



Highly-Available Linux on Itanium: A Case Study



Walt Saiko
Senior Technical Consultant
Hewlett-Packard

© 2004 Hewlett-Packard Development Company, L.P.
The information contained herein is subject to change without notice



agenda

- Introduction
- Session Is / Session Is Not
- Overview of Serviceguard
- Serviceguard for Linux on Itanium
- Tradeoffs of Supported Linux Distributions
- Case Study
 - Challenge
 - Alternatives
 - Solution

introduction - biography

- Ten years with HP
- Five years as HP customer
- Senior Technical Consultant
- Degrees:
 - BS Computer Science
 - MS Computer Science
 - MBA
- 15 years Unix experience
- 10 years high availability focus



introduction - my roles within HP

- HP Technology Solutions Group
 - Partner Technology Access Center (PTAC)
- Work exclusively with Independent Software Vendors (ISVs)
- Ensure that their applications perform best on the HP platform (HP-UX, Linux, PA-RISC, Itanium)



introduction – PTAC services

- Delivery of Free Services to ISVs, including:
 - High Availability Implementation & Validation
 - Application Performance Analysis & Tuning
 - Compiler Optimization
 - Application Porting & Migration
 - Application Profiling and Sizing



point of clarification

- Itanium = Itanium2 = IPF = Integrity

prerequisites

- System administration experience
- Network experience
- Serviceguard knowledge
- Linux knowledge
- HP-UX knowledge
- Understanding of issues surrounding high availability

session is **NOT**:

- Sales pitch for Serviceguard
- Sales pitch for HP hardware
- Tutorial on Linux/Open Source/High Availability
- Tutorial on HP's Serviceguard
- Recommendation of one distribution of Linux over another
- Comparison/contrast of high availability solutions on Linux and HP-UX

session IS:

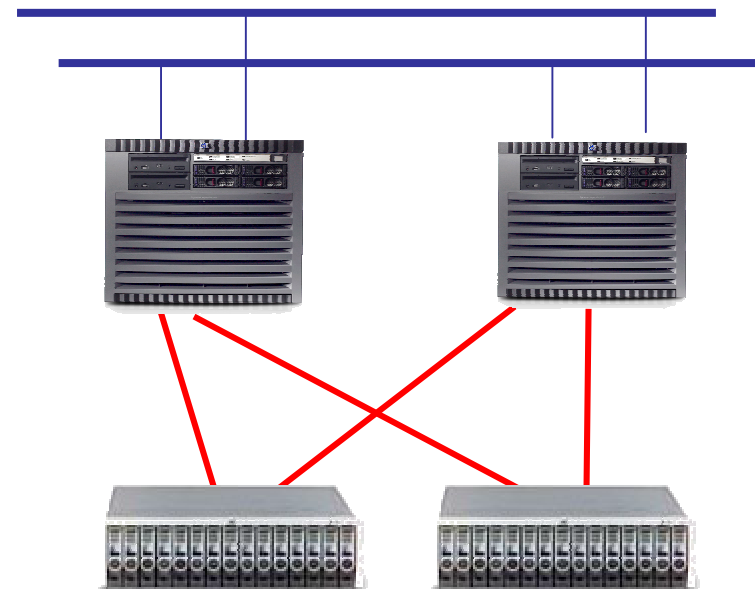
- Overview of Serviceguard for Linux on Itanium
- Examination of one company's effort to build a highly-available implementation of a Linux application on HP hardware
- Steps taken to provide a highly-available application environment for this ISV
- Issues faced
- Alternatives considered
- Decisions reached
- Solution built
- Lessons learned

Overview of Serviceguard functionality...



hp Serviceguard

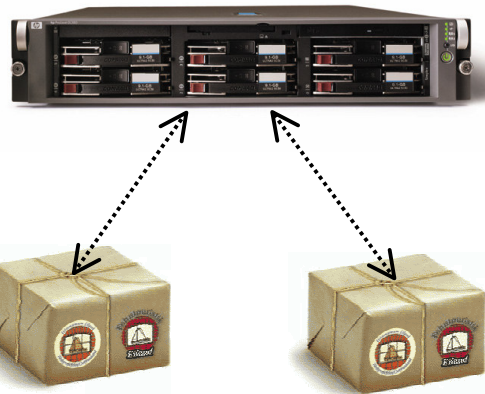
- No fault tolerance, no load balancing cluster
- Monitors hardware & software resources
- Requires redundant hardware components (no SPOF)
- Runs on standard hardware and operating systems
- Integration of applications without any modification
- More than 80,000 licenses sold (HP-UX & Linux)
- Supports local, campus, metropolitan and continental configurations



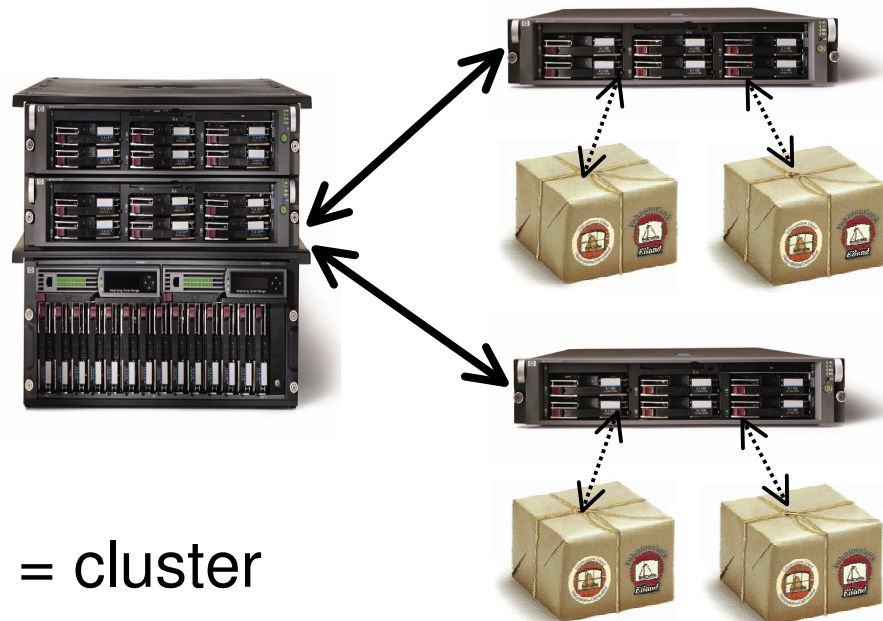
terminology



= node



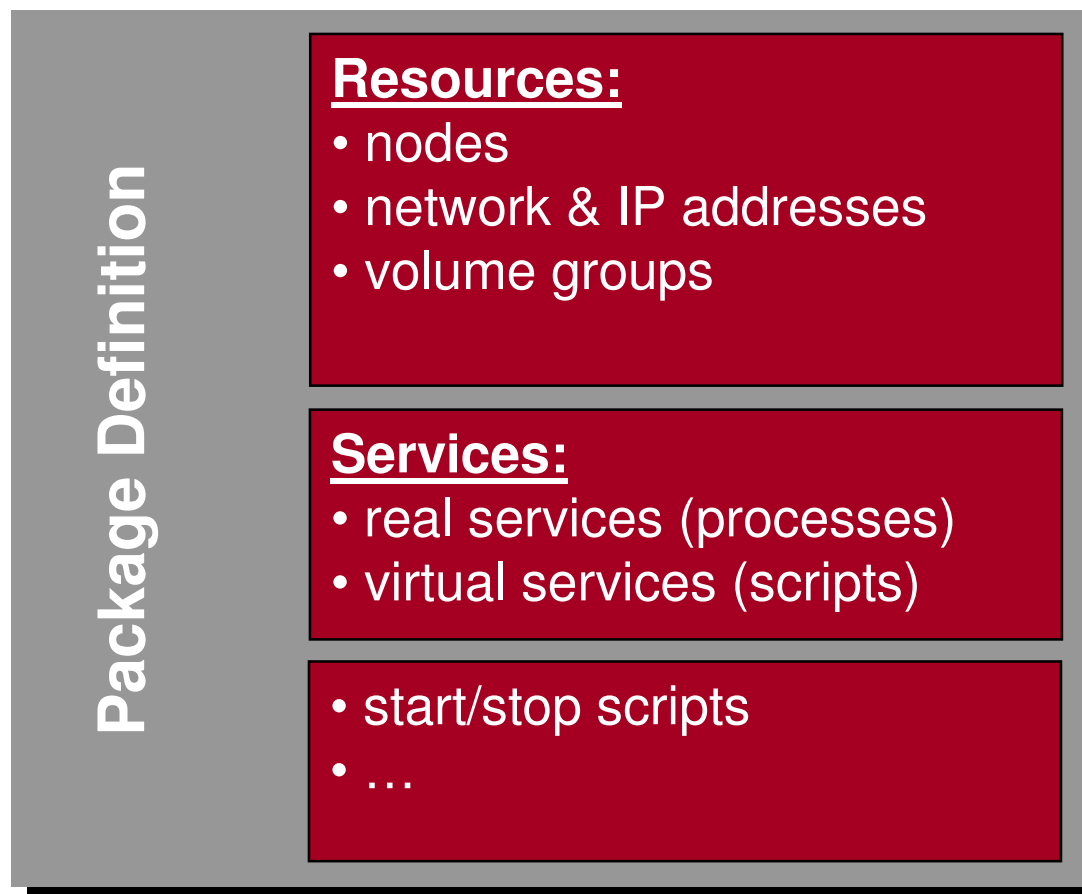
= package



= cluster

package definition

A **package** defines all resources required to run an application or part of an application on a cluster node.



package definition (continued)



- **IP Address :**

- 15.128.129.130 on subnet 15.136.129.0

- **disk space :**

- VG=vg01 LV=/dev/vg01/lvol1 FS=/appl1

- LV=/dev/vg02/lvol1 FS=/appl2

- **processes (via services):**

- application process 1

- application process 2

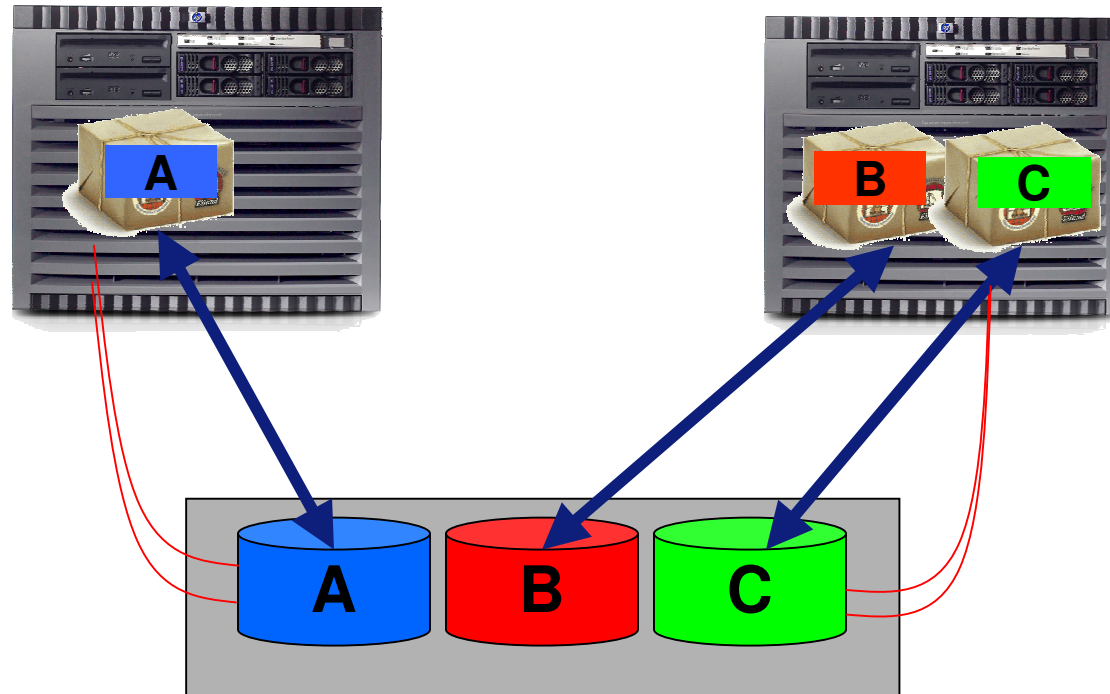
- middleware processes

- monitoring scripts

- **start & stop scripts**

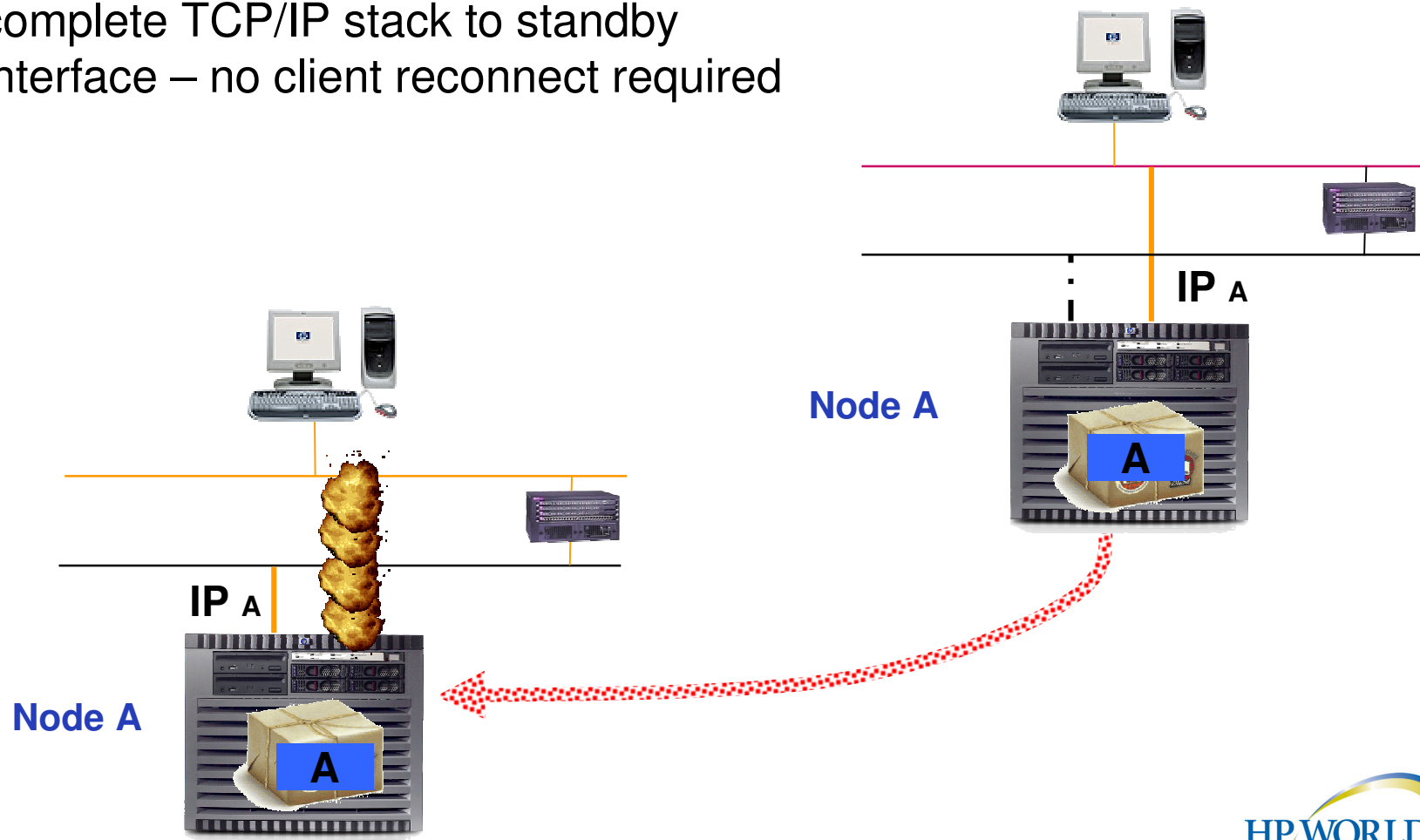
storage configuration

The **Nodes** of a **Cluster** share storage via multi initiator SCSI buses or fiber channel infrastructure.



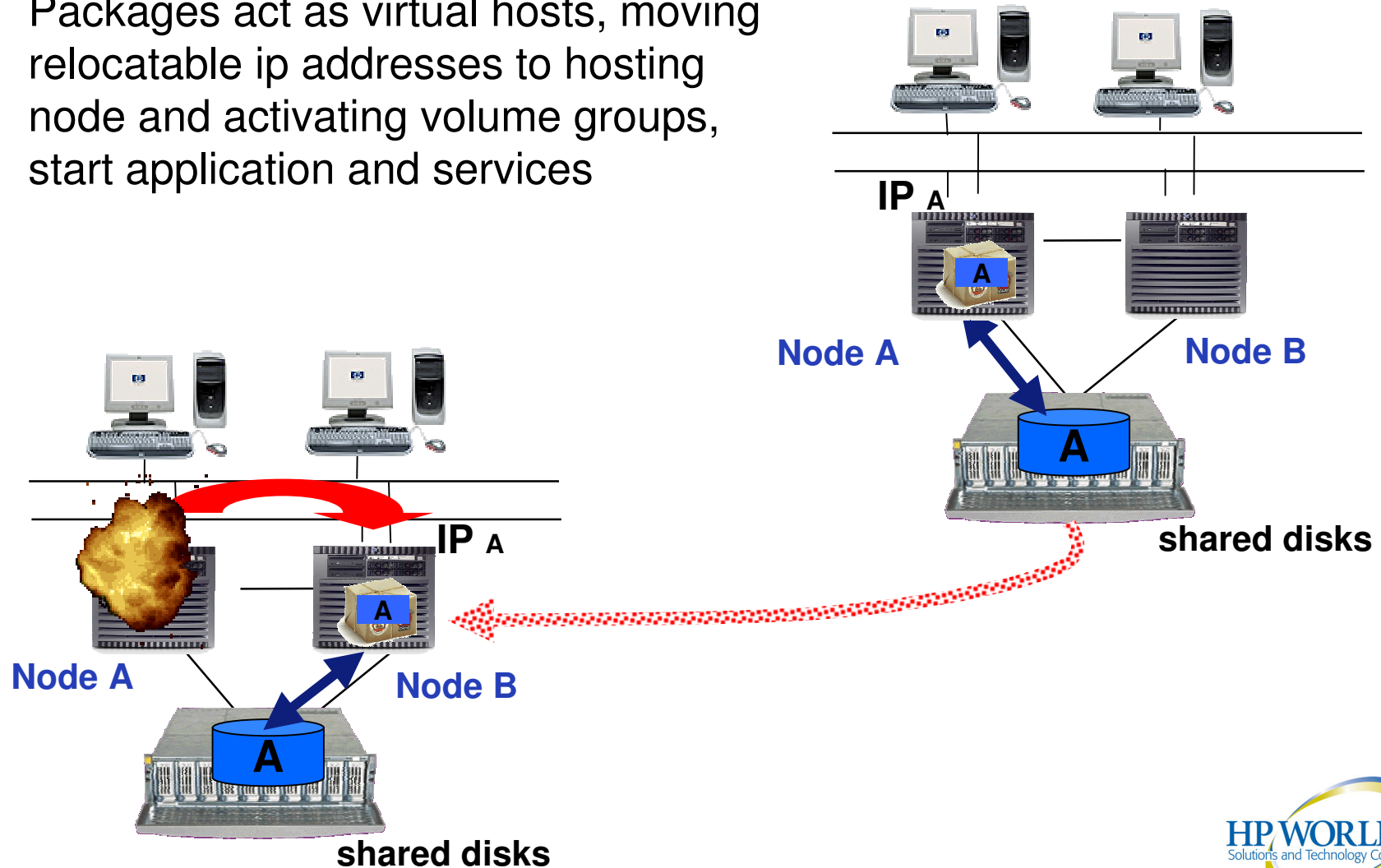
local LAN switch

Fast and transparent switch of complete TCP/IP stack to standby interface – no client reconnect required

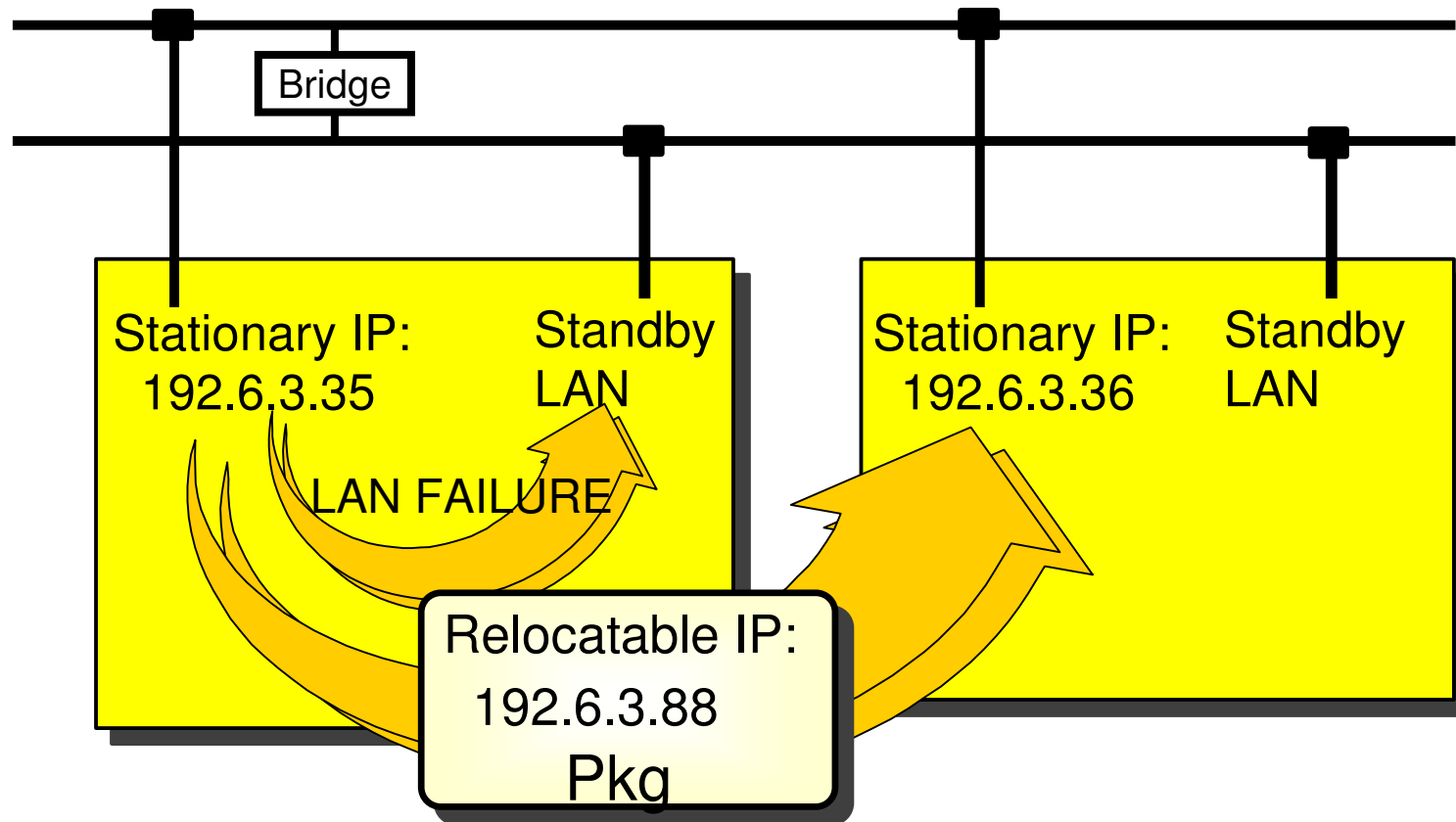


package switch

Packages act as virtual hosts, moving relocatable ip addresses to hosting node and activating volume groups, start application and services



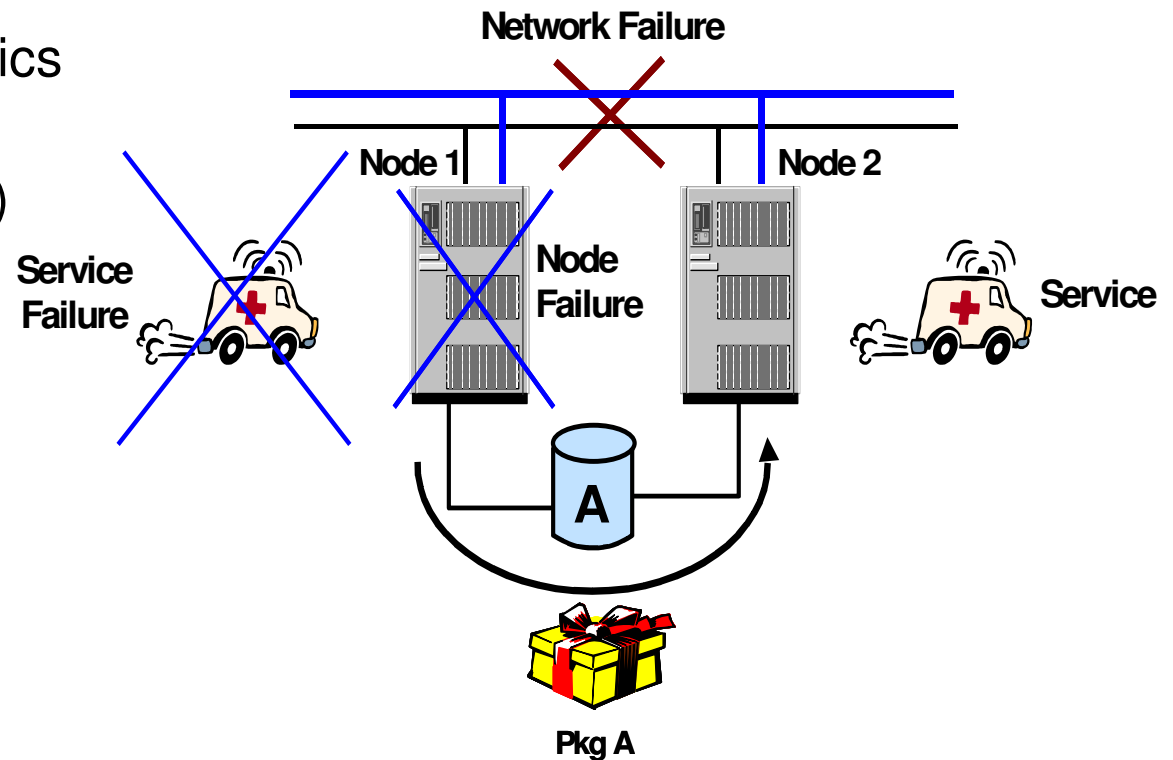
re-locatable IP address(es)



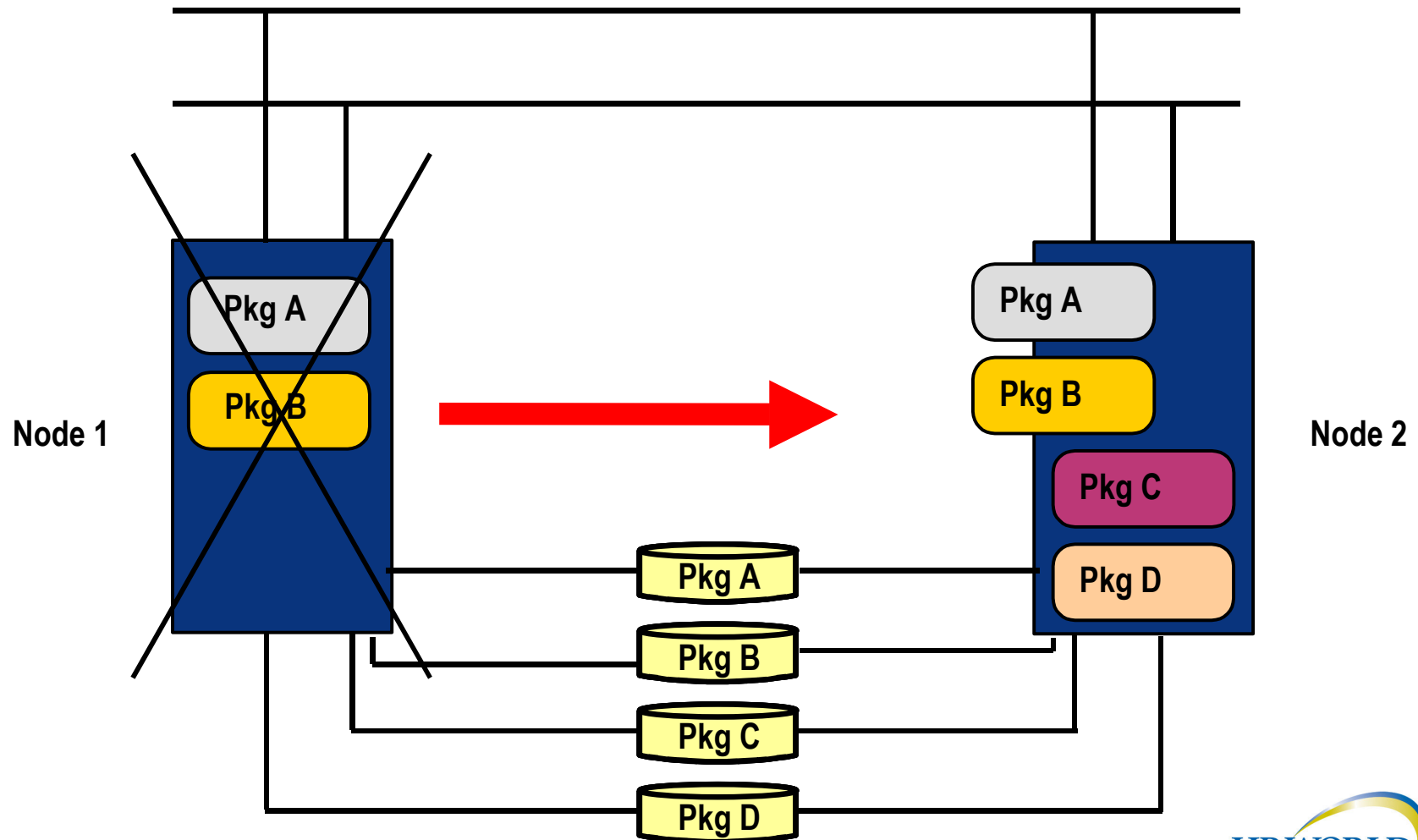
- Local IP relocation minimizes downtime
- IP at package level enables fail over to multiple nodes minimizing performance impact

cluster failure detection

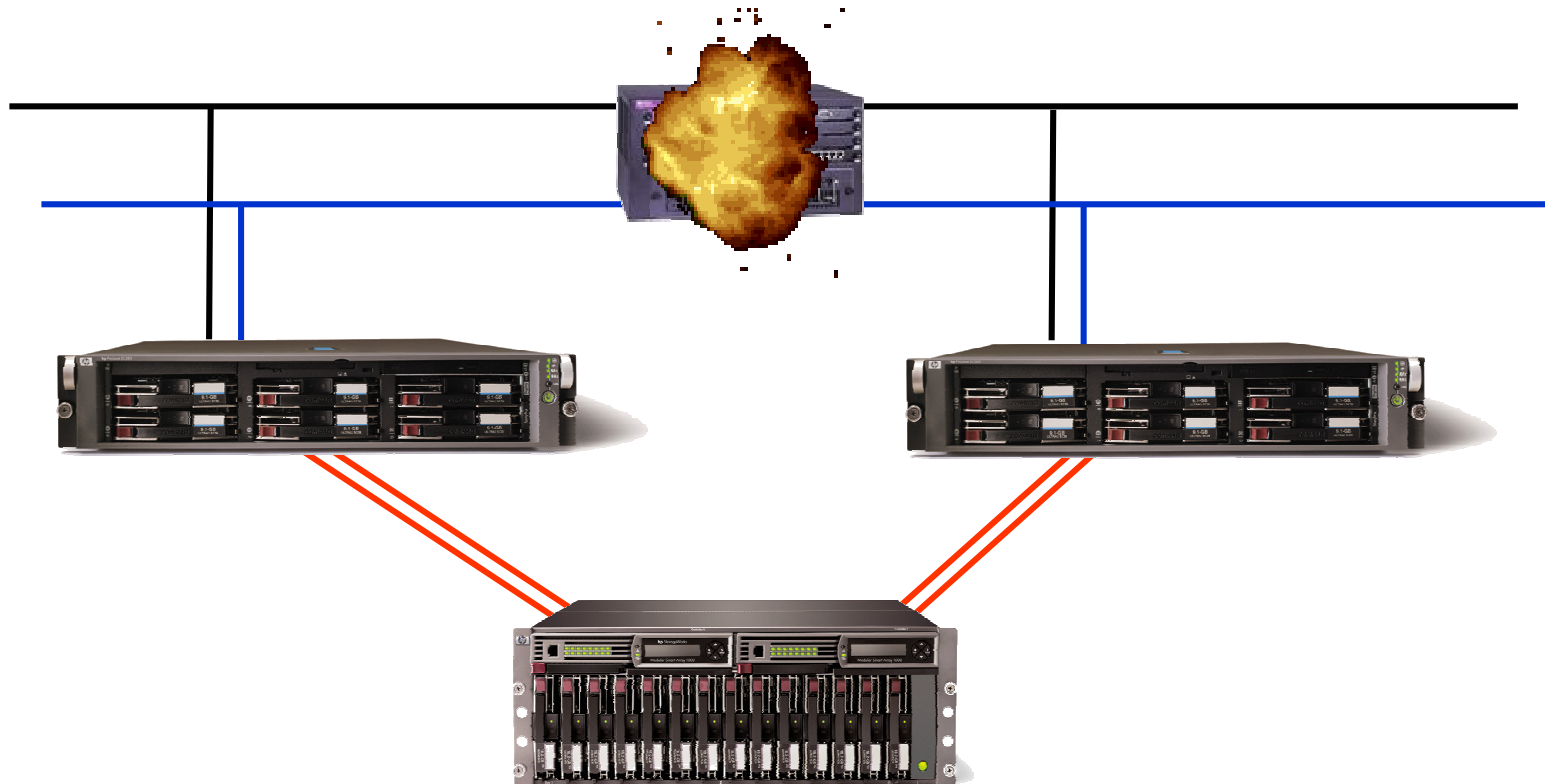
- Node failures
 - Operating system panics
 - Hardware failures
 - Pause failures (hangs)
- Network failures
 - Interface card failure
 - Cable disconnection
 - Loss of termination
- Service failures
 - Processes failure or termination



node failure



split brain situation



Cluster services cannot communicate any more → both nodes try to take all cluster packages

cluster lock

- Serviceguard was designed to prevent split brain by using one of the following cluster lock (or tie breaking) methods:
 - **cluster lock lun**
 - **quorum server**

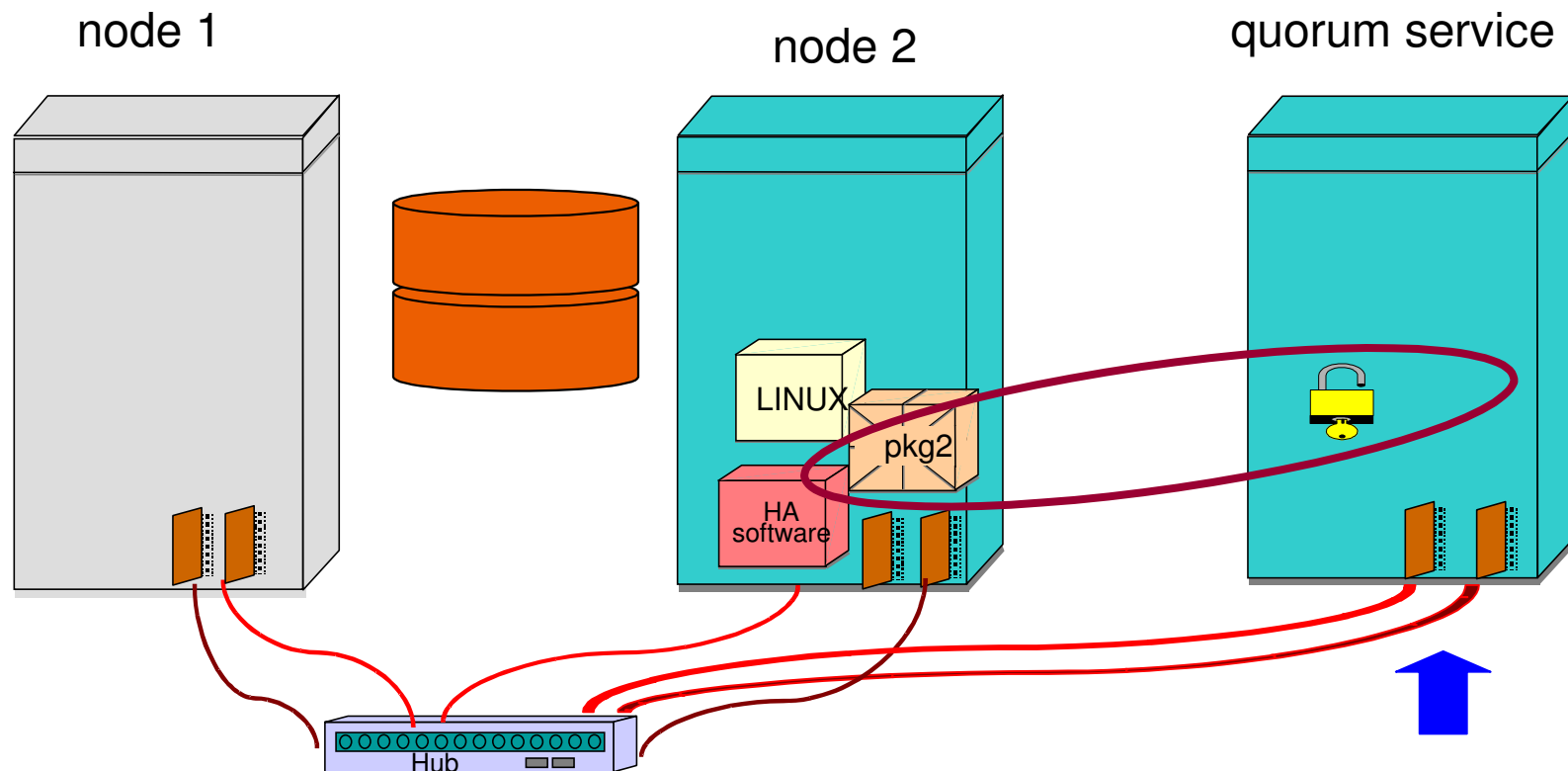
cluster lock → quorum server

- The QS is not running on a cluster member node.
- QS uses TCP/IP, and it listens at a known port (1238) for connection requests from the Serviceguard (SG) nodes
- QS machine can be used to run other applications besides QS
- QS can run as a SG package in one cluster to provide tie-breaking service for another SG cluster. Do not use reciprocal packages
- A single QS can be used as a cluster lock for multiple clusters. Maximum number of clusters supported is **50** and the total number of nodes supported is **100**.
- Heterogenous configurations are supported (HP-UX and Linux)
- The QS must be present at cluster configuration time (cmapplyconf)
- The QS must be present at cluster reforming time, when remaining nodes cannot build a quorum (less/equal 50% of nodes)
- QS is **required** for **2** node clusters, optional for **3** to **16** node clusters

cluster lock → lock lun

- Serviceguard for *Linux* supports a cluster lock LUN
 - Available starting with SG/LX version **A.11.15**
 - Supported for **2** to **4** node clusters
 - Configured on a shared lun with a Linux partition of 100k in size
 - This 100k lock LUN partition cannot be used for application data

quorum service arbitration



- Ensures data integrity
- 50 heterogeneous Linux clusters or 100 nodes supported by one QS

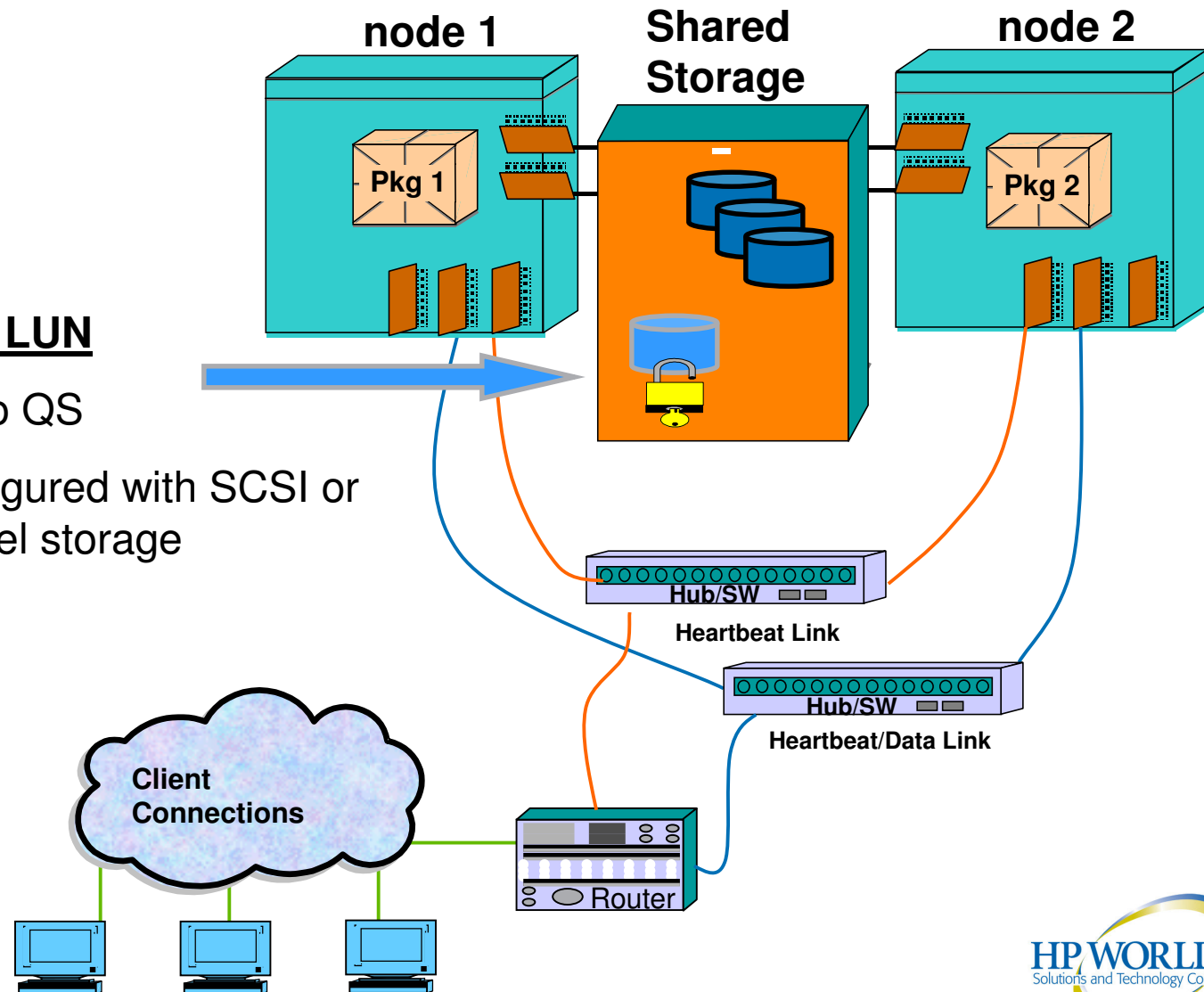
Bonding (Linux) can be configured on the QS for additional redundancy

cluster lock LUN



Cluster Lock LUN

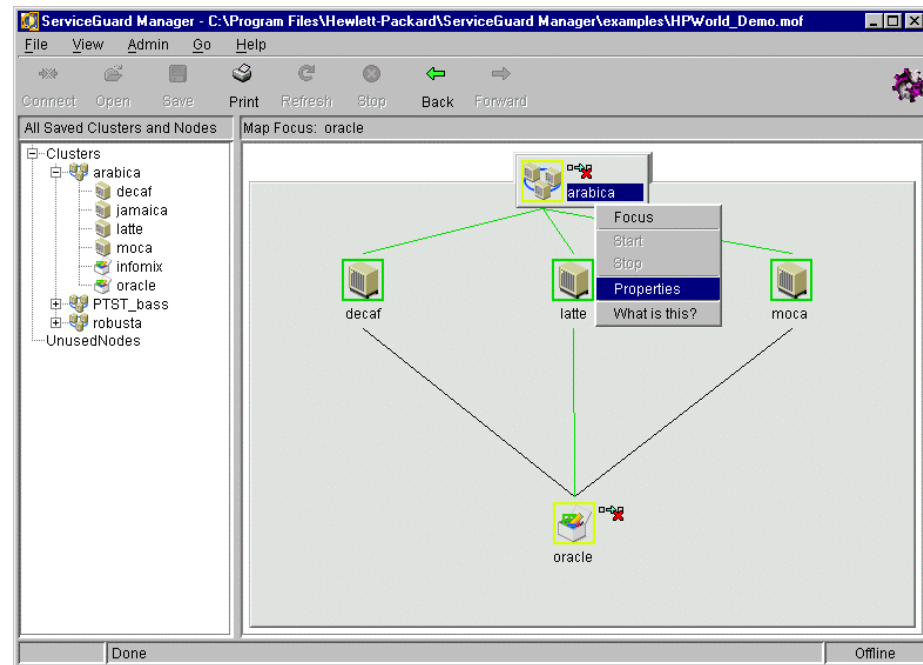
- Alternative to QS
- Can be configured with SCSI or Fiber channel storage



management interfaces

Serviceguard Manager:

- Java GUI
- uses cluster object manager (no SNMP)
- supports many platforms



command line interface:

cmapplyconf, cmcheckconf, cmdeleteconf, cmgetconf,
cmhaltcl, cmhaltnode, cmhaltpkg, cmhaltserv,
cmmakepkg, cmmodnet, cmmodpkg, cmquerycl, cmreadlog,
cmreadlog, cmruncl, cmrunnode, cmrunpkg, cmrunserv,
cmscancl, cmviewcl, cmviewconf



Overview of Serviceguard for Linux on Itanium...



HP Serviceguard for Linux

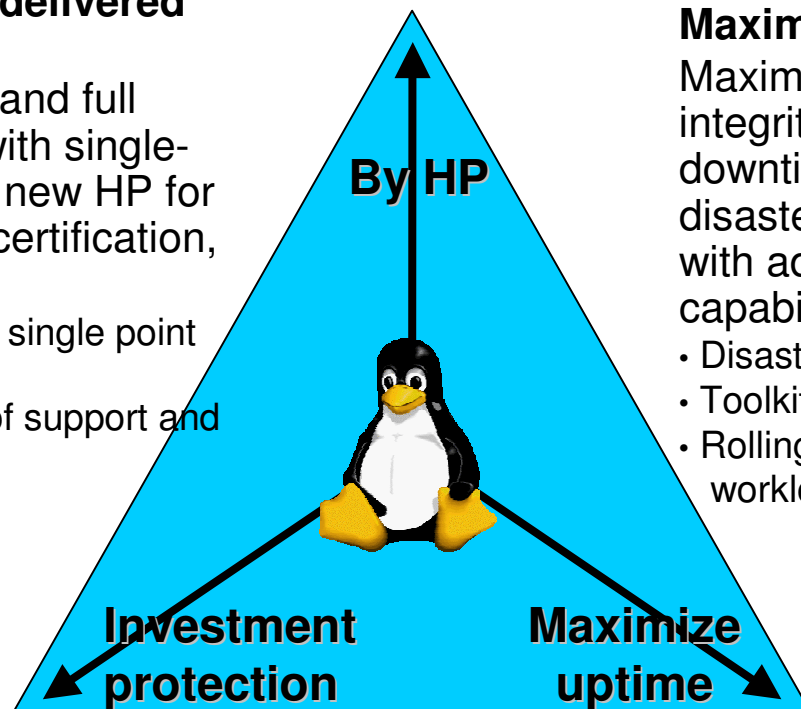
Customer benefits



Designed, developed, delivered and supported by HP

Ensure tight integration and full solution accountability with single-source service from the new HP for all hardware, software, certification, services and support

- Single point of contact, no single point of failure
- Comprehensive portfolio of support and services



Maximize uptime

Maximize service uptime and data integrity and minimize planned downtime and protect against disasters and major site outages with advanced, mission-critical capabilities at lower costs

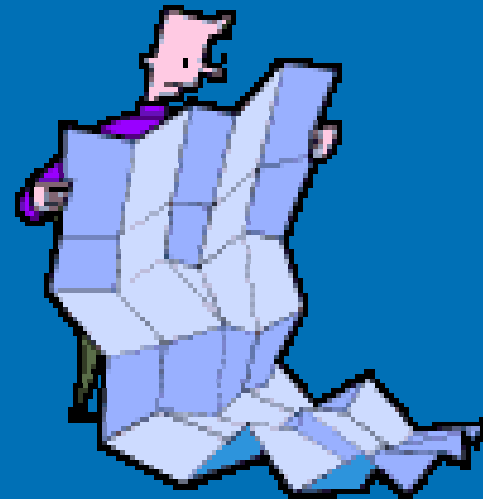
- Disaster tolerant configurations
- Toolkits for leading applications
- Rolling upgrade, online reconfiguration, workload allocation

Investment protection

- With over 80,000 licenses shipped of Serviceguard on HP-UX, HP's continued commitment is clear
- Easily integrate Linux clustering solutions into current HP-UX or multi-OS environments and quickly build Linux capabilities and expertise for mission critical applications.
- Smooth IA32 to Integrity Linux HA transition



Serviceguard for Linux Roadmap...

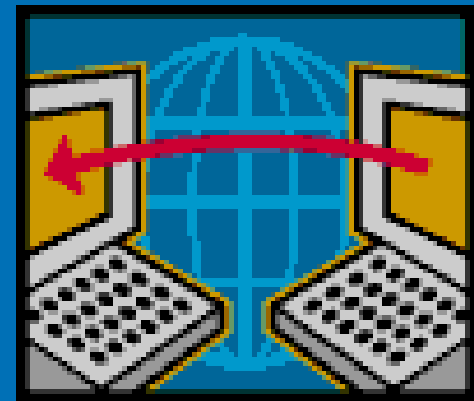


subject to change hp serviceguard for linux roadmap



- 1QFY04 (November'03)
 - 11.15.01
 - LockLUN, IPF support (SUSE/UL)
 - SGeSAP toolkit (SUSE/UL, IA32)
 - Blade Servers BL20P G2, BL40P
 - EVA, MSA storage (RH 2.1, IA32)
 - 2QFY04 (April'04)
 - 11.15.02
 - adding RedHat 3 IA32/IPF support
 - rx4640 & rx1600
 - Full EVA MSA
 - add support for SUSE/UL & RedHat 3
- ➔ MSA & EVA up to 16 nodes for RH 3 & SLES 8 with Qlogic HBA driver 7.00.3 (May '04)
- 2HFY04
 - Servers
 - DL585 (Opteron)
 - rx7620, rx8620
 - ISS G4 (Nacoma) 2-Way servers
 - DL360, DL380, ML350, ML370
 - BL20P-G3 BL25P
 - Storage
 - MSA500 G2
 - May have multipath & 4 node support
 - EMC Support – resource dependent
- 1HFY05
 - 11.16.xx
 - SLES9 (2.6 kernel)/RH3
 - Features under consideration
 - SG (GUI) Config of cluster and packages
 - RBA (Role Based Access)
 - Nimbus 2.0 support
 - Supportability enhancements
 - CFS Support – post release if possible
 - Equivalent feature to LVM exclusive activation – post release if possible
 - Linux Standards Base (LSB) – post release if possible
 - Orca (SuperDome)
 - Newer IPF servers
 - New 4-Way ISS server
 - Notes
 - BL30P – too few LAN connections
 - DL145 - too few LAN connections
 - LSB does NOT imply automatic use with other distros

Serviceguard for Linux Environments and Uses...



ideal customer environment

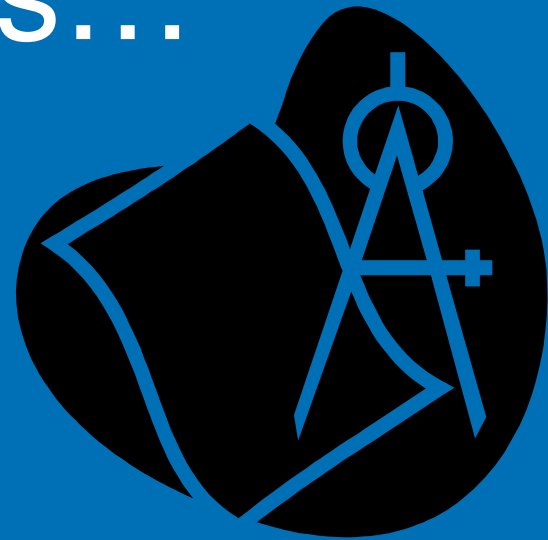
- Enterprise customers requiring low cost 24x7 availability and solution accountability
- Enterprise companies adopting Linux for business critical applications to achieve cost advantages deploying industry standard servers and that demand the same technology, support and services offered in proprietary UNIX environments

typical uses – SG/LX

- Make delivery of data and services to clients continuous
- Minimize planned and unplanned downtime for Linux applications such as:
 - Apache
 - MySQL
 - NFS
 - Oracle
 - Samba
 - Sendmail
 - SAP

Serviceguard for Linux

Design Information and Considerations...



design goals

- One code base for all Serviceguard ports
- Serviceguard for Linux first release was A.11.13
current release A.11.15.02
- Full functionality of HP-UX version for the long term
- Standard HP software support
- No kernel modifications **-or-** modifications are put under GPL
- Usage of additional open source software like:
 - Linux LVM
 - Linux Software RAID (md)
 - Linux bonding driver
 - ReiserFS





distribution support

A.11.15.02:

- RedHat EL AS 3.0 U1



A.11.15.01:

- SuSE ES8 UL 1.0 with kernel 2.4.21 of SP3



Dependencies:

- Linux LVM for kernel
- ext2, ext3 filesystem and ReiserFS



Maximum Cluster :

- 16-node cluster with fiber-channel shared storage



cluster size & CDs

Maximum Cluster :

- 16-node cluster with fiber-channel shared storage

Distribution:

- 2 CDs: with the following product structure
 - 1 - Serviceguard Distributed Components
 - (including SG Mgr, QS and Release Notes)
 - 2 - Serviceguard for Linux
 - (including SG/LX A.11.15, patches, documentation)

hardware support

- **servers:**
 - rx1600, rx2600, rx4640 and rx5670
- **storage:**
 - XP48, XP128, XP512, XP1024
 - VA7x00
 - EVA3000/5000
 - MSA1000
- Check the latest configuration guide for the most up-to-date information

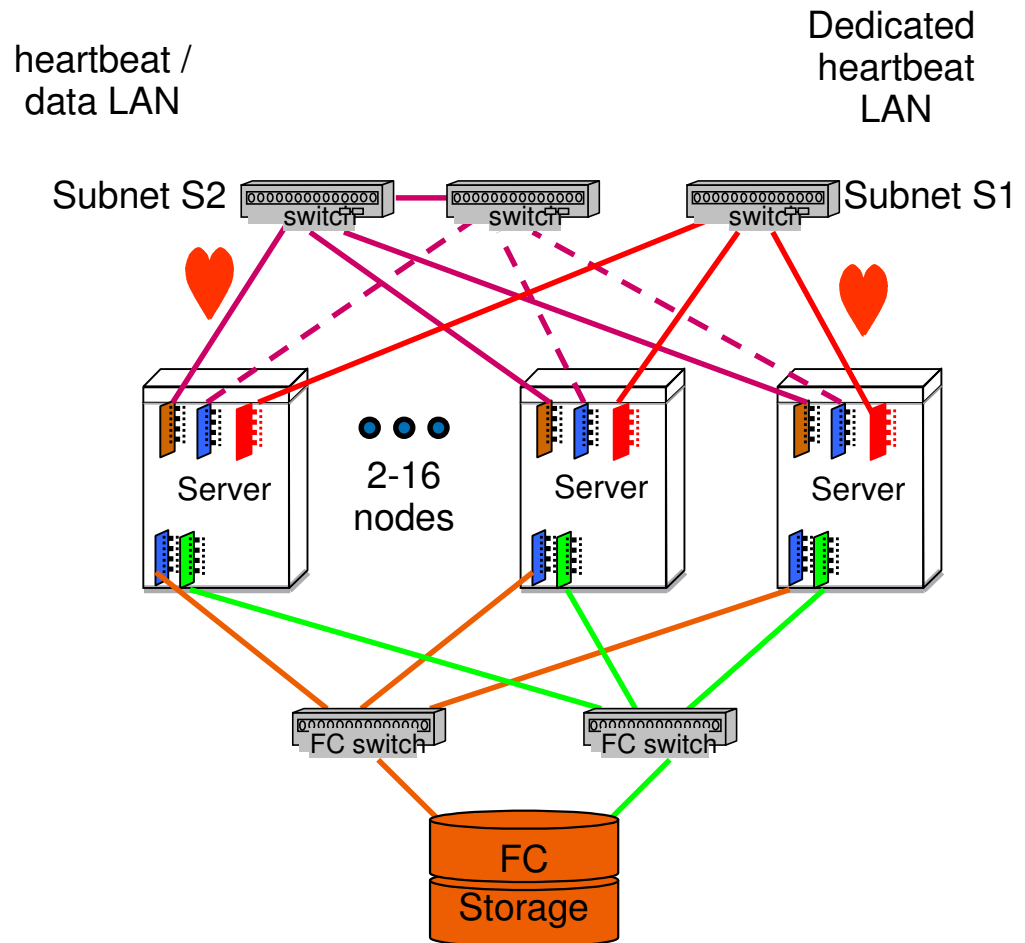
HP Serviceguard for Linux

Fiber-channel based configurations



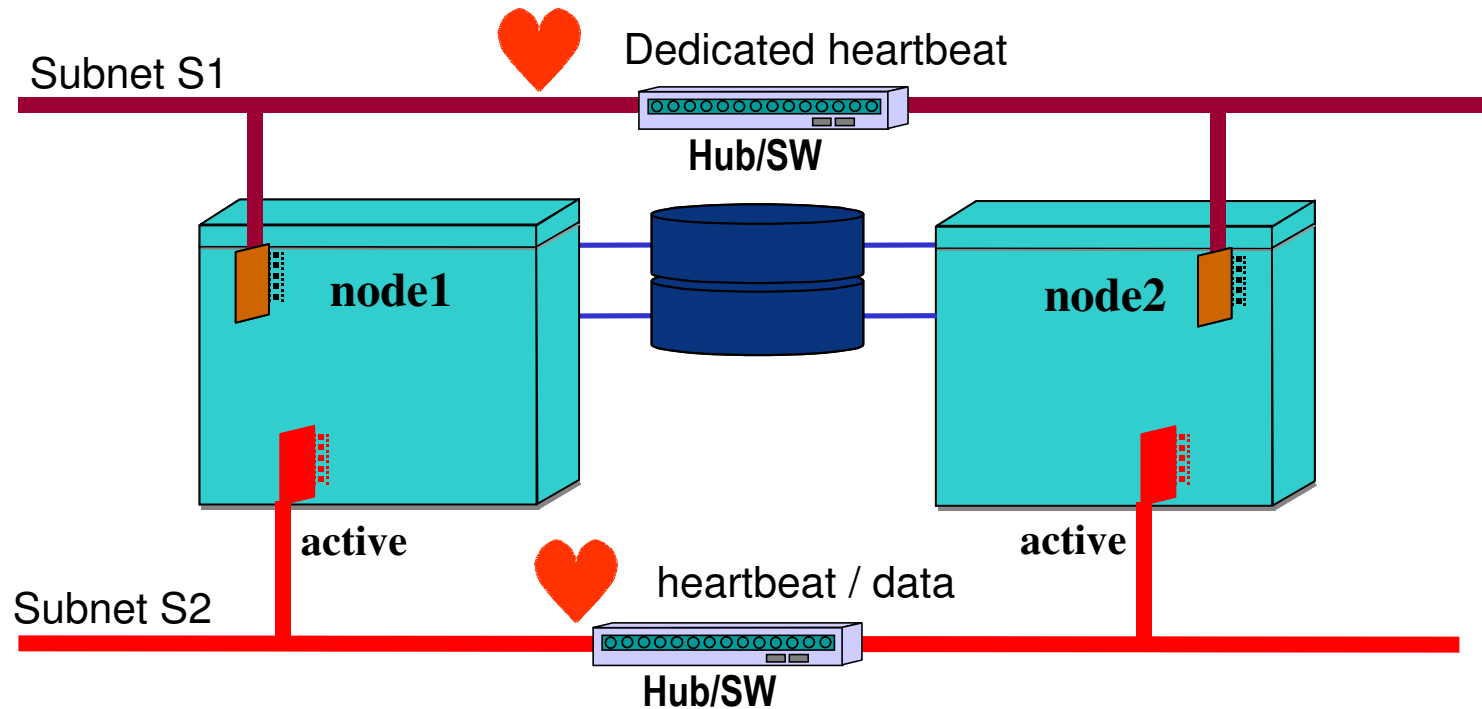
Server support

- rx1600
- rx2600
- rx4640
- rx5670



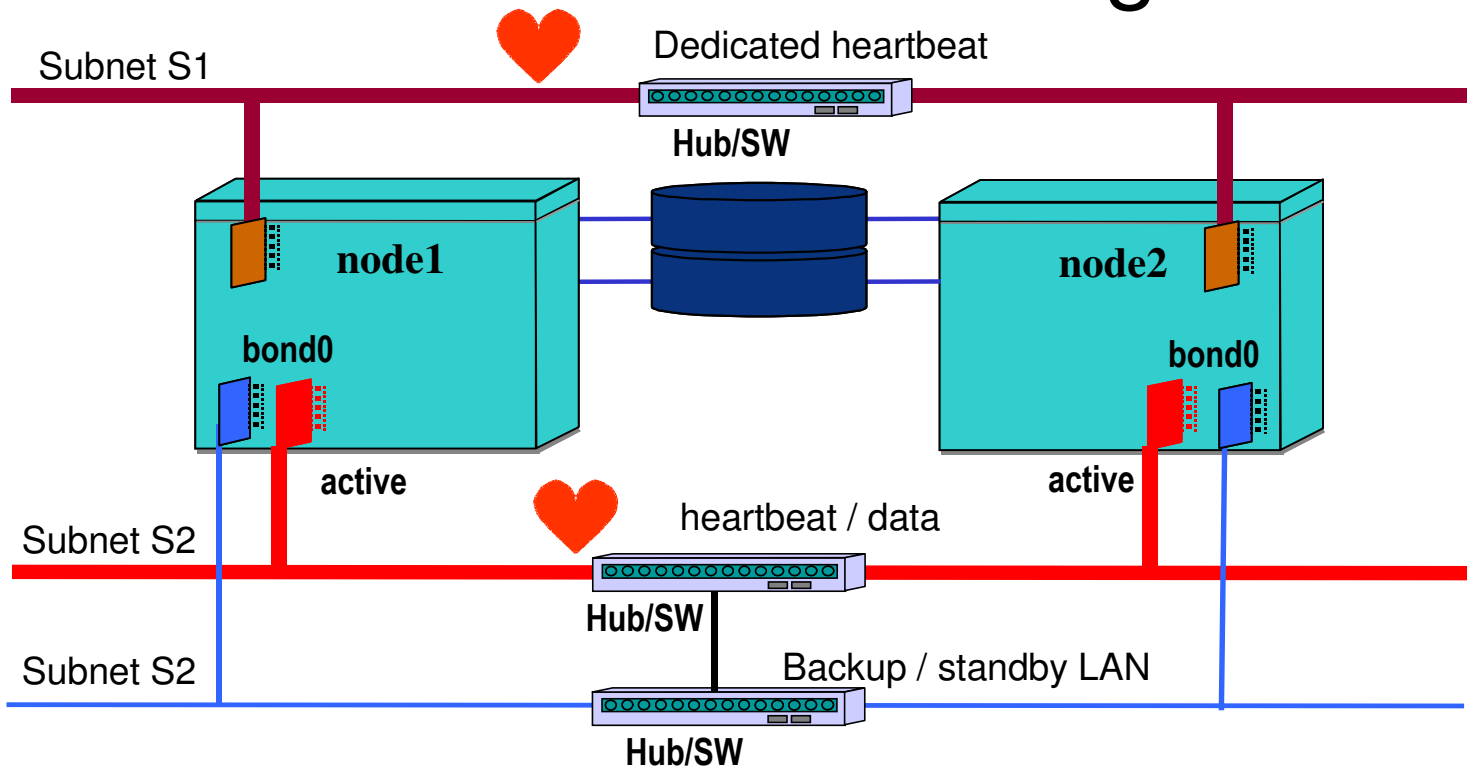
Note: No mixed IA32/Itanium clusters

minimum LAN configuration



- Two LAN interfaces are required to eliminate single points of failure (active/active)
- Or Bond the two LAN interfaces (active/standby)

recommended LAN configuration

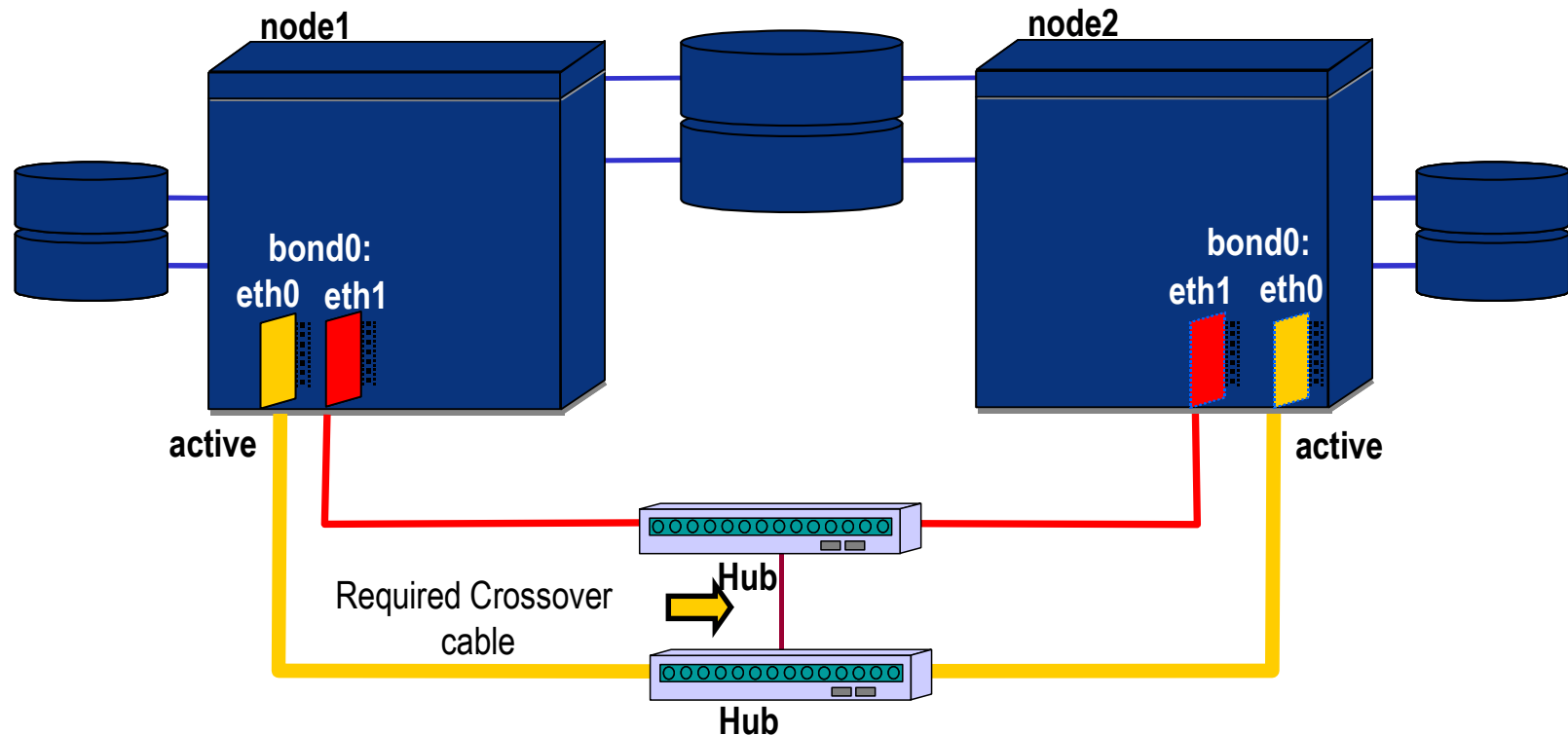


- The recommended configuration uses a dedicated heartbeat, data heartbeat, and backup connection

bonding driver

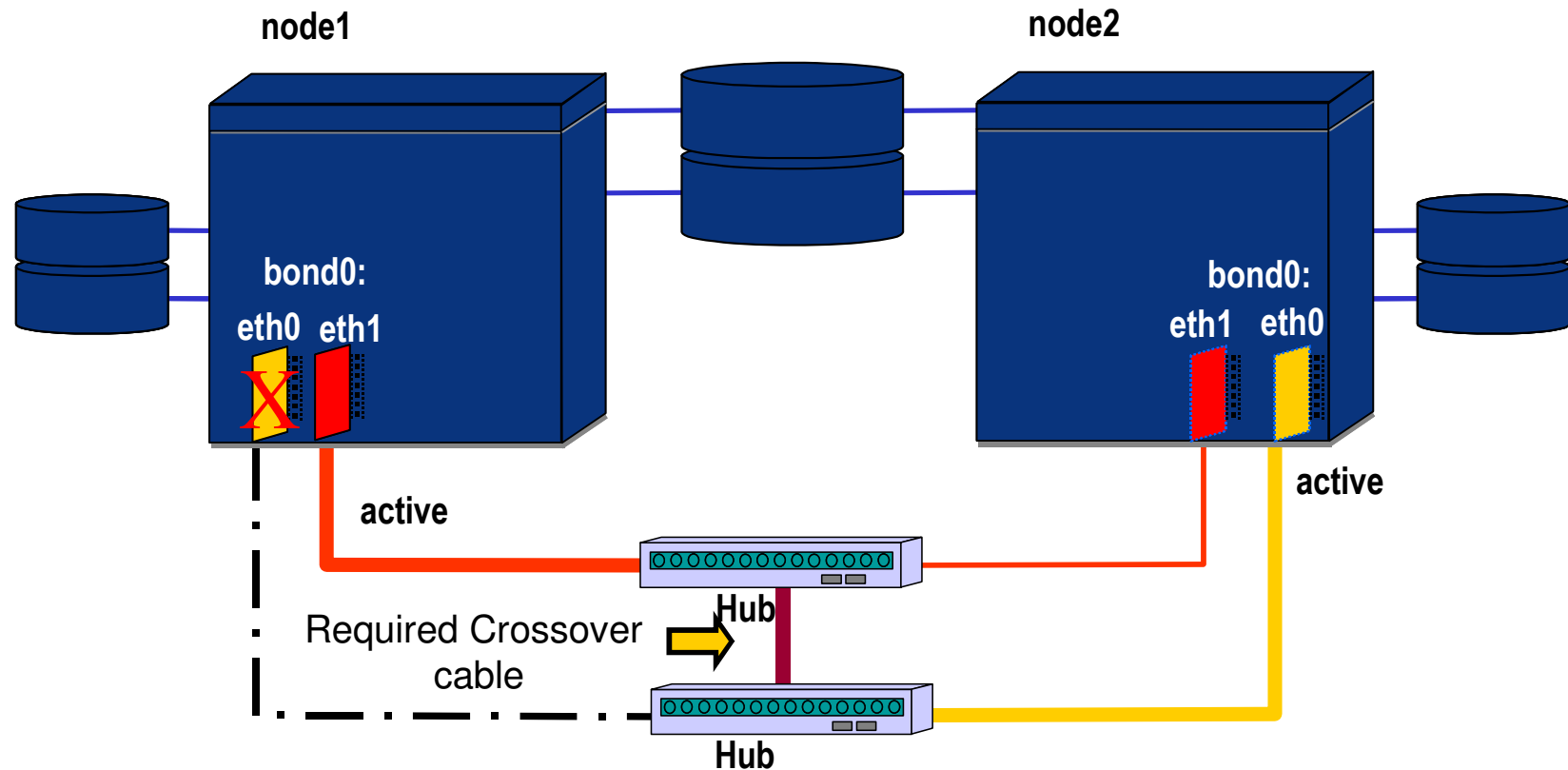
- The bonding driver originally came from Donald Becker's beowulf patches
- Its basic purpose is the bundling of network interfaces like HP's Autoport Aggregation
- Bonding is part of the official kernel distribution since kernel 2.2.18.
- Recommended to load as a module in order to be able to pass parameters to the driver
- Supports HIGH AVAILABILITY (=1) and LOAD BALANCING (=0) mode
- Supports any type of ethernet interface, a bond can use different cards at different speed
- HIGH AVAILABILITY requires **MI** state reporting supported by the card and the driver
- LOAD BALANCING requires switches with trunking capability
- More than 2 physical slave cards can be combined in a bond

bonded LAN ICs before failure



- Individual variable Ethernet interfaces are slaves and the bond is master
- The miimon controls how often the bonding driver monitors the status of NIC cards
- The recommended value for miimon is 100 milliseconds

bonded LAN ICs after failure



- In the failover process, all network stacks for the failed active card are maintained
- No TCP connections are broken

single pathing → disk monitoring

- Single path connections to disks build a **SPOF** which might not be discovered by standard hp Serviceguard monitoring daemons
- **disk monitoring**
Is a standard service which can be used for each package to continuously monitor the disks required for the volume group(s) used by the package

```
SERVICE_NAME[0]="cmresserviced_Pkg1"  
SERVICE_CMD[0]="cmresserviced /dev/sdd1 /dev/sde1"  
SERVICE_RESTART[0]=" "
```

LVM implementation

- Developed by Heinz Mauelshagen during an IBM project at German Telekom in Darmstadt
- Implementation is more than 90% compatible with HP-UX plus some enhancements and some IBM LVM features
- The CLI is a superset of HP-UX LVM with similar option parameters
- Linux LVM uses the same directory and naming scheme as HP-UX
- Since kernel version 2.4 official part of the kernel distribution (SuSE)
- The latest version is 1.0.9 (still developed by Heinz Mauelshagen)
- The LVM project found a technical home at <http://www.sistina.com>
which is **now** part of RedHat

LVM implementation (continued)

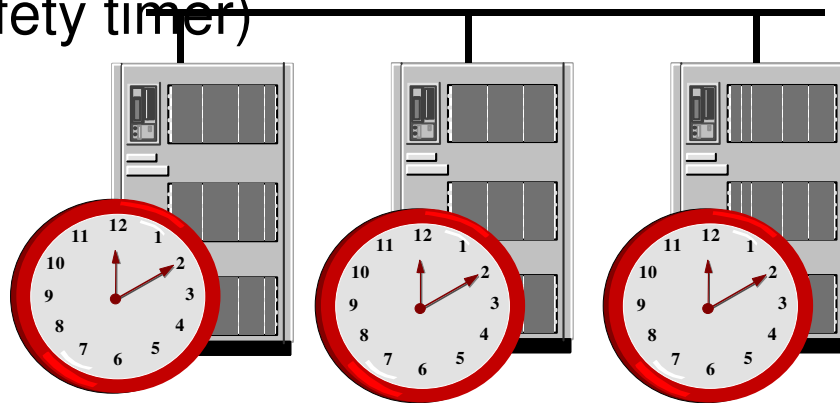
- Needs to be compiled into the kernel (already is – SuSE & RH)
- Kernel is able to boot from LVM based root disks by using a specific RAM disk layout for the boot process
- Physical volumes can be of any type of disk partition
- LVM can be combined with Linux software RAID
- Linux LVM uses the **/proc** filesystem as interface for the CLI

Limitations:

- No PV links (HP-UX LVM multipathing)
- No LVM based disk mirroring (HP-UX Mirror/DiskUX)
- No cluster aware volume groups (LVM exclusive activation)

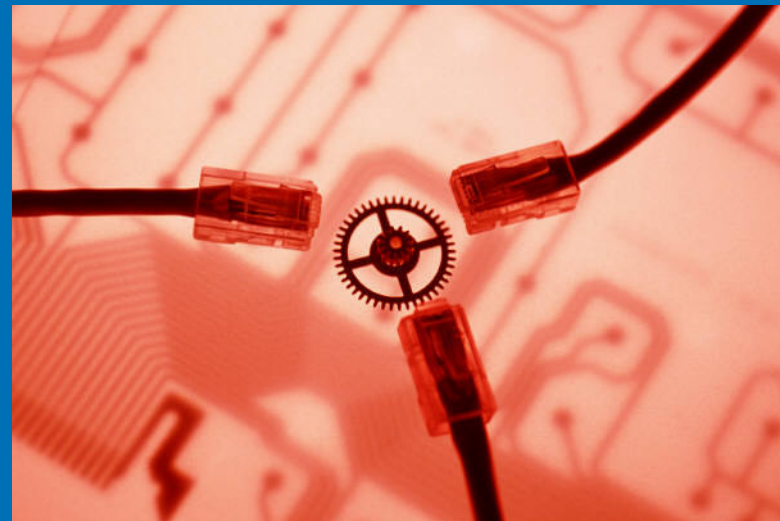
deadman timer

- Situations that could result in data corruption (split-brain)
 - Cluster daemon fails, but high-availability services continue running on that node
 - Node pauses, causing other nodes to believe the node has failed
- System reset/reboot occurs to protect the integrity of your data when
 - A cluster node cannot communicate with the majority of cluster members for the predetermined time
 - There is a kernel hang, a kernel spin, or a runaway real-time process
 - The Serviceguard cluster daemon fails
- Serviceguard for Linux relies on a kernel-implemented deadman timer (safety timer)



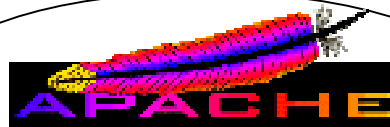

Serviceguard for Linux

Application Integration...



application integration

- Products
- Contributed Toolkits



application integration

- **products:**

- hp Serviceguard Extension for SAP for Linux
- hp Serviceguard for Linux Oracle database toolkit

- **contributed toolkits:**

- Apache
- MySQL
- NFS
- Samba
- SendMail
- PostgreSQL

} free download from
<http://www.software.hp.com>

Case Study

the company...



the company



- Developer and manufacturer of generic pharmaceuticals
- 30 years in operation
- Six locations in the US and Puerto Rico
- Recently acquired a long-established pharmaceutical company
- Now faced with two distinct IT platforms in two distinct locations to manage and maintain

the IT department

- Company typically hires recent college graduates into their IT organization
- Lots of open source knowledge
- Lots of open source “desire”
- Now operating in two separate locations in the US
- Needs to consolidate operations and simplify management

IT - hardware and OS

- HP Proliant servers (*Through acquisition*)
 - Windows
- Other IA-32 servers (*Through acquisition*)
 - Red Hat Enterprise Linux Advanced Server 3
- HP Integrity servers
 - HP-UX

the application at issue

- Client access via web interface
- Apache Web Server frontend
- MySQL backend
- Red Hat Enterprise Linux AS 3
- Currently running on high-end IA-32 server

What is the problem?





“This application is a critical component the acquired company’s revenue-booking process. We like the application and want to use it for the combined company. But, it needs to handle three times the current load and storage, and be available 24x7.”

CIO, Pharmaceutical Development & Manufacturing Company



the problem

- Load (and storage requirements) to be placed upon the application will increase threefold.
- Salespeople want remote connectivity and 24x7 availability of the application
- Like all salespeople, they wait until the last week of the time period to book their sales
- Application downtime at end of month translates to revenue which must be booked in the “next” month
- IT does not want to provide on-call support during off hours
- Application needs to automatically restart in event of application or hardware failure

constraints

- Cost
 - application already works on current platform; no need to invest major sums of money on reinvention of the wheel (i.e. a rewrite of the application)
- Time
 - for every month that the app is “un-protected”, the risk of non-booked revenue continues
- Existing Hardware and Infrastructure
 - reuse as much as possible
- Training and skills of current staff
 - utilize existing skill-sets

What is the solution?



the solution

- HP's Serviceguard for Linux on Itanium
 - two-node cluster of 4-way servers
 - shared disk storage
 - redundant LAN ICs
 - redundant switches and subnets

alternatives

- Hardware (processor)
 - Do we want two entry-level servers, with the thought that we can upgrade in the future?
 - Do we want to skimp and get one server of lesser capability than the “primary” server?
 - Can we run in “degraded” mode?
 - For how long?
- Hardware (shared disk)
 - App is running on internal disk now
 - Need shared disk for SG/LX cluster
 - Fiber channel is supported, but is more than what is needed for this solution (\$ and capacity)

alternatives (continued)

- Linux distro
 - Already running on RHEL AS 3, which is supported by SG/LX A.11.15.02
 - RH is frontrunner in North America
 - Novell acquired SuSE in Jan 2004
 - Business need to justify switch to SuSE ?
- How highly-available can we afford to be?
 - No SPOFs?
 - One SPOF (non-critical, hopefully)?
 - Several SPOFs (non-critical, hopefully)?
 - Fiber channel for supportability?

considerations

- Hardware (processor)
 - Make our lives as easy as possible
 - Failover server of same type (rx4640-4) will provide ability to run in degraded mode for infinite length of time, at least where capacity is concerned
- Hardware (shared disk)
 - Migrate app to external disk on shared bus
 - Fiber channel is supported by HP RC, but...
 - ds2300 with redundant LAN ICs, good backup strategy and disk mirroring will give what is needed

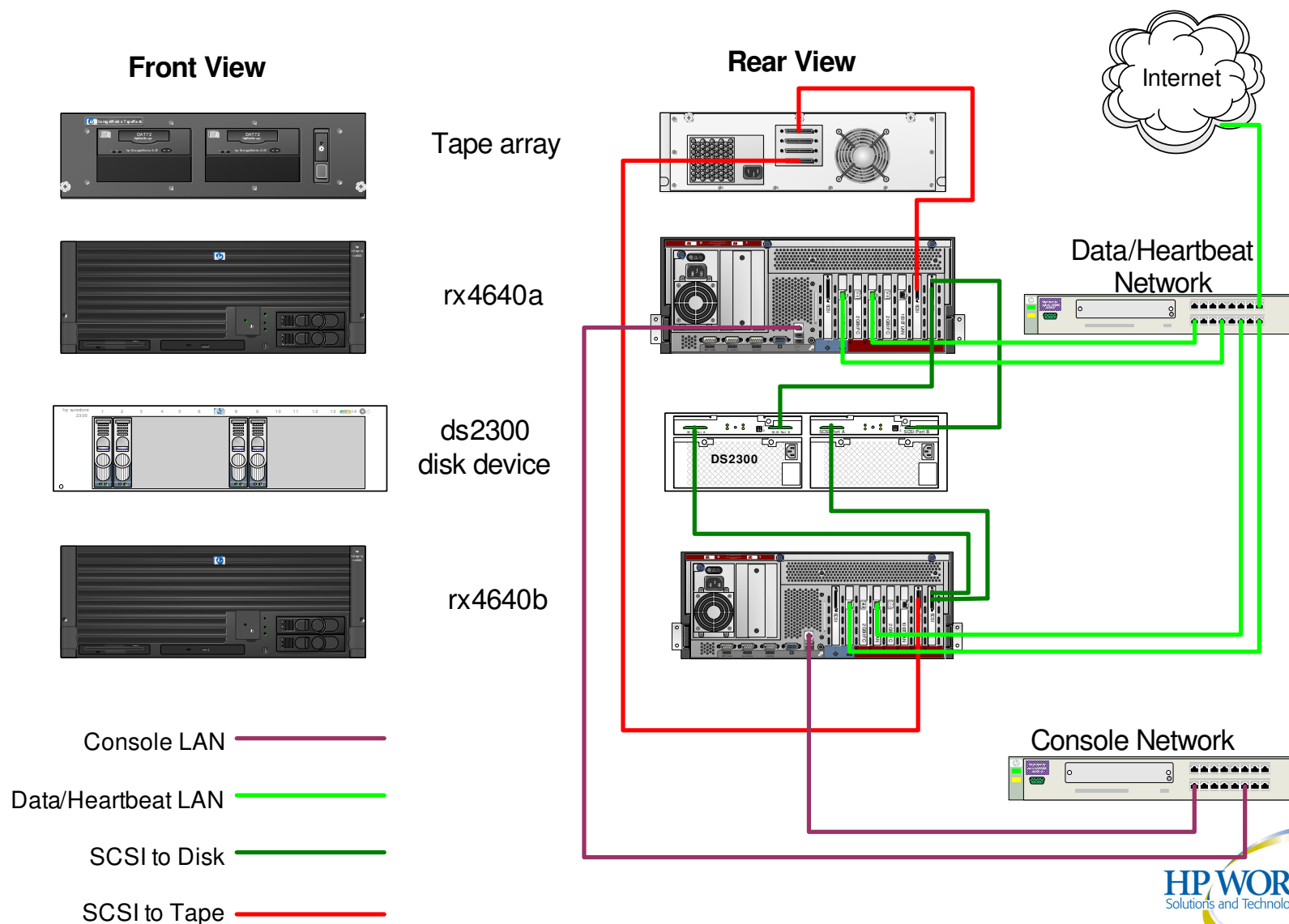
considerations: Red Hat vs SuSE

- Red Hat Strengths
 - IS organization and vendor acceptance
 - OSS community standing
 - Investment backing and cash
 - US market share
- Red Hat Challenges
 - Pricing
 - Organizational scaling and response
 - Countering SuSE's software strengths
 - Desktops

considerations: Red Hat vs SuSE

- SuSE Strengths
 - Novell value proposition
 - Multiple Architecture Support
 - Technical Skills
 - Contract negotiations
 - Channel partners
- SuSE Challenges
 - US penetration
 - ISV support
 - Channel training
 - Positioning
 - OSS relationship

hardware architecture



Serviceguard for Linux configuration



- Install Linux Distribution (Red Hat Enterprise Linux AS 3 U1)
- Serviceguard for Linux Installation
 - Serviceguard Installation tips
- Preparing the systems and security files
- Install and configure Cluster Lock LUN
- Configure Channel Bonding
- Configure the logical volume infrastructure
- Configure a cluster
- Configure packages
- Failure scenarios

installing Linux and HP Serviceguard



- Linux is loaded from a set of distribution CDs that includes the vendor's bootable CD and a group of Red Hat CDs.
 - “Getting Started with HP Serviceguard A.11.15 for Linux”
- Serviceguard product and a variety of related components are shipped as Red Hat Package Manager files (RPMs) on the SG/LX distribution media. The following directories are included on the CD:
 - *Red Hat*, which includes RPMs for ServiceGuard A.11.15.02 and ServiceGuard Cluster Object Manager B.02.02.01.
 - *contrib*
 - *documentation*, which includes English and Japanese versions
 - HP ServiceGuard for Linux product manuals and release notes.

Serviceguard installation

- Use rpm to install the ServiceGuard product.
- Use the following commands to install ServiceGuard for Red Hat Linux on *HP Integrity Servers*.

For Red Hat:

```
# cd /mnt/cdrom/RedHat/rpms
```

```
# rpm -i sgcmom-B.02.02.01-0.product.redhat.ia64.rpm
```

```
# rpm -i serviceguard-A.11.15.02-0.product.redhat.ia64.rpm
```

When the commands are complete, the prompt appears.

system setup

- Enable XDMCP for your display manager (e.g. gdm, kdm)
- putty
 - used for system access via ssh
- vi .bashrc
 - PATH=\$PATH:/usr/local/cmcluster/bin
 - exec bash
 - echo \$PATH
- vi /usr/local/cmcluster/conf/cnclnodelist
 - rx4640a root
 - rx4640b root

building the fiber channel driver

- Fiber Channel driver needs to be installed and built if you will be using Fiber Channel storage.
- HP Integrity Servers are supported with the A6826A host adapter. Documentation and drivers are available through the HP web page:
 - <http://www.hp.com/>
 - click on **support and drivers** and enter the appropriate product number (**A6826A**).

preparing your systems



All cluster nodes must have

- Appropriate security files
- Kernel configuration
- NTP configuration

Cluster File Locations for Red Hat

```
##### cmcluster.conf #####
```

```
#####
```

```
SGROOT=/usr/local/cmclusters                # SG root directory
SGCONF=/usr/local/cmcluster/conf             # configuration files
SGSBIN=/usr/local/cmcluster/bin               # binaries
SGLBIN=/usr/local/cmcluster/bin               # binaries
SGLIB=/usr/local/cmcluster/lib                # libraries
SGRUN=/usr/local/cmcluster/run                # location of core dumps
SGAUTOSTART=/usr/local/cmcluster/conf/cmcluster.rc # SG Autostart file
```



preparing your systems (continued)



To Enable use of ServiceGuard variables, include the file /etc/cmcluster.conf in your login profile for user root.

```
# . /etc/cmcluster.conf
```

When this command completes, the prompt appears. You can confirm the access to the one of the variables as follows:

```
# cd $SGCONF
```


editing security files (1 of 2)

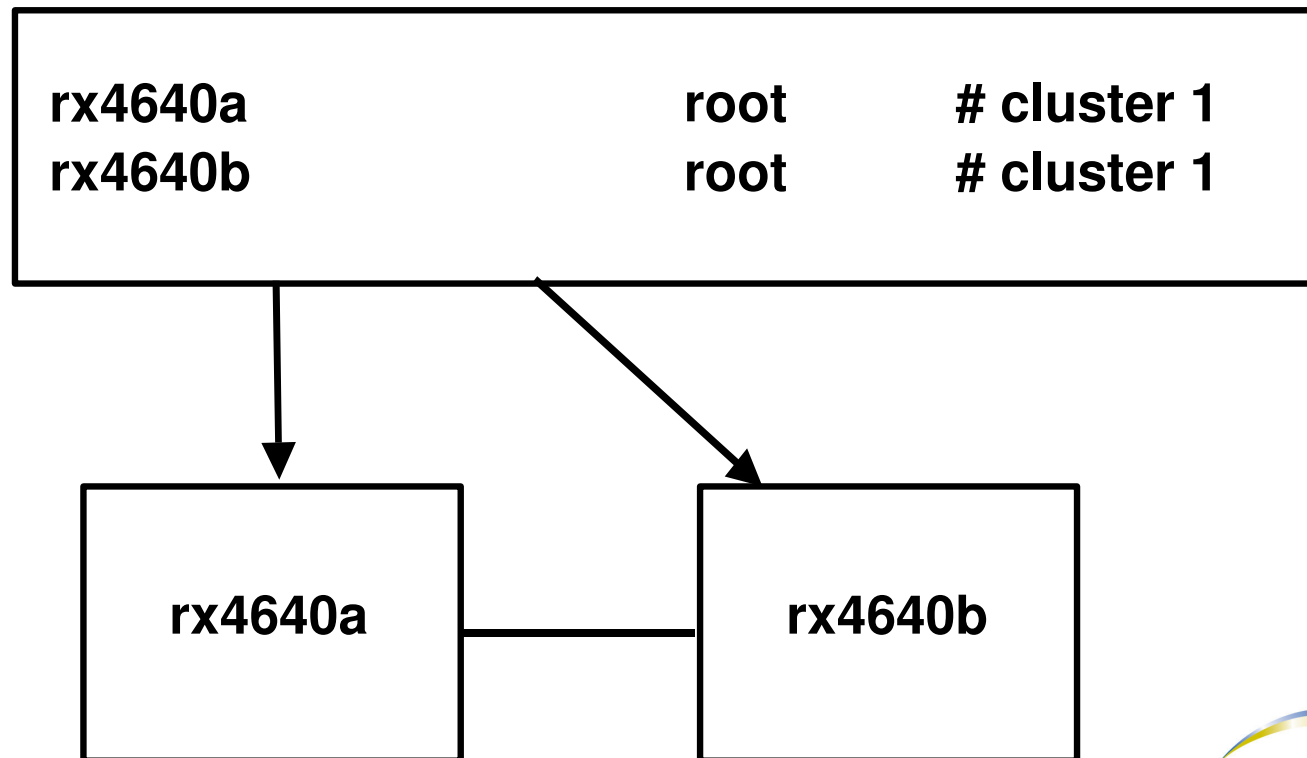


Files — /root/.rhosts
or /usr/local/cmcluster/conf/cmclnodelist

Format of files

[hostname or IP address] [rootuser] [#comment]

**Include all
nodes in
cluster**

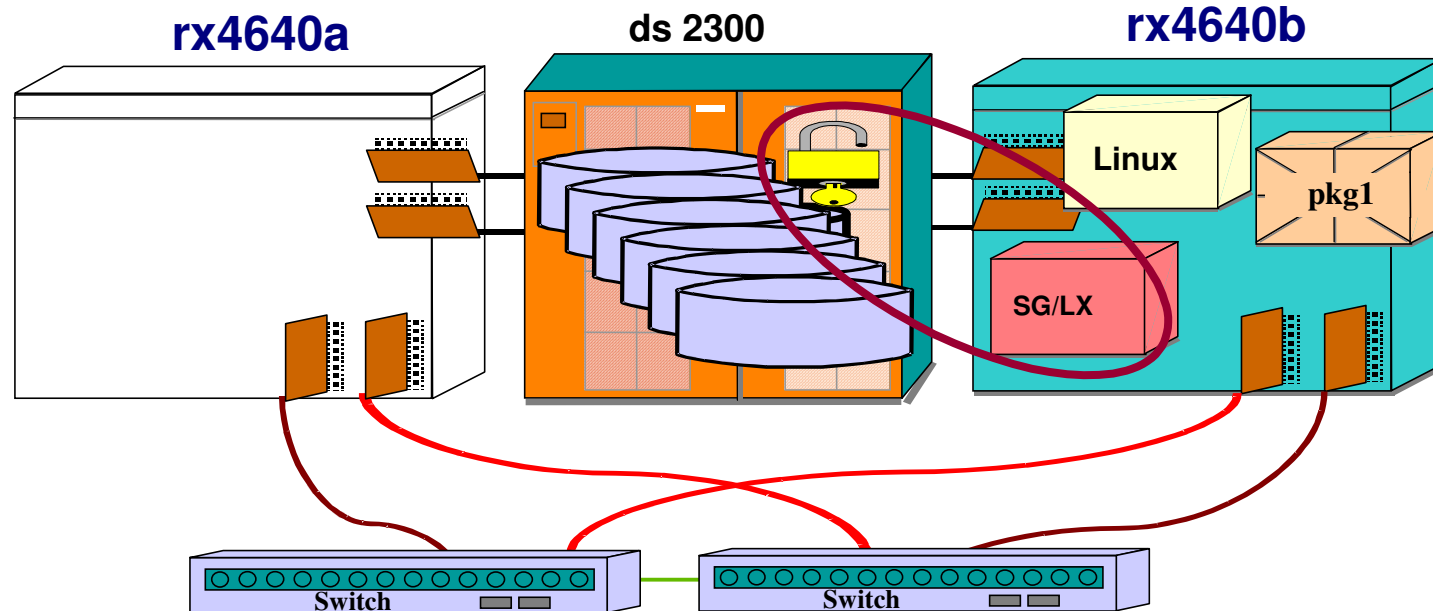




editing security files (2 of 2)

- Serviceguard uses ARPA services to ensure secure communication among cluster nodes
- Identify nodes that permit root user access on other nodes
- Serviceguard checks for the existence of the `$SGCONF/cmclnodelist` file first
 - If it exists, Serviceguard uses it to verify access
 - If it does not, it uses `/root/.rhosts`
- Copy the file to all nodes
- Serviceguard supports full domain names in both files

cluster lock — Lock LUN



- Cluster re-formation requires a majority of the nodes
- Exactly 50% of the nodes can re-form as a new cluster if the other 50% of the nodes do not also re-form
- The cluster lock LUN is used as a tie-breaker when
 - A running cluster fails
 - Serviceguard attempts to form a new cluster
 - The cluster is split into two subclusters of equal size

setting up the Lock LUN

- Lock LUN configuration
 - Configured on a disk physically connected to all cluster nodes
 - The lock LUN is dedicated for use as the cluster lock (is not configured as part of a LVM volume group)
 - Requires a partition (Linux type 83) of size 100K
 - Configured as part of the cluster configuration file
 - Cannot be added to the cluster online

Lock LUN – detailed steps

- Create the lock LUN partition
- - # fdisk /dev/sdb
 - p /* look for existing partitions */
 - d /* delete, if necessary */
 - n /* add new partition */
 - primary
 - partition 1
 - start cylinder 1
 - end cylinder 1
 - type /* Linux – default */
 - 83 /* select this type */
 - p /* print */
 - w /* write partition data */
- Cluster Configuration file parameters:
 - CLUSTER_LOCK_LUN /dev/sdb1

creating partitions

Prompt	Response	Action Performed
Command (m for help)	n	1. Create a new partition
Command action e extended p primary partition (1–4)	p	2. Create a primary partition
Partition number (1–4)	1	3. Create partition 1
First cylinder (1–nn, default 1)	Enter	4. Accept the default starting cylinder 1
Last cylinder or +size or +sizeM or +sizeK (1–nn, default nn)	Enter	5. Accept the default, which is the last cylinder number
Command (m for help)	P	6. Display partition data
Command (m for help)	w	7. Write data to the partition table

create data partitions (1 of 2)

– first package: apache

–

fdisk /dev/sdb

- primary
- partition 2
- start cylinder 2
- end cylinder +2000M /* 2 GB */
- t /* change partition's system ID */
- 8e /* Linux LVM */
- p /* print */
- w /* write partition data */

create data partitions (2 of 2)

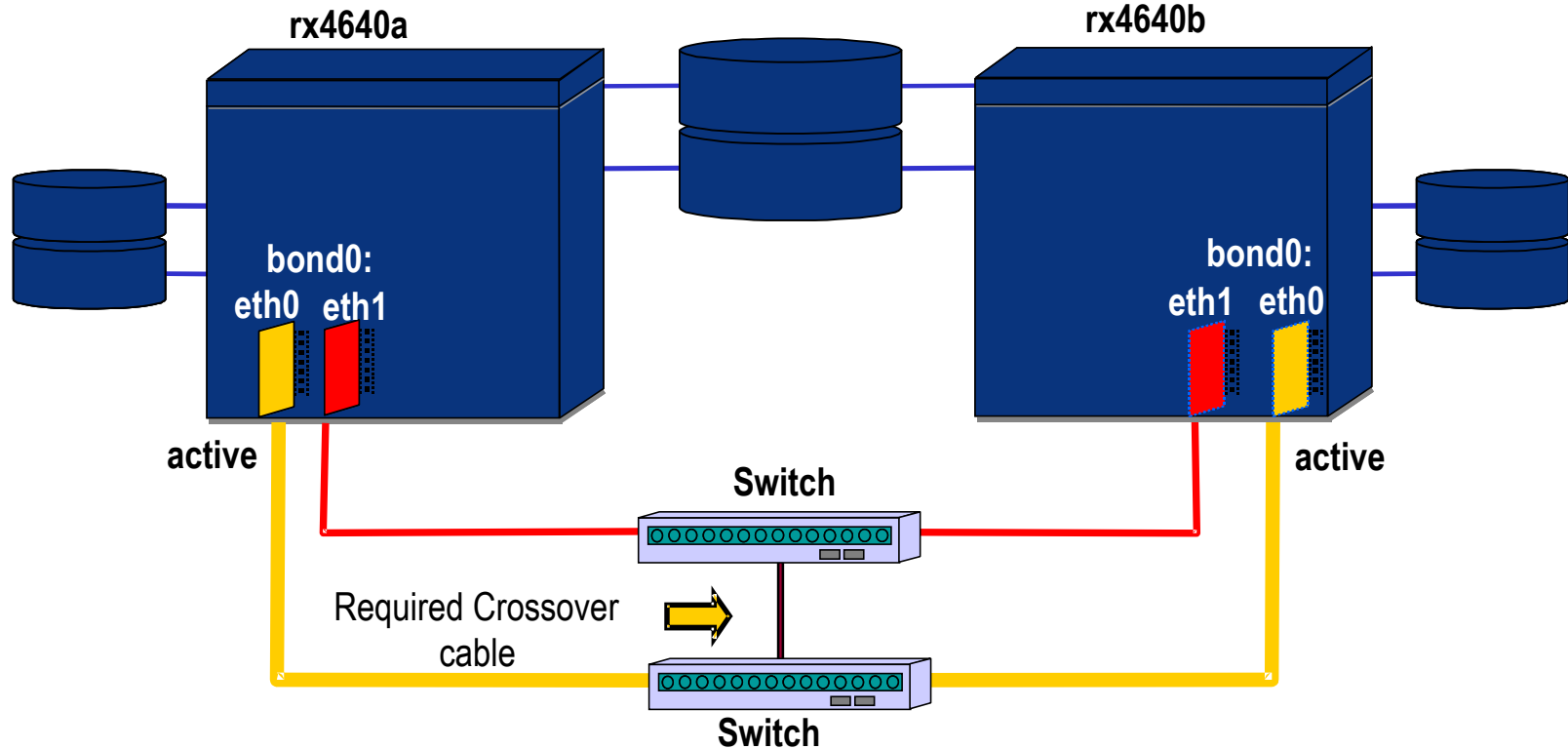
- second package: MySQL
- - # fdisk /dev/sdc
 - primary
 - partition 1
 - start cylinder 1
 - end cylinder +4000M /* 4 GB */
 - t /* change partition's system ID */
 - 8e /* Linux LVM */
 - p /* print */
 - w /* write partition data */

transfer disk partitions to other cluster nodes

- # ssh -l root rx4640b
- # fdisk -l /dev/sdb
- # fdisk -l /dev/sdc
- # sfdisk -R /dev/sdb
- # sfdisk -R /dev/sdc

- Verify partition table with the commands:
 - # fdisk -l /dev/sdb
 - # fdisk -l /dev/sdc

configure network channel bonding



- Individual variable Ethernet interfaces are slaves and the bond is master
- In this case, bond0 was the master and eth0 (active) and eth1 (inactive) were the slaves

bonding configuration (1 of 3)

Create file: /etc/sysconfig/network-scripts/ifcfg-bond0

```
# Broadcom Corporation|NetXtreme BCM5701 Gigabit Ethernet  
DEVICE=bond0  
BOOTPROTO=none  
BROADCAST=192.6.173.127  
IPADDR=192.6.173.2  
NETMASK=255.255.255.128  
NETWORK=192.6.173.0  
ONBOOT=yes  
USERCTL=no  
GATEWAY=192.6.173.1
```

bonding configuration (2 of 3)

Create file: /etc/sysconfig/network-scripts/ifcfg-eth0

```
# Broadcom Corporation|NetXtreme BCM5701 Gigabit Ethernet  
DEVICE=eth0  
BOOTPROTO=none  
ONBOOT=yes  
USERCTL=no  
MASTER=bond0  
SLAVE=yes  
PEERDNS=no  
TYPE=Ethernet
```

bonding configuration (3 of 3)

Create file: /etc/sysconfig/network-scripts/ifcfg-eth1

```
# Broadcom Corporation|NetXtreme BCM5701 Gigabit Ethernet  
DEVICE=eth1  
BOOTPROTO=none  
ONBOOT=yes  
USERCTL=no  
MASTER=bond0  
SLAVE=yes  
PEERDNS=no  
TYPE=Ethernet
```

activate bonding configuration

Add the following lines to /etc/modules.conf
alias bond0 bonding
options bond0 miimon=100 mode=1

/etc/modules.conf

```
alias eth0 tg3
alias eth1 tg3
alias bond0 bonding
options bond0 miimon=100 mode=1
alias scsi_hostadapter sym53c8xx
alias usb-controller usb-ohci
alias usb-controller1 ehci-hcd
alias md-personality-7 multipath
alias eth2 tg3
```

restart networking

From the console:

Restart networking

/sbin/service network restart

View the network configuration

/sbin/ifconfig

Ensure that routing is properly configured

bonding installation tips

1. Bonding doesn't work
 - a. Usually this is the wrong network driver (in `/etc/modules.conf`). The default driver is the TG3 driver for the installations, but you may need something different (e.g. BCM5701).
 2. Can I bond dissimilar network cards?
 - a. yes
 3. I have the systems running, everything installed, I can ping systems, ssh, etc., but the cluster won't form.
 - a. See section Editing Security Files (chapter 5) in the Users manual. Usually `cmclnodelist` (and `.rhosts`) are not correctly configured.
 4. There's confusion about bonding drivers.
 - a. Use the bonding drivers that come with RedHat3 (or later).
- Refer to the "Getting Started with HP Serviceguard A.11.15 for Linux" release notes for detailed installation steps.

building volume groups

`fdisk -l` /* list all disk partitions */

`vgscan`

`vgdisplay` /* see if anything already there */

`pvcreate -f /dev/sdb2` /* data disk for apache pkg */

`pvcreate -f /dev/sdc1` /* data disk for MySQL package */

`vgcreate /dev/vg01 /dev/sdb2`

`vgcreate /dev/vg02 /dev/sdc1`

`lvcreate -L 1G /dev/vg01`

`lvcreate -L 1G /dev/vg01`

`lvcreate -L 2G /dev/vg02`

`lvcreate -L 2G /dev/vg02`

creating file systems – primary node



- mke2fs /dev/vg01/lvol1
- mke2fs /dev/vg01/lvol2
- mke2fs /dev/vg02/lvol1
- mke2fs /dev/vg02/lvol2

- mkdir <all mount points>

- mount lvols on mount points

- df <view all mounted file systems>

creating file systems – secondary node



- mkdir <all mount points>
- Deactivate volume groups and reboot failover system
 - umount all shared file systems (primary node)
 - vgchange –a n vg01 (primary node)
 - vgchange –a n vg02 (primary node)
 - ssh –l root rx4640b
 - # reboot
- Following reboot, scan for new volume groups:
 - # vgscan
- Compare disk structure between nodes
 - ls –l /dev/vg*

verify data integrity on shared storage (1 of 2)



- Activate and mount volume groups and file systems on failover system
- rx4640b (failover node)
 - vgchange –a y vg01
 - vgchange –a y vg02
 - mount /dev/vg01/lvol1 /<directory_1>
 - mount /dev/vg02/lvol1 /<directory_2>
 - echo “This is a test” > /<directory_1>/junk
 - echo “This is a test” > /<directory_2>/junk
 - umount /dev/vg01/lvol1
 - umount /dev/vg02/lvol1
 - vgchange –a n vg01
 - vgchange –a n vg02

verify data integrity on shared storage (2 of 2)

- rx4640a (primary node)
 - vgchange –a y vg01
 - vgchange –a y vg02
 - mount /dev/vg01/lvol1 /<directory_1>
 - mount /dev/vg02/lvol1 /<directory_2>
 - cat /<directory_1>/junk
 - cat /<directory_2>/junk
 - umount /dev/vg01/lvol1
 - umount /dev/vg02/lvol1
 - vgchange –a n vg01
 - vgchange –a n vg02
 - vgcfgbackup vg01 vg02

alternate filesystem commands

To create the filesystems for the LVs

- mkreiserfs /dev/vgA/lvol1
 - *(making a journaled filesystem, reiser filesystem in /dev/vgA/lvol1)*
- mke2fs /dev/vgB/lvol1
 - *(making extend2, a non-journaled filesystem in /dev/vgB/lvol1)*

preventing boot-time vgscan

- On Red Hat, use the following procedure to prevent a vgscan command at boot time:
 - After your volume groups have been created on a node, back them up using vgcfgbackup, then comment out the following lines in the /etc/rc.d/rc.sysinit file:

```
# LVM initialization
if [ -e /proc/lvm -a -x /sbin/vgchange -a -f /etc/lvmtab ];
then
action $"Setting up LVM:" /sbin/vgscan && /sbin/vgchange -a y
fi
```

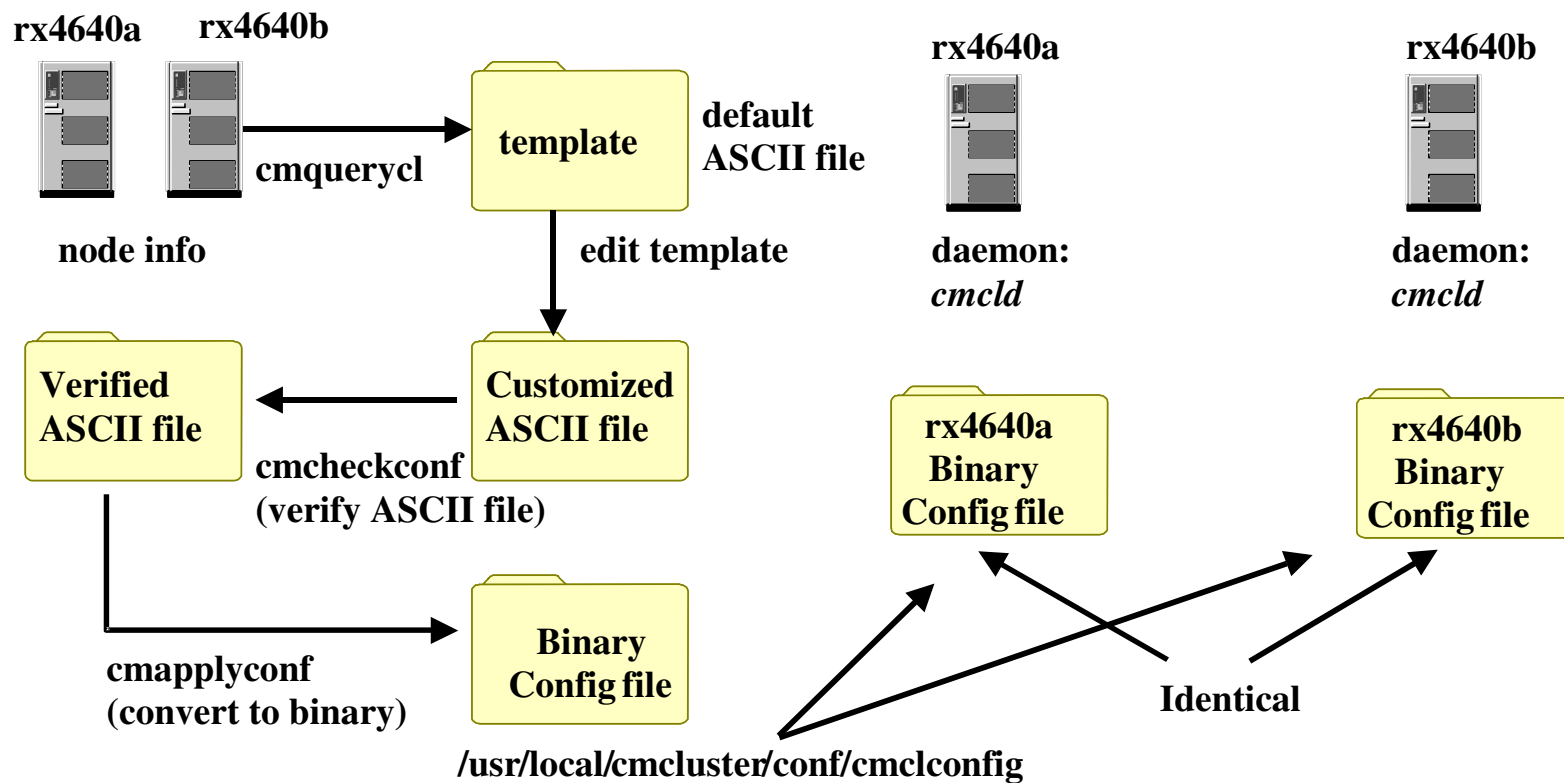
- Commenting out these lines will prevent the vgscan from removing the old volume group device files.

cluster configuration commands

- Configuration commands
 - cmquerycl
 - cmcheckconf
 - cmapplyconf
 - cmgetconf
 - cmdeleteconf
- Administration commands
 - cmrunnode
 - cmhaltnode
 - cmruncl
 - cmhaltcl
 - cmviewcl
 - cmviewconf
 - cmscancl

Same commands as SG on HP-UX!

generating a cluster configuration file (cmquerycl)



cmquerycl command

– cmquerycl command

- Specifies a set of nodes to be included in the cluster
- Generates an ASCII template for the cluster configuration file. The ASCII configuration file specifies the configuration of the cluster and its nodes
- `cmquerycl -v -L /dev/sdb1 -C /usr/local/cmcluster/cluster.ascii -n rx4640a -n rx4640b`

cluster configuration

- Cluster and package creation/manipulation commands are identical to the commands on HP-UX !
- Configure cluster as you would on HP-UX

differences from Serviceguard on UX



- Edit cluster.ascii to comment-out quorum server entry and add lock lun entry
- In the package control script, the file system type (in this case) was “ext2” and the mount options were “-o rw”
- Location of log files
 - package log files - /usr/local/cmcluster/<package_directory>
 - system log file - /var/log/messages
- Remember that rcp is not available on Linux.
 - use scp instead

Solution implementation



system monitoring

- Script written to examine the system log file (/var/log/messages) periodically for reports on all configured HA devices (disk, LAN, server).
- The presence of errors relating to a device were flagged as failover situations

application monitoring

- Application state
 - Apache
 - MySQL
- Filesystem availability and capacity

failover scenarios tested

- Node power failure
- Loss of single LAN interface per server
- Loss of all LAN interfaces on a server
- Loss of shared disk connectivity by one server
- Loss of monitored application
 - Apache
 - MySQL

Apache Web Server configuration

- IP and application failover
- No application state preserved in case of failover
- No state data transferred to failover server
- Configuration data located on shared disk.
- Provide continuous availability (or minimum disruption) for the employee who is connecting to the HA cluster via a Web browser.
- Employees must reconnect to Apache after failover
- In case of any failure (OS, network, or server), heartbeat or the app monitor will detect the outage and initiate the failover of the package to the secondary machine.

MySQL database configuration

- IP, application and data failover
- Database tables, including all committed transactions, transferred to failover server via shared disk
- Configuration data located on shared disk
- Employees will not have to reconnect to Apache, as Apache maintains the backend connection to MySQL
- Goal is to provide continuous availability (or minimum disruption) for the employee who is connecting to the HA cluster Web server via a Web browser.
- In case of any failure (OS, network, or server), heartbeat or the app monitor will detect the outage and initiate the failover of the package to the secondary machine.

Lessons learned



lessons learned (1 of 3)

- SG/LX support of RHEL AS3 made application transition a non-event
- Monitor as much as is humanly possible, to provide automatic failover
- Do not configure automatic package failback
- Disk mirroring is available for Linux, and can prevent data loss
- RAID would have been nice, but the customer placed a higher priority on processing power

lessons learned (2 of 3)

- You can't architect a failsafe solution with a limited budget (e.g. diesel generators, multi-level network redundancy, etc.)
- Architect to address those portions of your environment that stand the highest chance of failure (i.e. disk mechanism, LAN cable, network switch, network router, server power supply)
- Eliminate as many of the SPOFs as you can
- Disaster recovery would have been nice, but was not do-able, financially, at this time

lessons learned (3 of 3)

- Contributed SG/LX scripts for Apache and MySQL:
 - worked “out of the box”
 - helped immensely in the implementation of this solution
- Integration of SG/LX with RHEL AS3 is incredibly smooth
- Integration documentation was first-rate

Customer satisfaction



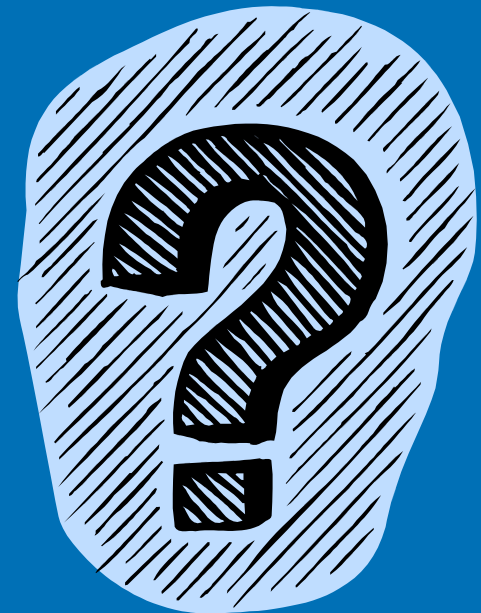
“When all the work was done, the system was up, and could recover automatically from any number of failover scenarios. SG/LX let us utilize our open source expertise to make a critical application available 24x7, handle our capacity requirements and keep our revenue flowing in.”



CIO, Pharmaceutical Development & Manufacturing Company



Questions?



HP WORLD 2004

Solutions and Technology Conference & Expo

Co-produced by:



RECOMMENDED TRAINING VENUE FOR THE
HP Certified Professional

