High Availability NFS on Linux

Winson Wang

Hewlett-Packard Company
Cupertino, CA
Email: winson_wang@hp.com
Tel: 408.447.4537

Agenda

- Network File Systems on Linux
- Clustering Concepts Used in MC/ServiceGuard-Linux
- Toolkits for High Availability Network File Systems
- Toolkit Basic/Advanced Features
- Examples and Scenarios

Network File Systems on Linux

- Existing Network File Systems on Linux
  - Coda File System
  - NFS
  - SMB (CIFS)
  - AFS (Andrew File System)
  - NCP (Novell NetWare Core Protocol)
- Two most widely used Network File Systems
  - NFS (serve Unix-based users)
  - CIFS (serve Windows users)

NFS Basics

- Available on all Unix-Based Servers
- Versions
  - NFSv3, NFSv2, NFSv4
  - Linux kernel 2.4 supports NFSv3 & NFSv2
- Features
  - Seamless access
  - Security
  - Invulnerable to system crash or reboot
  - High performance

NFS Characteristics

- Remote procedure calls
- Retransmissions of messages
- Idempotent operations – executing operation again does not change the outcome
- A stateless server
- NFS file handle to identify files
- Caching on the client
- Maintaining Unix file system semantics

NFS File Handle

- A key data passed between Server and Client
- Encode NFS file information
  - A file system identifier, an index number of a mounted local file system
  - The inode number of the file within the file system
  - An *inode generation number*
  - Other information (listed in the /usr/include/linux/nfsd/nfsfh.h)

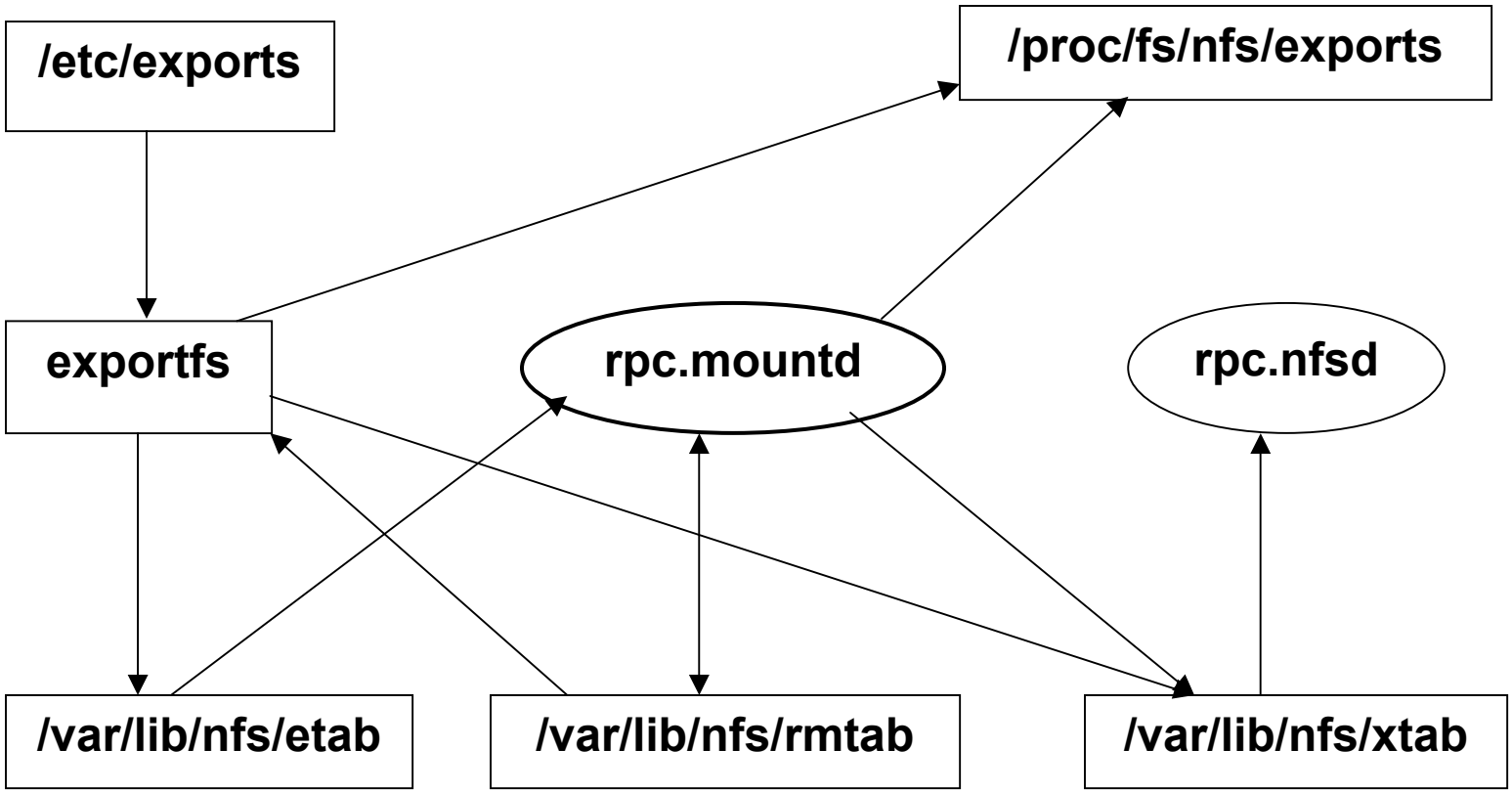
Maintaining Unix file system semantics

- Inode Generation Number
- Hidden.nfs files on client
- File and record locking
  - Need to maintain a state about the locked files
  - A separate locking daemon (lockd) for all clients

NFS States

- Utility (daemons) states are in files under /var/lib directory
- Mount and export file system states are in three files
  - **rmtab**
  - rtab
  - xtab
- File lock states are in files (file name is each client IP address)
  - statd/sm/{client-ip-address}

NFS Utilities and State Files



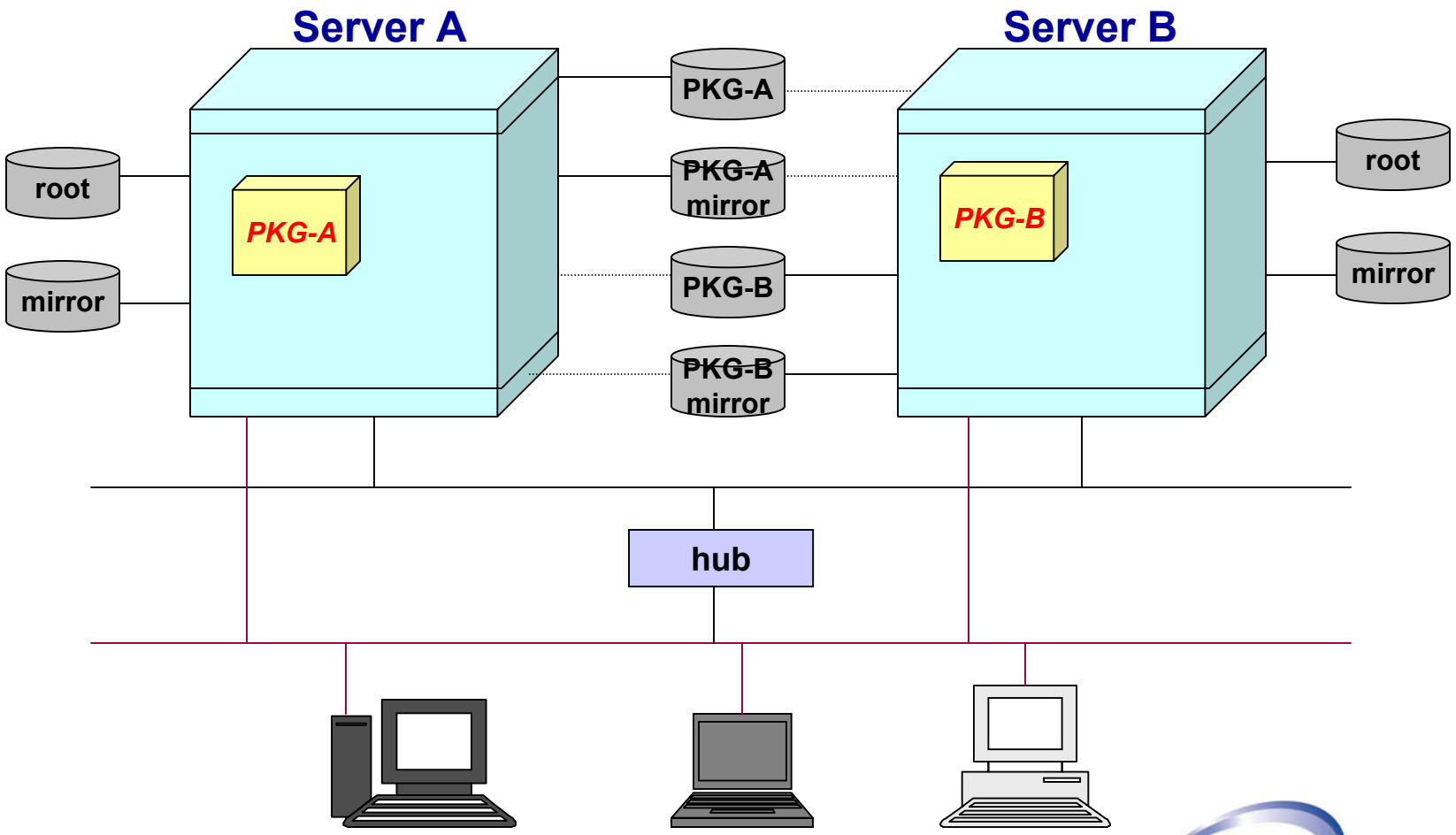
SAMBA Basics

- Provide SMB, CIFS services on Unix-based systems
  - File & print server
  - Authentication and Authorization
  - Name resolution
  - Service announcement

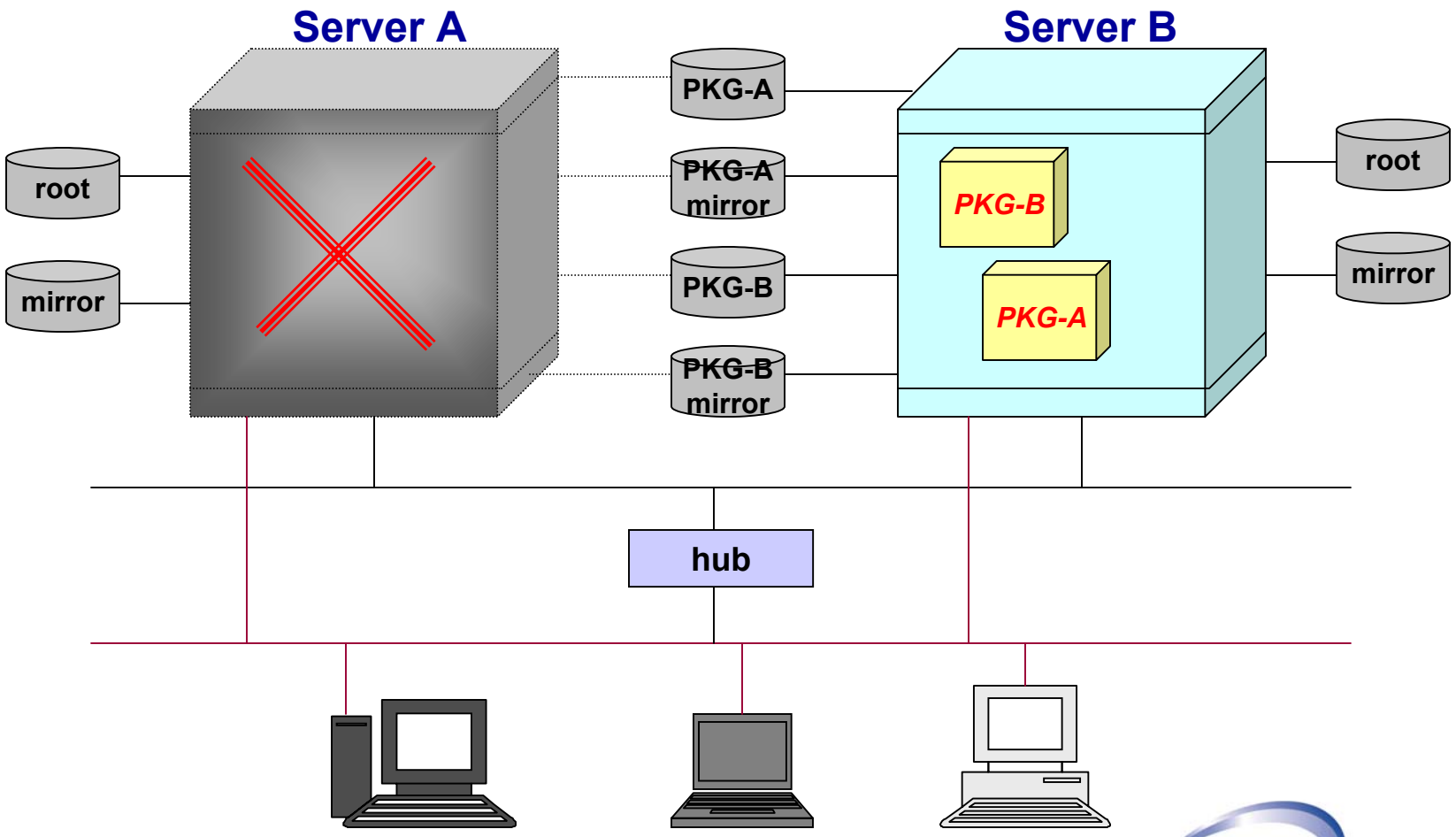
Clustering Concepts Used in MC/ServiceGuard

- Introducing MC/ServiceGuard
- High Availability with MC/ServiceGuard
- Features of MC/ServiceGuard
- Benefits of MC/ServiceGuard
- How MC/ServiceGuard Works
- MC/ServiceGuard Packages

Introducing MC/ServiceGuard



High Availability with MC/ServiceGuard



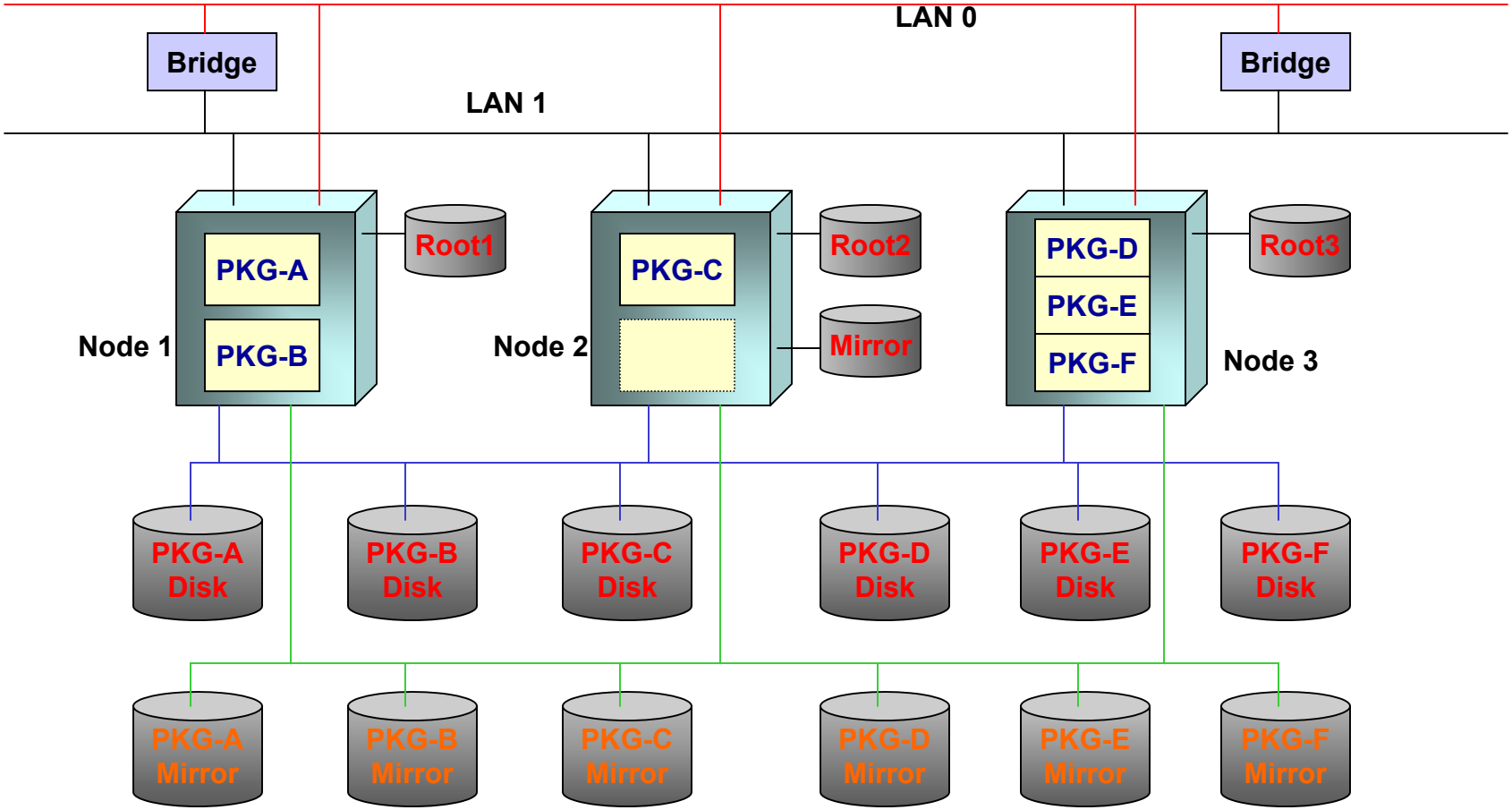
Features of MC/ServiceGuard

- Highly Available Cluster (applications recover to alternate node in < 60 seconds)
- LAN failure protection (fast local switch to standby LAN adapter using same IP address)
- Application Packages allow all resources for a package to be defined in one place
- Automatic cluster reconfiguration after a node failure
- Intelligent cluster reconfiguration after a node failure
- No idle resources
- Facilitates online hardware and software updates

Benefits of MC/ServiceGuard

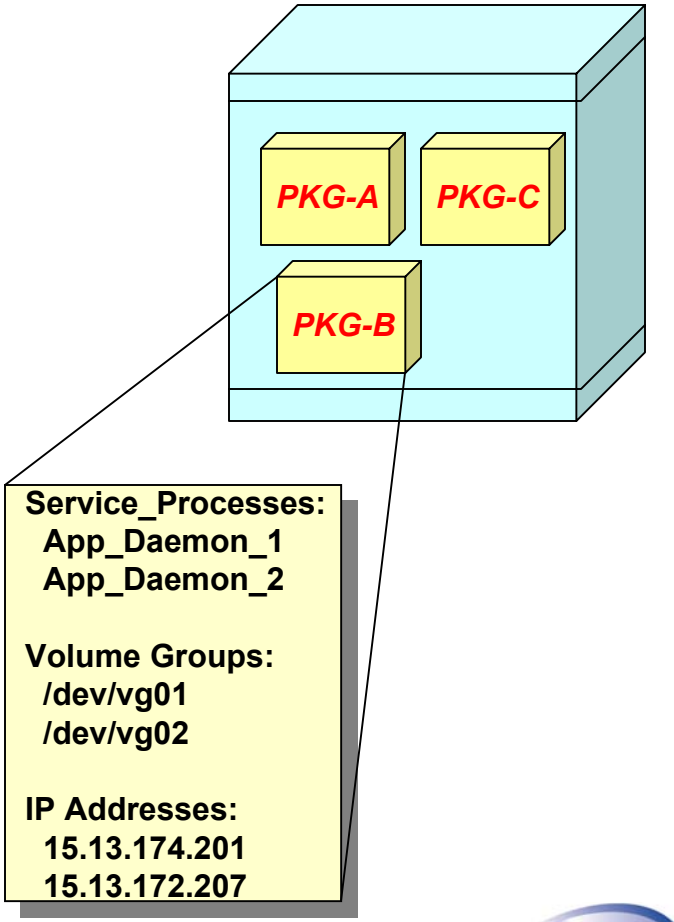
- Applications remain available to users, even after a hardware or software failure
- LAN card failures do not cause an application outage
- Applications can be moved easily and transparently without client reconfiguration
- No manual user intervention is needed to recover from a node failure
- Data integrity is preserved during a node failure
- Every node runs a production application
- Applications available during hardware and software upgrades
- Flexible load balancing during failover

How MC/ServiceGuard Works



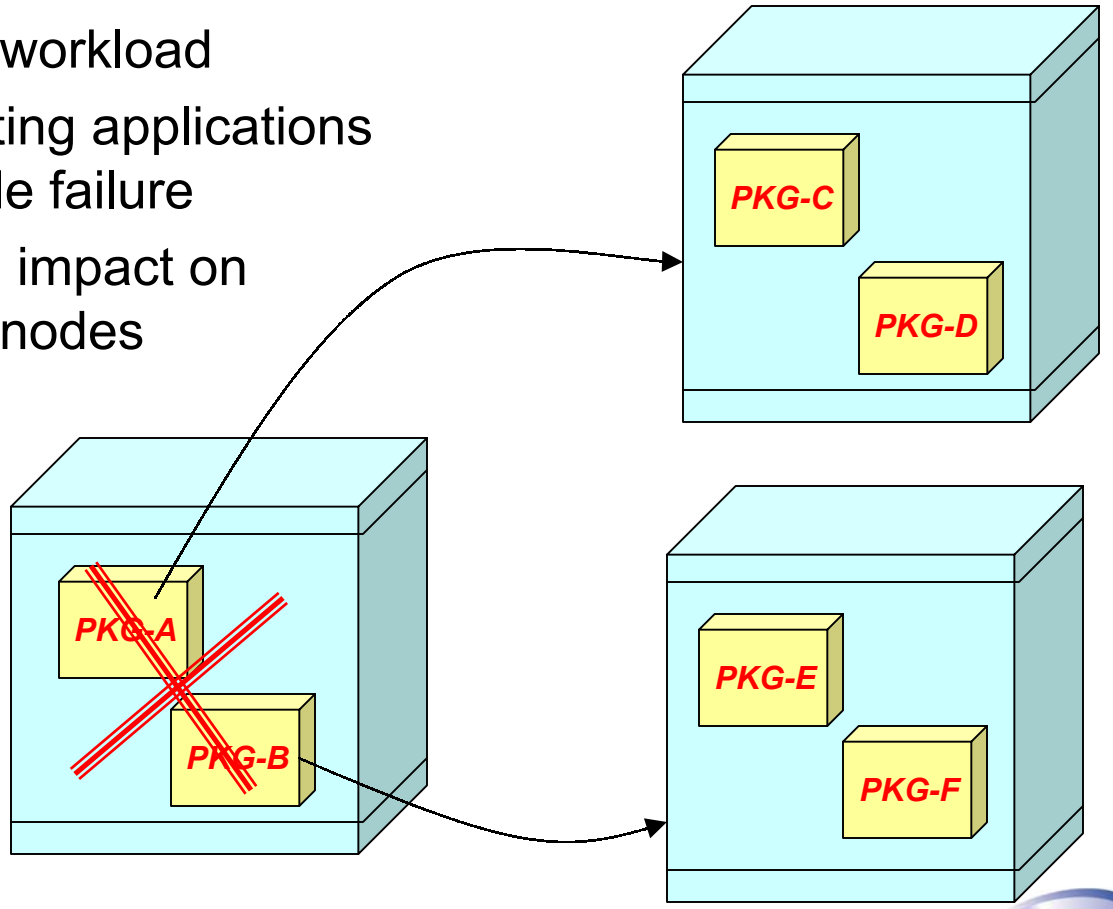
MC/ServiceGuard Packages

- Simple, flexible
- Package group all the resources for an application, including:
  - Volume groups
  - IP addresses
  - Services (processes)
- Packages are the entities that are managed and moved within a cluster environment



Distributing Application Packages

- Balancing workload
- Redistributing applications after a node failure
- Minimizing impact on remaining nodes



Toolkits for High Availability Network File Systems

- Running under MC/ServiceGuard environment
- App configuration and interface with Package Control Script (a ServiceGuard component)
- Provide HA functions and configurations
- Configuration for Workload Distribution
- Toolkit for NFS
  - Bash script (for basic functions)
  - C code (for advanced functions)
- Toolkit for SAMBA
  - Bash script (for basic functions)

Package Control Script

- Package resources configuration
  - Mirror disks
  - Volume groups and file systems
  - Package (re-locatable) IP addresses
- Resources control functions on package Start & Stop
  - Start/stop mirroring disk (MD) process
  - Activate/Deactivate disk volume groups
  - Mount/un-mount file systems
  - Enable/Disable IP addresses
  - Start/stop package service processes
  - Invoke toolkit script

NFS Toolkit

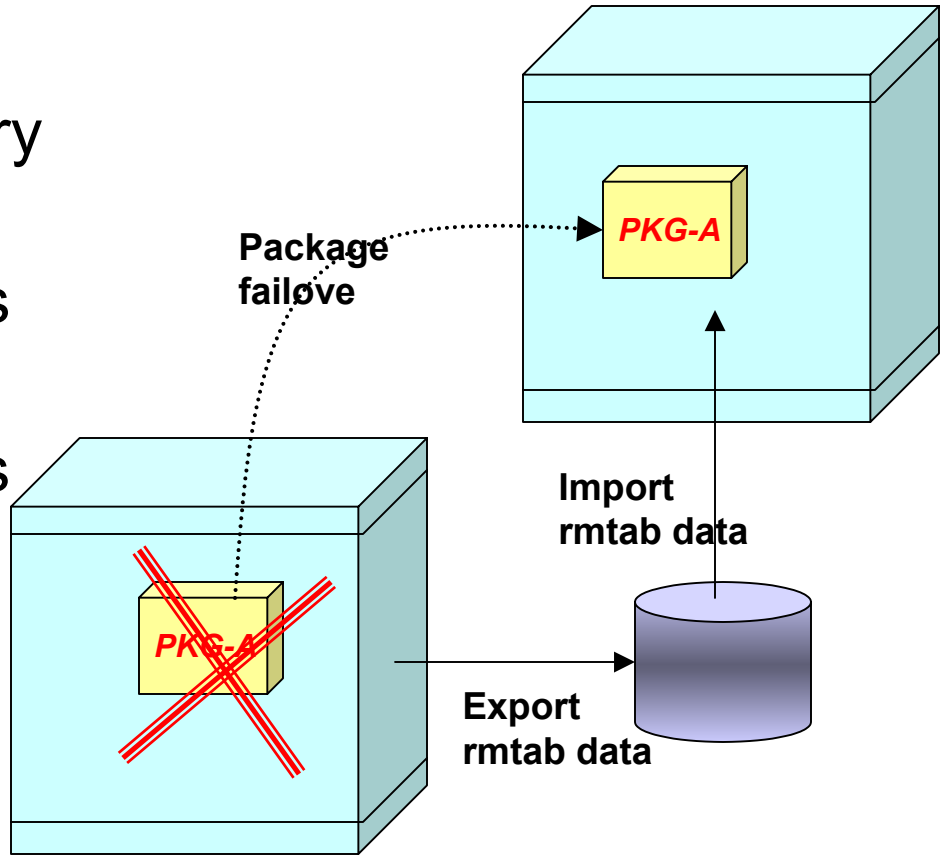
- Configurations
  - Export File Systems (Export NFS directories that will be mounted by NFS client users.)
  - NFS Monitoring Service (Monitoring script)
- Functions for NFS Start & Stop
  - Start/stop NFS server (daemons)
  - Export/Un-export file systems
  - Start/stop monitor process
  - Synchronize NFS server states
    - Current clients mount states (rmtab)

NFS daemons

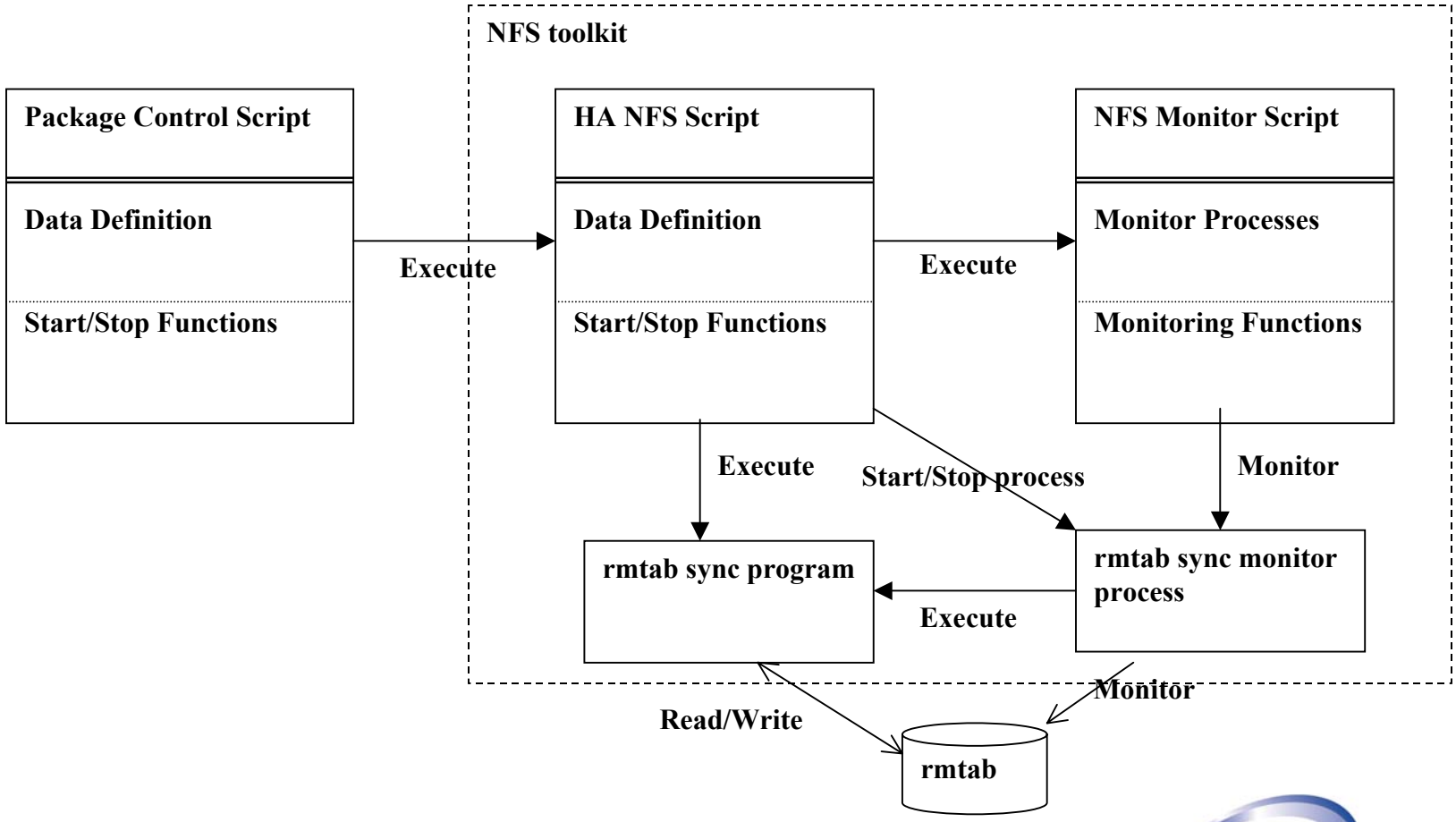
- Daemons to be started & monitored
  - portmap – protmapper
  - rpc.mountd – mount daemon
  - nfsd – nfs daemon
  - lockd – lock daemon
  - rpc.statd – state (lockd uses it)
  - rpc.rquotad

rmtab Synchronization

- Monitoring rmtab (every 5 seconds)
- Use shared storage as a media
- Toolkit imports/exports rmtab data to/from shared storage



NFS Toolkit Architecture



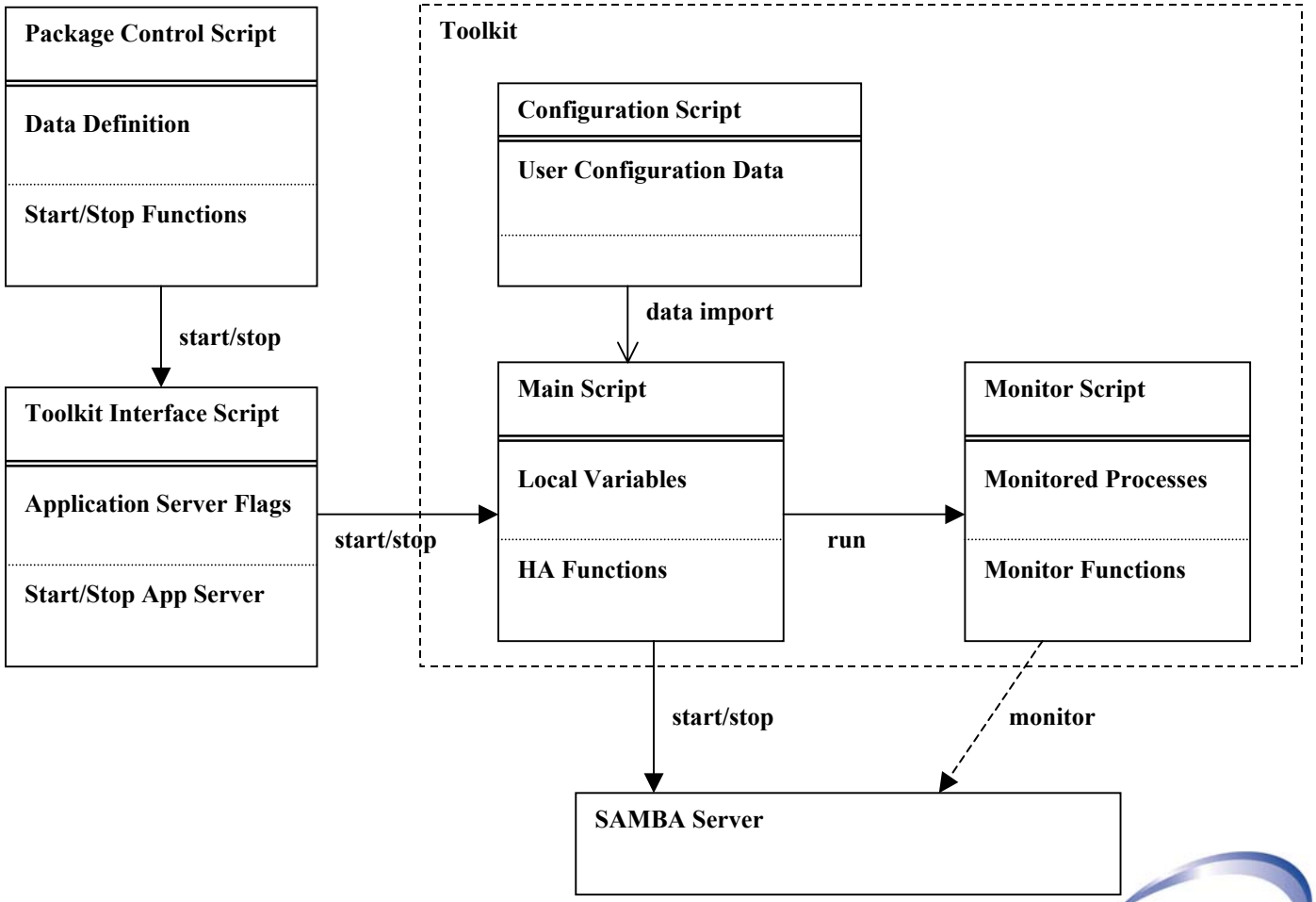
SAMBA Toolkit

- Configurations
  - SAMBA base configuration file
  - A sub-configuration file for a SAMBA package
    - For workload distribution
  - SAMBA Monitoring Service (Monitoring script)
- Functions for SAMBA Start & Stop
  - Start/stop SAMBA server (daemons)
  - Start/stop Monitor Process
  - Generate/regenerate SAMBA configuration file (for workload distribution)
  - Restart SAMBA server (for workload distribution)

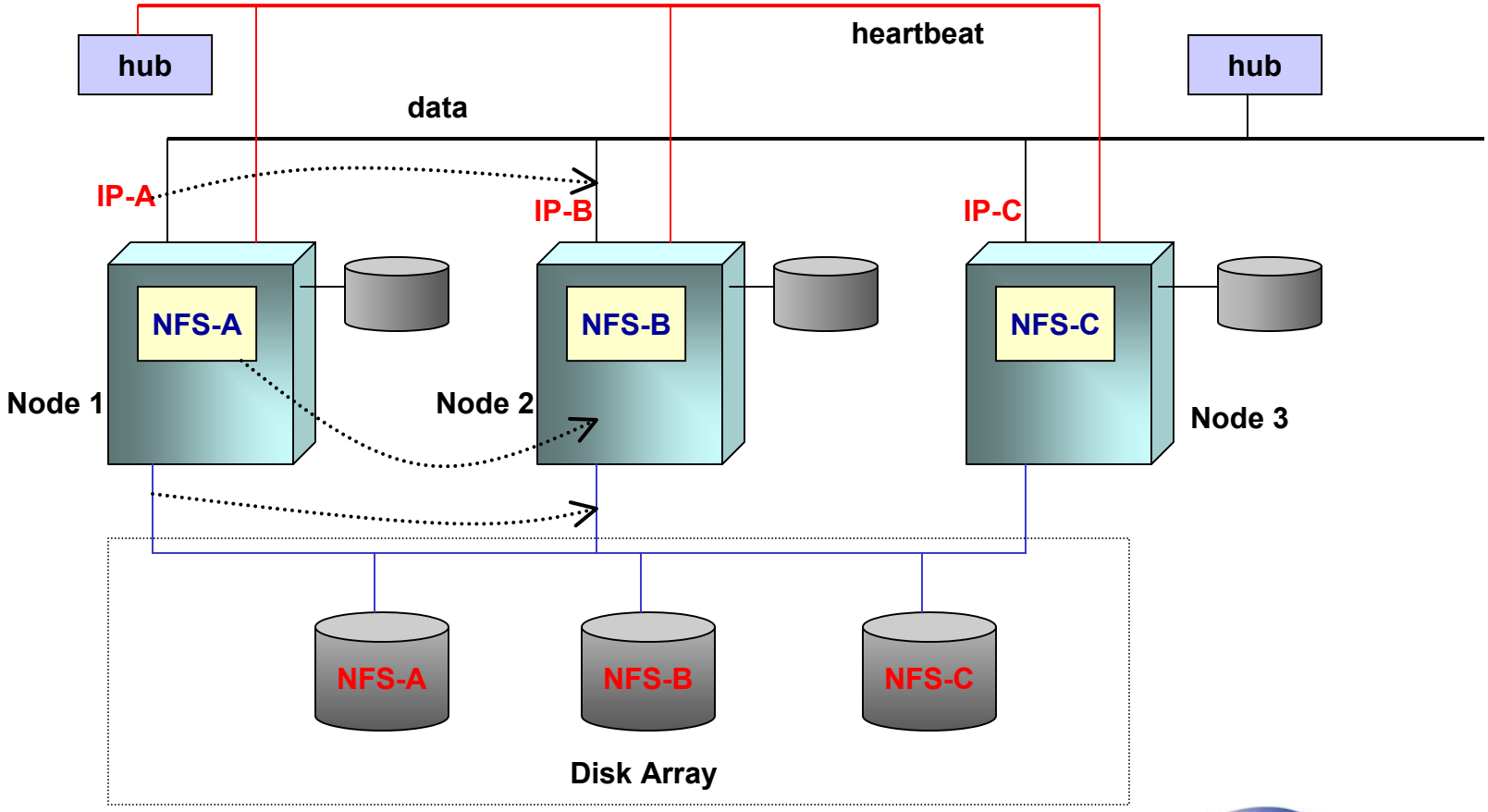
SAMBA daemons

- Daemons to be started & monitored
  - smbd
  - nmbd

SAMBA Toolkit Architecture



Sample Configuration



NFS Toolkit Limitation

- NFS failover
  - Very similar to system restart (server crashed and reboot)
  - Start on the standby node
  - Lost kernel & system state info
  - Lost daemons state info
- Client may experience a momentary hang
- Access continues when failover completes

NFS Failover

- Lost kernel & system states
  - Inconsistent file handle between Server and Client (Server fails to decode the file handle)
  - Client may get error message during file read/write.
    - Input/Output error, Stale NFS file handle
    - Write error, Stale NFS file handle
  - Workaround: give a retry. (i.e. re-open the file.)

NFS Failover (cont)

- Lost daemons states
  - Current client's mount states
  - Exported file system states
  - File lock states
- Toolkit capabilities
  - Synchronize client's mount states
  - Export file systems
  - (Does not synchronize file lock states. Client need to reclaim the file lock)

Q & A