
- Storage (Disk)

HA Software
- Clustering
- Automation

Cluster concepts
- Mature Hypervisor
- Hardware assist for performance and recovery

Network HA
- Hipersockets, VSWITCH, OSA

Hardware Provided HA
- Spare CPUs
- N+1 power supplies
- Chip sparing in memory
- Concurrent Maintenance

Network HA
- Disk mirroring Global/Metro
- Hyperswap
- Multipathing / Failover
Clustering Concepts

• Computer Cluster
  - A computer cluster consists of a set of loosely connected computers that work together so that in many respects they can be viewed as a single system. (Wikipedia definition: Computer Cluster)

• High Availability Cluster
  - A computer cluster where each cluster operates as workload node. When one node fails another node takes over the entire workload: IP address, data access, services, etc.
  - The key of High Availability is avoiding single points of failure
  - High Availability adds costs because of added complexity due to redundant resources in the environment
High Availability Considerations
Linux, z/VM, System z

<table>
<thead>
<tr>
<th>Single Point of Failure</th>
<th>Probability of Failure</th>
<th>Cost to fix</th>
</tr>
</thead>
<tbody>
<tr>
<td>System z hardware</td>
<td>Very Low</td>
<td>High</td>
</tr>
<tr>
<td>LPAR</td>
<td>Very Low</td>
<td>Low</td>
</tr>
<tr>
<td>z/VM</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>Linux</td>
<td>Low</td>
<td>Very Low</td>
</tr>
<tr>
<td>Disk Subsystem microcode</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>Application</td>
<td>High</td>
<td>Very Low</td>
</tr>
</tbody>
</table>

- Besides hardware and software failures, what else can cause production downtime?
  - System z hardware upgrades requiring Restart
  - LPAR configuration changes requiring reboot of the LPAR
  - z/VM maintenance (if not SSI clustered)
  - Linux kernel maintenance that requires reboot
  - Application maintenance
IBM System z
Redundancy Build Into The System

Rear View

Front View

Internal Batteries (redundant)

Power Supplies (redundant)

2 x Processor Drawers, Memory & HCAs

RAIM Memory (Redundant Array of Independent Memory)

I/O Drawer (redundant)

PCIe I/O drawers (redundant)

2 x Support Elements
Storage High Availability Options

- Multiple Storage Subsystems
- FICON attached Disks
  - ECKD disks
  - multi channel connections
- FCP/SCSI attached disks
  - SCSI disks - LUN attached
  - Multi pathing / LVM
  - Streched SVC for HA
- Disk Replication
  - Metro / Global Mirror (PPRC)
  - Virtual disk mirror (SVC)
  - LVM mirror
  - Oracle ASM mirror
- z/VM Hyperswap
  - Online disk swap

- LPAR 1
  - Appl
  - WAS
  - DB
  - LVM
  - FCP

- LPAR 2
  - Appl
  - WAS
  - DB
  - HyperPAV
  - FICON

- z/VM

- Stretched SVC

- Disk mirror

- ECKD

- Hyperswap

- FCP/SCSI
High Availability Scenarios
Active / Passive with System z

- **Active / Passive Deployment**
  - Workload normally contained at site 1, standby server capability at site 2
  - Primary and secondary disk configurations active at both sites
  - During fail over, **Capacity Upgrade on Demand (CUoD)** adds resources to operational site, and standby servers are started. Helps save hardware and software costs, but requires higher recovery time.

- **Hot / Cold scenario**
  - Workload is not split
  - Each site is configured to handle all operations
  - Cold environment needs longer to get active
  - Often used in DR

- **Hot / Warm scenario**
  - Workload is not split
  - Each site is configured to handle all operations
  - Warm environment is idling
High Availability Scenarios (2)
Active / Active with System z

- **Active / Active Deployment**
  - Workload is normally split between 2 or more sites
  - Each site is configured to be able to instantly cover the workload if needed
  - During normal operation, excess capacity at each site is consumed by lower priority, work like development or test activities
  - In a failover situation, low priority work is stopped to free up resources for the production site’s incoming work

- **Capacity Upgrade on Demand (Active / Active)**
  - Workload is normally split between sites
  - Each site is configured with capacity to handle normal operations
  - Special setup with Capacity Upgrade on Demand (CUoD)
  - In a failover situation, additional CPUs are enabled at the operational site
Unique: Capacity On Demand
Cold or HotStandBy Workload Migration

- 1 system online, workload moved
- Disks are mirrored by Storage Unit
- Processor are activated on demand
- HotStandBy system continue asap
- ColdStandBy system are started
Components Of High Availability

Software Provided HA
- Web Srv.
- Appls.
- Database

Operating System HA
- Linux
- z/VM

Network HA
- Network

Hardware Provided HA
- System z
- Storage (Disk)

HA Software
- HA Software
- Clustering
- Automation

Cluster concepts

Mature Hypervisor
- Hardware assist for performance and recovery

Hypersockets, VSWITCH, OSA

Spare CPUs
- N+1 power supplies
- Chip sparring in memory
- Concurrent Maintenance

Disk mirroring Global/Metro Mirror
- Hyperswap
- Multipathing / Failover
System z Virtualization

Note: There are typically dozens or hundred of Linux servers in a z/VM LPAR.

P1 – P8 are Integrated Facility for Linux (IFL) Processors

* - One shared Pool of CPUs per machine (CEC) only
Network Interface
HA and Automated Failover

OSA-card HA
Channel Bonding for enhanced bandwidth

- Linux *bonding* driver enslaves multiple OSA connections to create a single logical network interface card (NIC)
- Detects loss of NIC connectivity and automatically fails over to surviving NIC
- Active/backup & aggregation modes
- Separately configured for each Linux

Network HA with z/VM VSWITCH
Port aggregation for enhanced bandwidth

- z/VM *VSWITCH* enslaves multiple OSA connections. Creates virtual NICs for each Linux guest
- Detects loss of physical NIC connectivity and automatically fails over to surviving NIC
- Active/backup & aggregation modes
- Centralized configuration benefits all guests
Network Alternatives
Linux, z/VM, LPAR, System z

Linux on System z
- Application
- TCP/IP
- HIPER Sockets
- VSWITCH / VLAN
- OSA
- LPAR 1

z/OS
- Application
- TCP/IP
- LPAR 2

External network
Components Of High Availability

Software Provided HA
- Web Srv.
- Appls.
- Database

HA Software
- Clustering
- Automation

Operating System HA
- Linux
- z/VM

Cluster concepts
- Mature Hypervisor
- Hardware assist for performance and recovery

Network HA
- Network

Hypersockets, VSWITCH, OSA

Hardware Provided HA
- System z
- Storage (Disk)

Spare CPUs
- N+1 power supplies
- Chip sparring in memory
- Concurrent Maintenance

Disk mirroring Global/Metro Mirror
- Hyperswap
- Multipathing / Failover
**z/VM V6.3 - Increase Availability for Linux guests**

Single System Image, Clustered Hypervisor, Live Guest Relocation

- **Single System Image (SSI)**
  - Connect up to four z/VM systems as members of a cluster
  - Provides a set of shared resources for member systems and their hosted virtual machines
    - Directory, minidisks, spool files, virtual switch MAC addresses
  - Cluster members can be run on the same or different z10, z196, zEC12 or BC servers
  - **Simplifies systems management** of a multi-z/VM environment
    - Single user directory
    - Cluster management from any member
      - Apply maintenance to all members in the cluster from one location
      - Issue commands from one member to operate on another
    - Built-in cross-member capabilities
    - Resource coordination and protection of network and disks

- **Live Guest Relocation (LGR)**
  - Dynamically move Linux guests from one z/VM member to another
  - Reduce planned outages; enhance workload management
    - Non-disruptively move work to available system resources **and** non-disruptively move system resources to work
    - When combined with Capacity Upgrade on Demand, Capacity Backup on Demand, and Dynamic Memory Upgrade, you will get the best of both worlds
High Availability for Operating Systems
z/OS and Linux

- z/OS Parallel Sysplex HA
  incl. memory sharing

- SUSE Linux Enterprise
  High Availability for System z
  using cluster software
Components Of High Availability

Software Provided HA
- Web Srv.
- Apps.
- Database

Operating System HA
- Linux
- z/VM

Network HA
- Network

Hardware Provided HA
- System z
- Storage (Disk)

HA Software
- Clustering
- Automation

Cluster concepts
- Mature Hypervisor
  - Hardware assist for performance and recovery

Hypersockets, VSWITCH, OSA
- Spare CPUs
- N+1 power supplies
- Chip sparring in memory
- Concurrent Maintenance

Disk mirroring Global/Metro Mirror
- Hyperswap
- Multipathing / Failover
Independent Tiered Execution Streams
High Availability Clustering Cases
Heterogeneous Environment

- SUSE Linux Enterprise High Availability Extension
  - For SUSE Enterprise Server Linux environments

- Tivoli System Automation for Multiplatforms
  - For z/OS
  - For multiplatforms
  - For distributed heterogeneous environments
IBM Tivoli System Automation
Multiplatform Support

• IBM Tivoli System Automation for Multiplatform (SA MP) can take advantage of:
  - Linux-HA heartbeat environment
  - Enable cross platform HA
  - z/OS High availability together with Linux
  - Implements advanced resource group automation
  - Dependencies and policy management and hierarchies
  - Can be used in HA for non-clustered systems and applications
  - Supports various platforms including System z
  - Contains a variety of predefined HA adapters for middleware (i.e. DB2, SAP, WebSphere, Apache …)
HA with z/OS GDPS and xDR with z/VM guests

- **Proxy**
  - One Linux system is configured as a Proxy for GDPS which has special configuration
    - Memory locked, Access rights to VM, One-Node-Cluster
  - Is used for tasks that have z/VM scope
    - HyperSwap, shutdown z/VM, IPL z/VM guest

- **Production Nodes**
  - Run Linux Workload
  - Are used for local actions (Shut down node, Maintenance Mode)

- **Other Systems**
  - Enabled for HyperSwap via xDR Proxy (Linux, z/VSE)
  - No re-IPL in place, no start/stop via GDPS (init, reipl, maint)
SUSE Linux Enterprise
High Availability Extension
Fundamentals Of High Availability

• Redundancy, Redundancy
  - Duplicate to eliminate single points of failure

• Handle Relations Of Redundant Objects Consistently
  - Status, dependencies, constraints, etc

• Early detection
  - To keep offline time as short as possible
  - Reduce risk of wrong interpretation and unnecessary failover
  - Keep offline time as short as possible (mean-time-to-repair MTTR)
Fundamentals Of High Availability (2)

• Automate Detection and Failover
  - Let the system do the work in order to minimize outage windows
    - Multipath I/O
    - VIPA – Virtual IP Addresses
    - Monitoring and heart-beating
    - Clustered middleware
    - Clustered operating systems

• Protect Data Consistency
  - Provide ability for data and file systems to return to a point of consistency after a crash
    - Journaling databases, journaling file systems
    - Mirroring, routine database backups
SUSE® Linux Enterprise High Availability Extension

Service availability 24/7
- Policy driven clustering
  - Messaging and membership layer
  - Pacemaker cluster resource manager

Sharing and Scaling data-access by multiple nodes
- Cluster file system
  - OCFS2
  - Clustered Logical Volume Manager

Disaster tolerance
- Data replication via IP
  - Distributed Replicated Block Device
- Node recovery

Scale network services
- IP load-balancing

User friendly tools
- Graphical user interface
- Unified command line interface
Key Use Cases
SUSE® Linux Enterprise High Availability Extension

• Achieve high availability of mission-critical services

• Active / active services
  – OCFS2, Databases, Samba File Servers

• Active / passive service fail-over
  – Traditional databases, SAP setups, regular services

• Private Cloud
  – HA, automation and orchestration for managed VMs

• High availability across guests
  – Build full HA on top of virtualization

• Remote clustering
  – Local, Metro, and Geographical area clusters
Local & Stretched Cluster

Clients

SLES SLE HA
SLES SLE HA
SLES SLE HA
SLES SLE HA
Geo Cluster – Setup

Site A

Node 1

Node 2

Site B

Node 7

Node 8

Site C (Arbitrator)

boothd
Reasons for SUSE® Linux Enterprise High Availability

- Long history track record
- Most up-to-date Open Source High Availability stack
- Deep OS integration
- Ready for Virtualization
- Integrated Data Replication
- Superior Cluster File System
- Geo cluster support
SUSE Linux Enterprise HA
HA Components

Linux HA stack, incl. pacemaker, corosync, hawk, and YaST module:

- resource-agents – monitor availability of resources
- stonith – fencing support (VMs and LPARs)
- corosync and OpenAIS – cluster infrastructure
- pacemaker – cluster resource manager
- hawk – Web console for cluster monitoring and dependencies editing
- crm – command line to interact with the CIB: editing, prepare multiple changes & commit once, syntax validation, etc.
- crm GUI - graphical interface for cluster resource and dependencies editing
- YaST module – easy basic installation
Detailed Architecture
SUSE® Linux Enterprise High Availability Extension

Detailed view of components per node:
Management Tools

- SUSE Linux Enterprise High Availability Extension includes a powerful new unified command-line interface for experienced IT managers to quickly and easily install, configure and manage their clustered Linux servers.
- Graphical user interface that provides operators with a simple, user-friendly tool for monitoring and administering their clustered environment.
- New YaST2 modules for the configuration:
  - of DRBD,
  - openAIS
  - multipath

Supported Platforms
- SUSE Linux Enterprise Server 11
- for x86, x86_64,
- Itanium*
- Power*
- System z*
Disaster Recovery Plan

- Set Objectives
- Business Impact Analysis
- Infrastructure Adjustment
- Process Plan and Execution

- Repeat testing and refine all of the above!

- “Virtual Linux Server Disaster Recovery Planning”
  Rick Barlow, NationWide at SHARE Boston 2013
  https://share.confex.com/share/121/webprogram/Session13726.html
Ideal High Availability Architecture

Allows service to continue no matter what fails ...

- An HA architecture protects the service from product failures by eliminating Single Points of Failure (SPoFs) at all layers (not just internal within the box).
  - Facilities, HW & SW components, Middleware or subsystems, Applications, Data, etc.

- This ideal approach is typically referred to as an active/active solution and may eliminate any service disruption for a single failure scenario.
System z Unique Selling Points
"Changing Tires At 100mph"

Replacing a z10 book during production
Provide high available services. Experience resilience.

www.suse.com/products/highavailability/
www.suse.com/mainframe

Thank you.
Appendix
SUSE Linux Enterprise High Availability
for Linux on System z

- HA for Linux on z using clustering
  - is implemented in the SLES Distro for Linux on System z
  - License is part of SLES for Linux on System z
  - HA implementation bases on Linux-HA
  - Graphical tools included for cluster management and monitoring resources

- SUSE Linux Enterprise High Availability Extension delivers all the essential monitoring, messaging and cluster resource management functionality

- HA Add-on: Geo Clustering for SUSE Linux Enterprise High Availability Extension lets you deploy Linux clusters between data centers spread anywhere in the world.

- In the SLE HA, Pacemaker is included, a scalable cluster resource manager with a flexible policy engine that supports n-node clusters

- OpenAIS, as one of the leading standards-based communication protocol for server and storage clustering is used for communication
  - Using OpenAIS and Pacemaker, you can continuously monitor the health of your resources, manage dependencies, and automatically stop and start services based on highly configurable rules and policies
Additional documentation

- SUSE Linux Enterprise High Availability 11 SP3 Guide

- Linux-HA project Open source
  http://www.linux-ha.org/wiki/Main_Page

- IBM Redbook
  Achieving High Availability on Linux for System z with Linux-HA Release 2

- Tivoli System Automation for Multiplatforms
Unpublished Work of SUSE. All Rights Reserved.
This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE. Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer
This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.