



# Solaris Cluster Roadshow

*January 2007*

Sun Microsystems



# Agenda

- Sun Cluster Architecture and Algorithms
- Sun Cluster 3.2 New Features Deep Dive
  - > New Command Line Interface (CLI)
  - > Quorum Server
  - > Service Level Management
  - > Dual Partition Software Update (aka Quantum Leap)
  - > Solaris Containers extended support
  - > Agent development facilities
  - > Data Service Configuration Wizard
  - > Solaris Cluster for Oracle RAC
  - > HA ZFS
  - > Solaris Cluster Geographic Edition



# Solaris Cluster: Architecture & Algorithms

*Solaris Cluster Roadshow, 2007*  
Sun Microsystems



# Outline

- Introduction
- Solaris Cluster Building Blocks
  - > HA Infrastructure
  - > Resource Management Infrastructure
  - > Agent Development
  - > Manageability
  - > Disaster Recovery
- Test Infrastructure
- Availability Characterization
- Summary

# Solaris Cluster (SC)

Provides the software for Service Availability, Data Integrity, Business Continuity, and Disaster Recovery

*Availability is our customers'  
most critical requirement*

*Sun Cluster VOC Survey*

*Fifty percent of enterprises that lack a  
recovery plan go out of business within  
one year of a significant disaster*

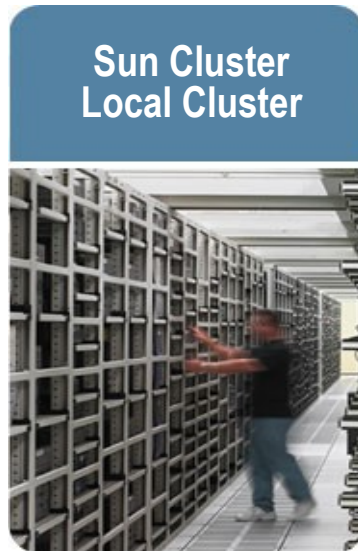
*Gartner Group*

# Solaris Cluster

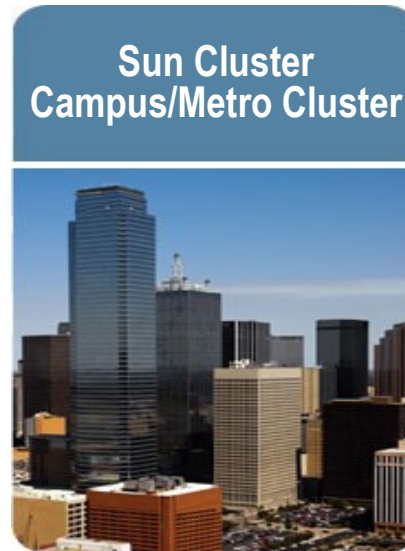
- Provides business continuity within the datacenter or across the planet
- Meets a wide range of availability needs



Single Server



Local Data Center



Hundreds of km

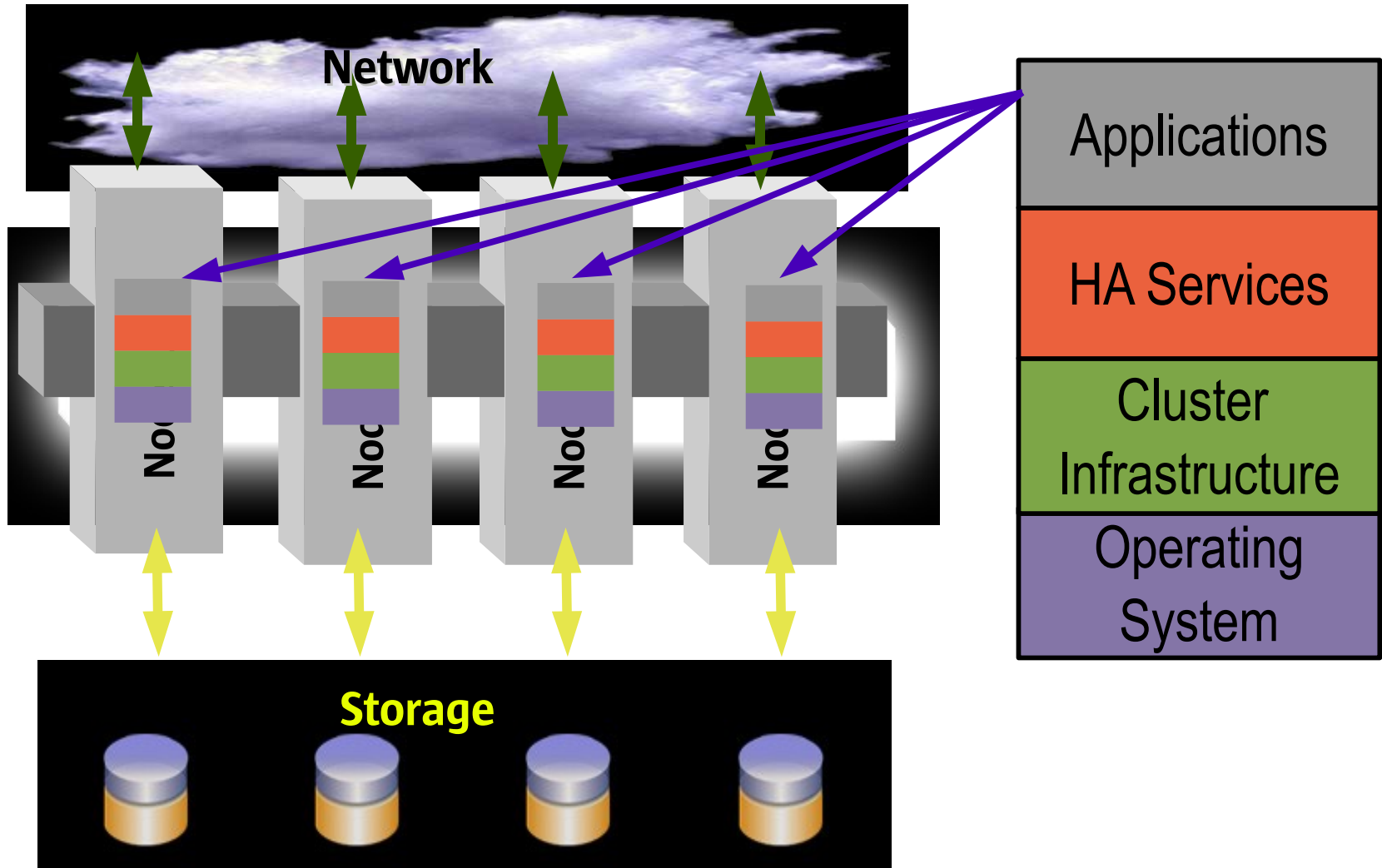


Unlimited Distance

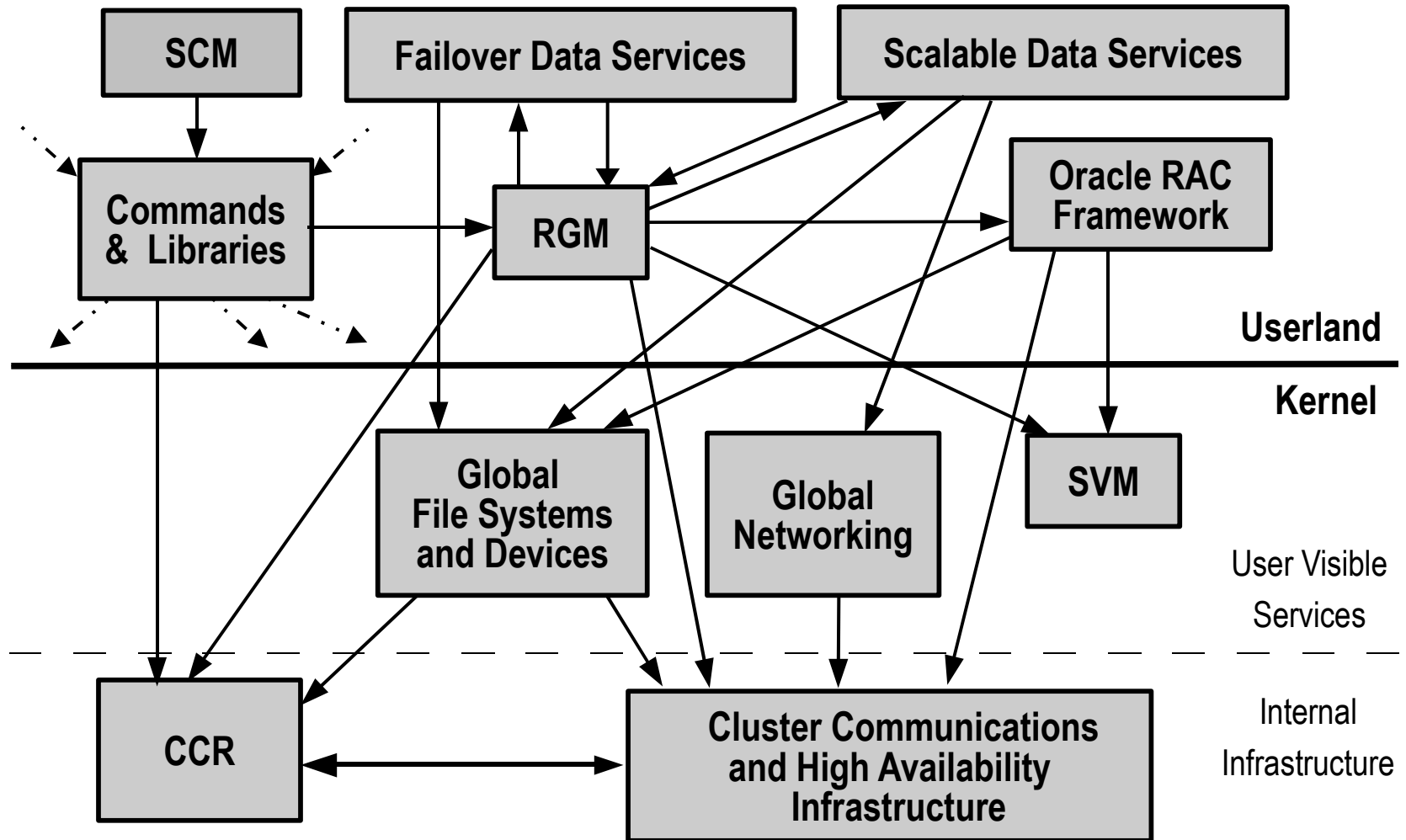
# Solaris Cluster

- Also known as Sun Cluster (SC)
- Sun's High Availability (HA) product
- Integrated with Solaris Operating System (Sparc, x64)
  - > Allows the infrastructure to be resilient to load
  - > Exploit kernel hooks to get faster reconfiguration
  - > Both these lead to higher and more predictable availability
- Supports both traditional **failover** and **scalable** HA
- History
  - > SC 3.2 FCS - Dec. 2006
  - > SC 3.0 FCS - Dec. 2000, and several updates/releases in between
  - > Prior products: HA 1.x – 1990s, SC 2.y -1990s.

# SC Stack



# SC Architecture



# SC Algorithms

- Heartbeats
  - > Monitor nodes in the cluster over the private n/w, triggering reconfiguration when nodes join/leave
  - > Resilient to load
- Membership
  - > Establishes clusterwide consistent cluster membership
  - > Coordinates reconfiguration of other layers
- Cluster Configuration Repository
  - > Global repository, local copy on each node
  - > Updates made atomically
  - > Nodes can join and leave arbitrarily

# SC Algorithms

- Quorum
  - > Prevents partitions (*split brain, amnesia*) in the cluster
    - Protects against data corruption
  - > Uses a majority voting scheme
    - 2 node clusters require a quorum device (an external tie-breaker)
- Disk Fencing
  - > Used to preserve data integrity
  - > Non cluster nodes are *fenced off* from updating any shared data

# SC Algorithms

- Membership changes trigger algorithms of upper layers, including the:
  - > ORB, Replica Framework
  - > CCR
  - > Global File System or PxFS
  - > Global Device Service
  - > Global Networking
  - > Resource Group Manager (in user space)

# SC Algorithms

- Resource Group Manager (RGM)
  - > Rich and extensible framework for plugging applications into Sun Cluster
  - > Application is wrapped by an RGM resource, supplying methods for controlling the application
    - Start, Stop, Monitor, Validate
  - > Closely related resources placed in Resource Groups
    - Ex. HA-NFS: RG has 3 resources: NFS, IP, Storage
    - An RG is a basic failover unit
  - > Supports both *failover* and *scalable* RGs

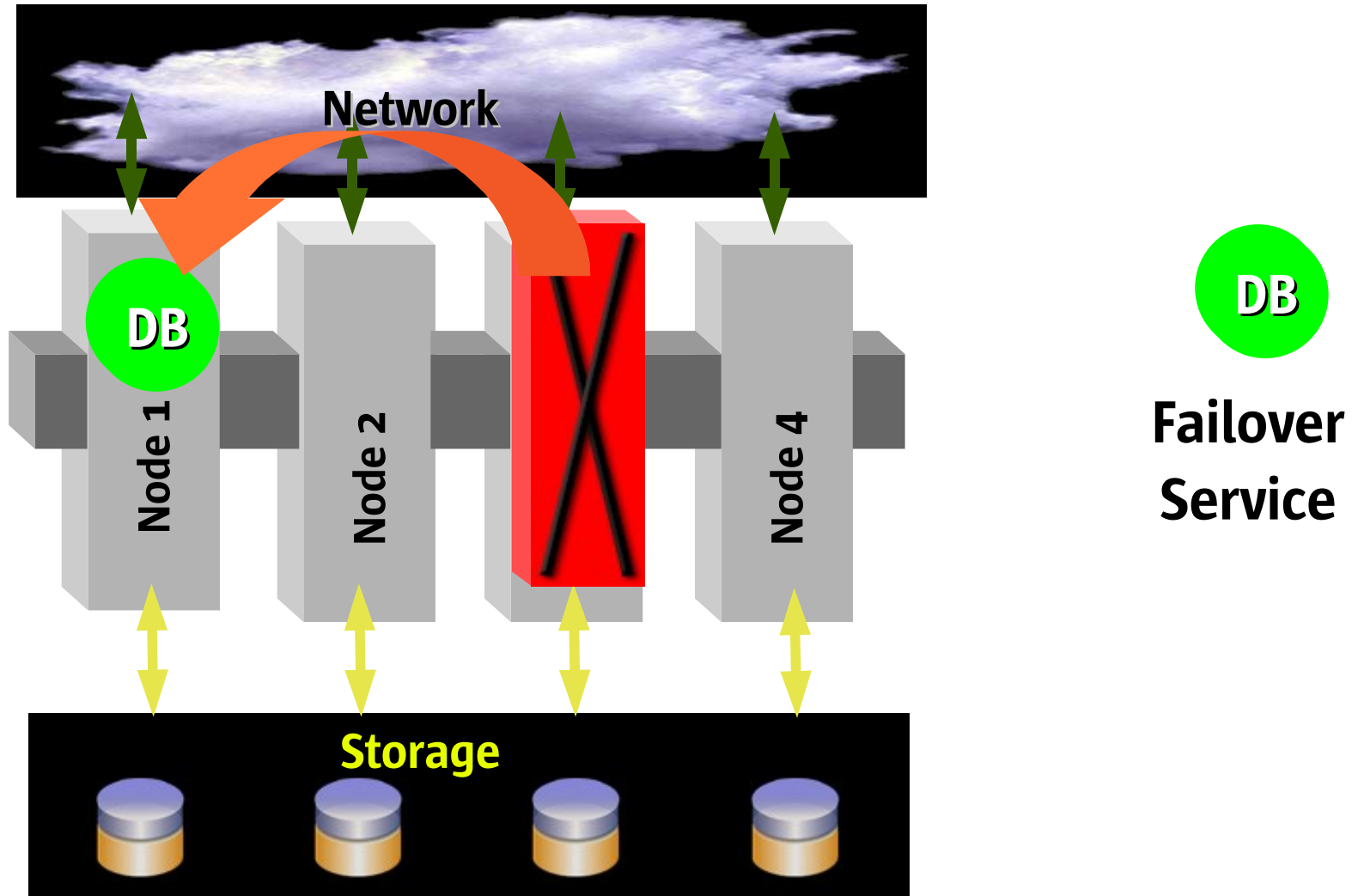
# SC Algorithms

- Resource Group Manager (RGM) - continued
  - > Support for rich dependencies between Resources and RGs
  - > Additional semantics for inter RG dependencies
  - > Solaris SMF support (SC 3.2)
    - Wrap SMF Manifest with an RGM Resource
    - Leverages SMF delegated restarter interface
    - Enables reuse of customer and ISV SMF manifests
    - After too many local (same host) restarts in a time period, recovery is “escalated” by failing over to another host
    - Configuration to control inter-host failover is attached to the RG

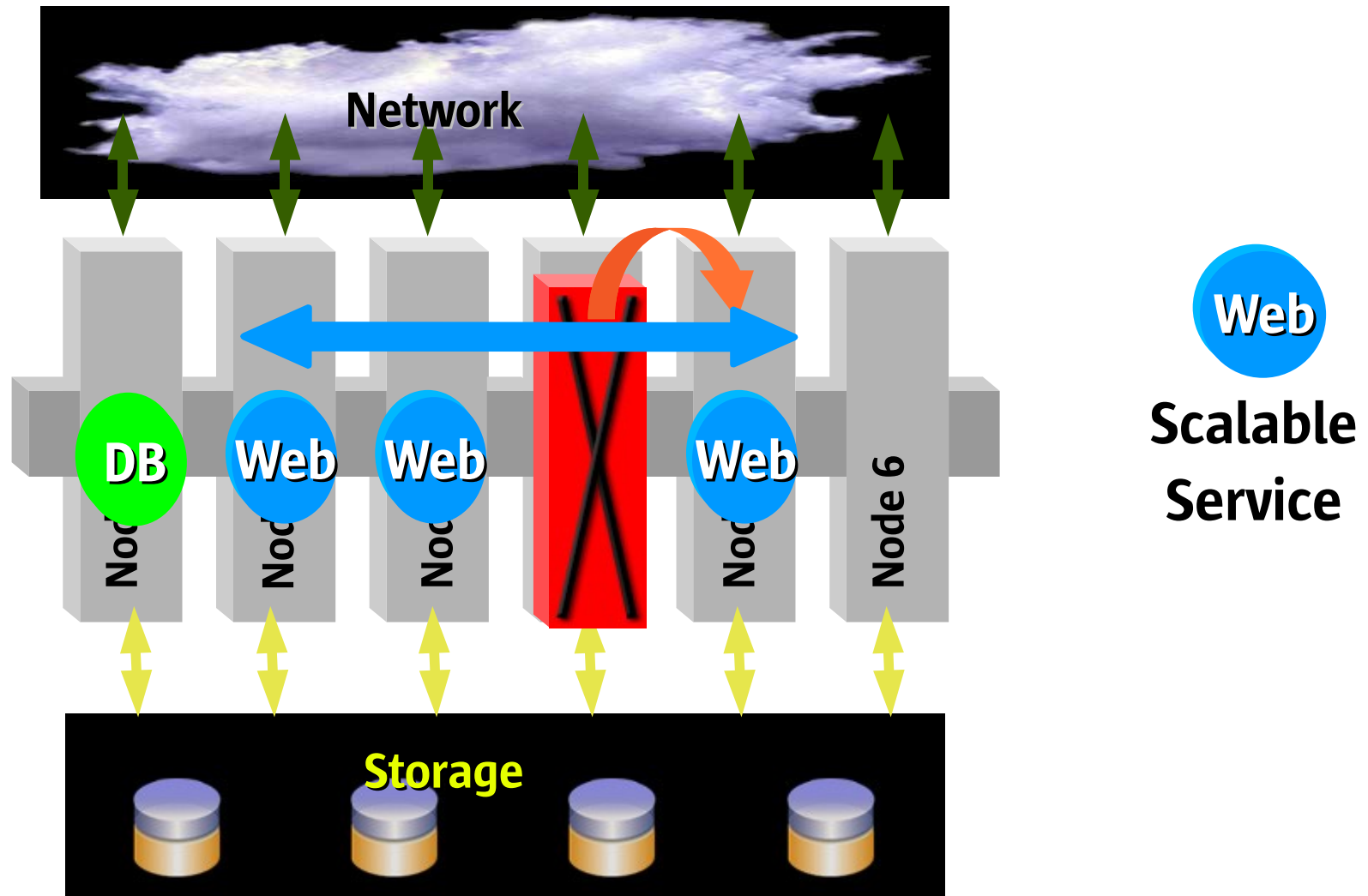
# Data Services

- Failover service
  - > Service is hosted by a *primary* node in the cluster, with backup capability on one or more *secondary* nodes.
  - > Exactly one service instance active at a time
- Scalable service
  - > Service is hosted by several nodes in the cluster at the same time, with backup capability on zero or more nodes.

# Failover Service



# Scalable Service



# Data Services Development

- Several choices available
  - > Generic Data Service (GDS)
  - > Data Service Development Library (DSDL)
  - > RGM Application Programming Interface (API)

# Large Portfolio of Supported Applications

## Web Tier / Presentation

- HA Sun Java System Web Server
- HA Sun Java System Messaging Server
- HA Sun Java System Message Queue
- HA Sun Java System Calendar Server
- HA Sun Java System Instant Messaging Server
- Scalable Sun Java System Web Server
- HA Apache Web/Proxy Server
- HA Apache Tomcat
- Scalable Apache Web/Proxy Server

## Database Tier

- Oracle Parallel Server
- HA Oracle9i and Oracle 9i RAC
- HA Oracle 10g and Oracle 10gRAC
- HA Oracle E-business Suite
- HA Oracle
- HA Sybase
- IBM DB2<sup>+</sup>
- Informix<sup>+</sup>
- HA MySQL
- HA SAP/MaxDB Database
- HA PostgreSQL

## Business Logic Tier

- HA Sun Java System App Server PE/SE
- HA Sun Java System App Server EE
- HA Sun Java System Directory Server
- HA Agfa IMPAX
- HA BEA Weblogic Server
- Scalable Broadvision One-To-One
- HA IBM Websphere MQ
- HA IBM Websphere MQ Integrator
- IBM Lotus Notes<sup>+</sup>
- HA Oracle Application Server
- HA SAP liveCache
- HA SAP J2EE Engine
- HA SAP Enqueue Server
- Scalable SAP
- HA Siebel
- HA SWIFT Alliance Access
- HA Sybase Adaptive Server

## Management Infrastructure Tier

- HA Sun N1 Grid Engine
- HA Sun N1 Service Provisioning System
- HA DNS, HA NFS
- HA DHCP
- HA Kerberos
- IBM Tivoli<sup>+</sup>
- Mainframe Rehosting (MTP)
- HA Samba
- HA Solstice Backup
- HA Solaris Container
- HA Symantec NetBackup

+ developed/supported/delivered by 3<sup>rd</sup> party

And much more through our  
Professional Services teams

# SC Manageability

- New and improved Command Line Interface (CLI) in SC 3.2
- Sun Cluster Manager (SCM) – SC GUI
- Data Services Configuration Wizards – for a set of most common data services
- Service Level Management in SC 3.2
- Upgrade options – Live Upgrade, Quantum Leap

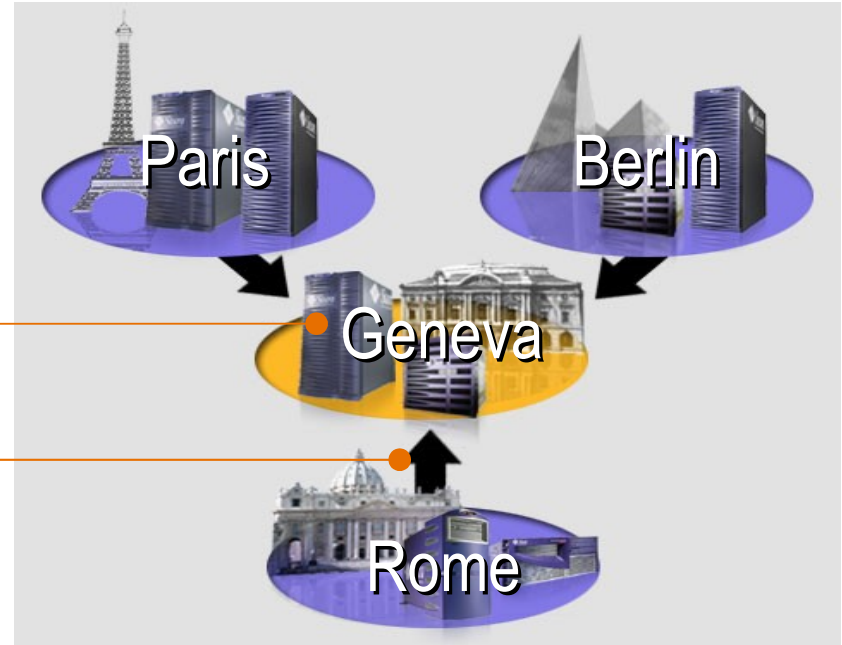
# SC Geographic Edition

## Multi-Cluster and Multi-Site capability

### N+1 Multi-site Support

One site backs up multiple cities

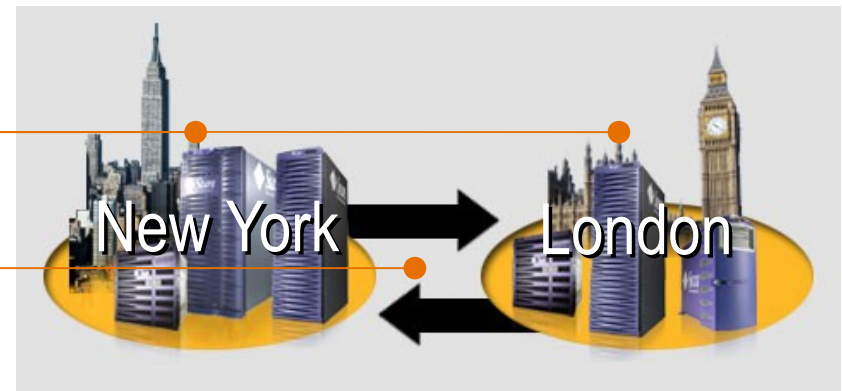
One way Data Replication



### Active – Active Configuration

Each site backs up the other

Bi-directional Data Replication



# SCATE\*

## SC Automated Test Environment

- A suite of automated tests and tools
- Distributed test development framework
  - > Fault injection (FI) framework
    - Source code based FI (white box)
    - System FI (black box)
- Distributed test execution framework
  - > Client/server architecture
  - > Easy to plug-in new test suites

\*2002 Sun Chairman Award for Innovation

# More on SCATE

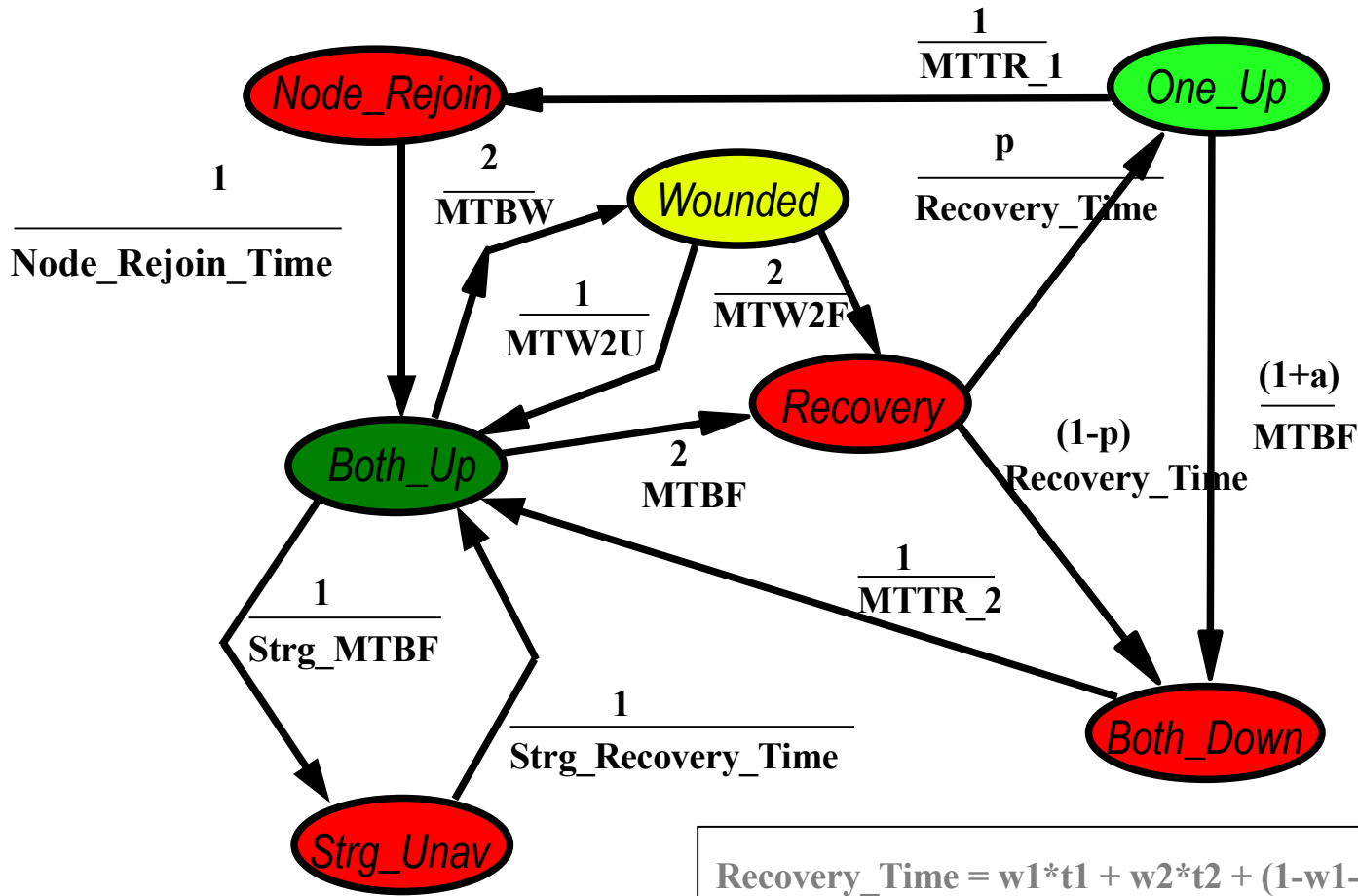
- Test Assets
  - > 50+ automated test suites, each with 100s of tests
  - > 500+ fault points in the product
- 350000+ functional tests, 45000+ faults injected
- External Qualifications
  - > Enable internal partners and 3<sup>rd</sup> party vendors to qualify their hardware (ex. Storage) and software (ex. Agents)

*SCATE had been extended/expanded to CTI (Common Test Interface), which is being used for Solaris test development.*

# Availability Characterization

- Extensive SC work on Availability Modeling, Measurement, and Improvement
  - > Code instrumentation and detailed measurements taken
  - > Leading to code improvement
- Goals: faster failover times & predictable failover times
- Availability measurements part of release testing: no regression in failover times.
- Important for meeting customer Service Level Objectives
- Application failover time and customer workload key

# 2-Node SC Availability (RAScad) Model



# SC Strengths

- Tight integration with Solaris – faster failure detection-> faster recovery -> higher availability
- Robust HA infrastructure – resilient to single points of failure, and also to many multiple points of failure
- No data corruption guarantee – many protection mechanisms in SC (membership, quorum, fencing) to enable this
- Flexibility across the stack – flexible platform for developing HA applications, and broad range of configuration choices
- Data-driven availability prediction – provides mathematical basis for offering SLAs
- Rich portfolio of supported applications
- Simple, powerful disaster recovery solution
- Sophisticated, industry-leading test framework, used both inside and outside Sun
- ...



# What's New in Solaris Cluster 3.2?

*Solaris Cluster Engineering Staff*  
Sun Microsystems



# Agenda

- This Presentation
- New Command Line Interface (CLI)
- Quorum Server
- Service Level Management
- Quantum Leap Upgrade

# This presentation

- Introduces the new features in SC 3.2
- Not much detail, just overview information
- Want more? Sign up for training



# New Command Line

# New CLI: Benefits

- Object-Oriented
- Easy-to-Remember Command Names
- Easy-to-Remember Subcommands
- Consistent use of Subcommands and Options
- Helpful Help
- Configuration Replication
- Existing CLI available
  - > All existing commands continue to work
  - > Retraining not required

# New CLI: Example 1

## Examples - Object-Oriented

### **Create a resource group**

```
node# clresourcegroup create rg1
```

Object type is *resource group*

Object is *rg1*

### **Display status of a resource group**

```
node# clresourcegroup status rg1
... <status is listed> ...
```

### **Display status of all cluster objects using umbrella command**

```
node# cluster status
... <status is listed> ...
```

Object type is *cluster*

Implied object is *this cluster*

# New CLI

## Benefit – Configuration Replication

- Ability to easily replicate configurations
  - > Most commands support *export* subcommand
    - Outputs cluster configuration to **XML**
  - > Most *create* subcommands accept *–input* option
    - Uses **XML** file as input for creating objects in operand list
    - Command line options over-ride config file content
- Possible future enhancements
  - > A single command to import entire config
  - > Apply changes to already existing objects

# New CLI: Example 2

## Example – Configuration Replication

### ***Export entire cluster configuration to XML config file***

```
node# cluster export > cluster_config.xml
```

### ***Delete all resources and groups from the cluster***

```
node# clresourcegroup delete -force +
```

The `-force (-F)` option first deletes any resources

### ***Rebuild groups and resources from the XML config file***

```
node# clresource create -a -i cluster_config.xml +
```

The `-a` option causes **clresource** to first create **rgs**

The “+” operand wildcards to mean all objects of this type



# Quorum Server

# Quorum Server Overview

- SC3.2 introduces a new type of quorum device
- Quorum Server is a Quorum Device
  - > Runs on a host external to the cluster
    - External host may be part of a cluster
    - But, it may not be part of the cluster for which it provides quorum
    - Only Solaris 9 and 10 supported
  - > Can act as a quorum device for multiple clusters
  - > Quorum Server identified by:
    - IP address
    - Port number

# Quorum Server Overview (2)

- Network Connectivity
  - > Clusters & QS may be on different subnetworks
  - > May be used in campus cluster configuration
- Interoperability - Cluster and QS Host
  - > May run different OS releases
  - > Quorum server and Cluster need not be the same architecture

# Quorum Server Installation

- Quorum Server is part of Java Enterprise System
  - > Availability Services
    - Sun Cluster Quorum Server
- Quorum Server software is distributed separately, because the software will reside on different machines
- Quorum Server must be installed and configured before cluster can configure quorum server as a quorum device



# Service Level Management

# Service Level Management in SC 3.2

- System Resource Utilization Status Monitoring/Telemetry
  - > Monitor node/zone and RG CPU/memory/swap usage
  - > Monitor disk and network adapter IO usage
  - > Threshold monitoring and trigger utilization status change event
  - > Historical data can be viewed in graph or exported to other applications.
- CPU allocation/prioritization on Cluster Resource Group
  - > Solaris Resource Management Integration on CPU resource
    - CPU share per RG
    - Dedicated CPU(Processor Set). Dynamically calculate CPU requirement for all RGs running in the same zone container, and attempt to allocate and bind CPU resource(processor) to the zone container based on the calculation if there is enough CPU resource.

# Benefit of the Utilization Status Monitoring

- Know the head room of cluster nodes. Help to consolidate data services/RGs running on the cluster nodes.
- Know how data services/RGs are being performed, and some other aspects unhealthy status in data services(such as memory leakage and etc). Help to do planned switchover instead of failover.
- Future system resource planning based on the weekly statistic data.

# GUI Interface: Nodes/RG Utilization

Sun Cluster Manager – Web Browser

File Edit View Go Bookmarks Tools Window Help

Back Forward Reload Stop  Search Print

Home Bookmarks

APPLICATIONS VERSION REFRESH LOG OUT HELP

User: root Server: pbike1

## Sun™ Cluster Manager

Java™ Sun™ Microsystems, Inc.

**bike**

- Tasks
- Nodes
  - pbike1
  - pbike2
  - pbike3
- Resource Groups
- Storage
- Private Interconnects
- IPMP Groups
- Quorum
- SNMP Modules

**Status Properties Topology Utilization**

### Cluster Utilization

This page displays utilization status of cluster components. For information about system resource monitoring, see [System Resource Monitoring](#).

Cluster name: bike

#### Nodes (3)

Graph...

<input checked="" type="checkbox"/>	<input type="checkbox"/>	Node	Total CPU (CPUs)	CPU Used (CPUs)	5 Minute Load Average	Total Memory (MB)	Memory Used (MB)	Total Swap (MB)	Swap Used (MB)
<input type="checkbox"/>		pbike1	2	0.19	0.36	2048	1394.25	5570	1192.0
<input type="checkbox"/>		pbike2	2	1.11	0.95	2048	2017.25	5477	2628.99
<input type="checkbox"/>		pbike3	2	0.22	0.27	2048	1136.65	5529	899.47

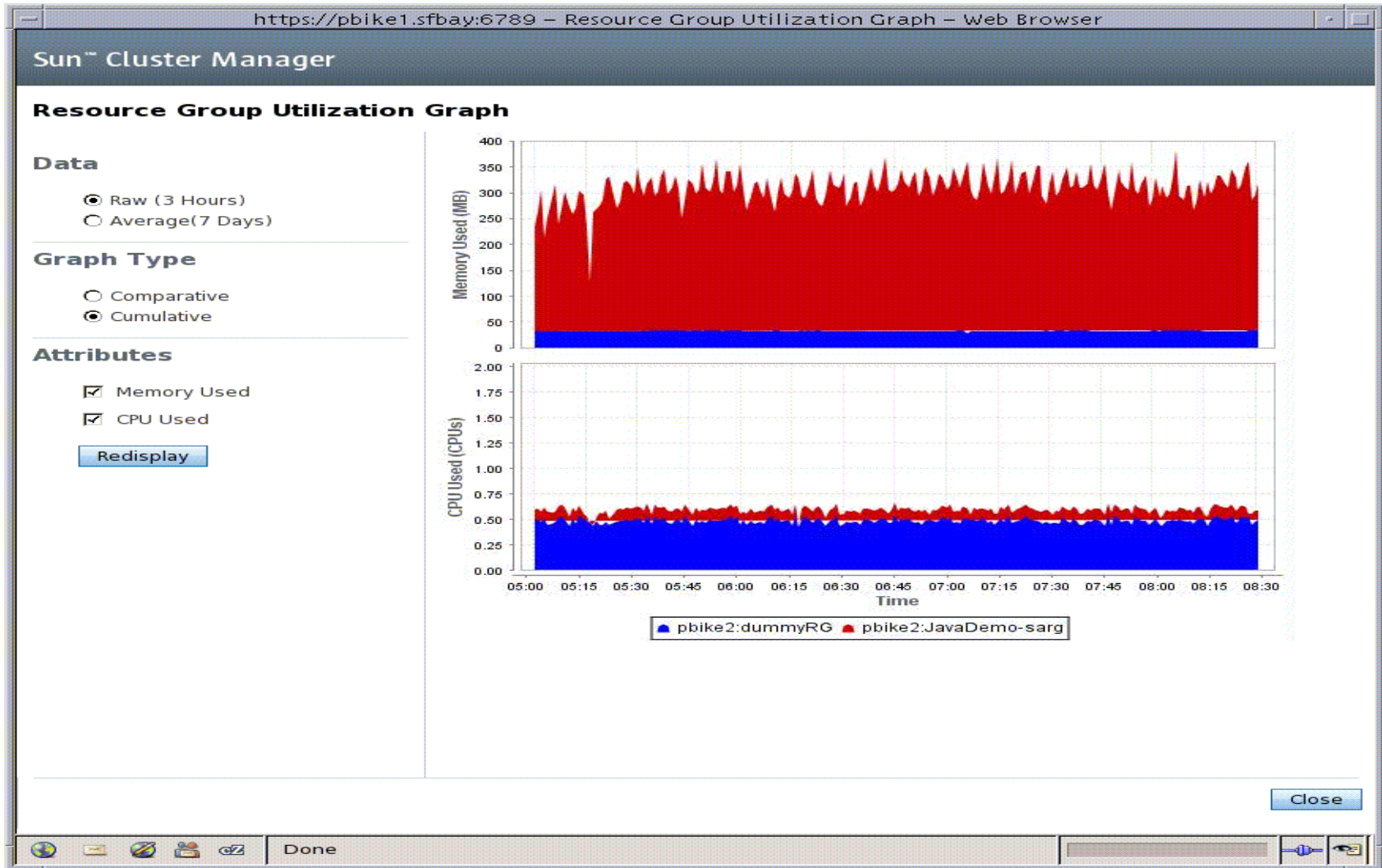
#### Online Resource Groups (14)

Graph...

<input checked="" type="checkbox"/>	<input type="checkbox"/>	Resource Group	Node	Memory Used (MB)	CPU Used (CPUs)
<input type="checkbox"/>		bike-2-rg	pbike2		
<input type="checkbox"/>		bike-3-rg	pbike2		
<input type="checkbox"/>		cl-db-rg	pbike1	201.93	0.04
<input type="checkbox"/>		cl-tlmtry-rg	pbike2	2.07	0.0
<input type="checkbox"/>		cl-tlmtry-rg	pbike1	3.51	0.0
<input type="checkbox"/>		cl-tlmtry-rg	pbike3	3.68	0.0
<input type="checkbox"/>		dummyRG	pbike2	32.02	0.48
<input type="checkbox"/>		ffzone-rg	pbike2	4.88	0.0
<input type="checkbox"/>		JavaDemo-sarg	pbike2	234.22	0.15
<input type="checkbox"/>		JavaDemo-sarg	pbike1	125.66	0.0
<input type="checkbox"/>		JavaDemo-sarg	pbike3	122.57	0.08
<input type="checkbox"/>		pgs-rg	pbike1	5.15	0.0
<input type="checkbox"/>		pgsz32-rg	pbike2:zone32		
<input type="checkbox"/>		sharedaddress-rg	pbike3		

Done

# Utilization Graph for Two RGs (stack view)



# Agenda

- This Presentation
- New Command Line Interface (CLI)
- Quorum Server
- Service Level Management
- **Quantum Leap Upgrade**

# What is Quantum Leap?

- Quantum Leap (QL) is a fast cluster upgrade technology
- Divide cluster into two partitions
  - > Exactly two, no more, no fewer
- Upgrade one partition at a time
- Quickly move applications from old version partition to new version partition
  - > Use new partition for production while upgrading old
- Marketing name is “Dual Partition Software Swap”
  - > Could be used to downgrade as well, but that is untested

# Advantages of Quantum Leap

- Quantum Leap provides the means to upgrade the entire software stack on a cluster with only a small outage.
  - > OS, cluster, applications, etc., may be upgraded
  - > Outage similar to application switchover
- Rolling Upgrade can only upgrade cluster software .
- Quantum Leap dramatically reduces the cost of engineering development and testing.

# What things are supported by QL?

- Upgrade from SC 3.1 FCS & all updates to SC 3.2
- Upgrade from S8/S9/S10 to S9u7/S10u3 and ahead
- Installing future patches & updates
- Upgrading other software:
  - > applications – Oracle, SAP, etc
  - > volume managers – SVM, VxVM
  - > file systems – VxFS
- Can use QL for upgrades with or without SC changes



# Solaris 10 Containers and Solaris Cluster

# Solaris Cluster support for S10 containers

- Available since Sun Cluster 3.1 8/05: HA Container Agent
  - > Zone is a resource; zone can fail over between nodes
  - > All RGs configured in global zone
  - > RG contains zone resource and application resource
- Significantly Enhanced in Sun Cluster 3.2: “Zone Nodes”
  - > Zones are Virtual Nodes
  - > Multiple RGs can fail over independently between zones
- Coexistence and combination of both approaches in Sun Cluster 3.2

# Why use S10 containers with Solaris Cluster?

- Combine benefits of clustering and containers
  - > Solaris Cluster provides high availability and load balancing
  - > Containers provide application isolation, fault containment, and control of system resource allocation
- Each application can run in its own zone
- Upon failure, application and/or zone can fail over to another node

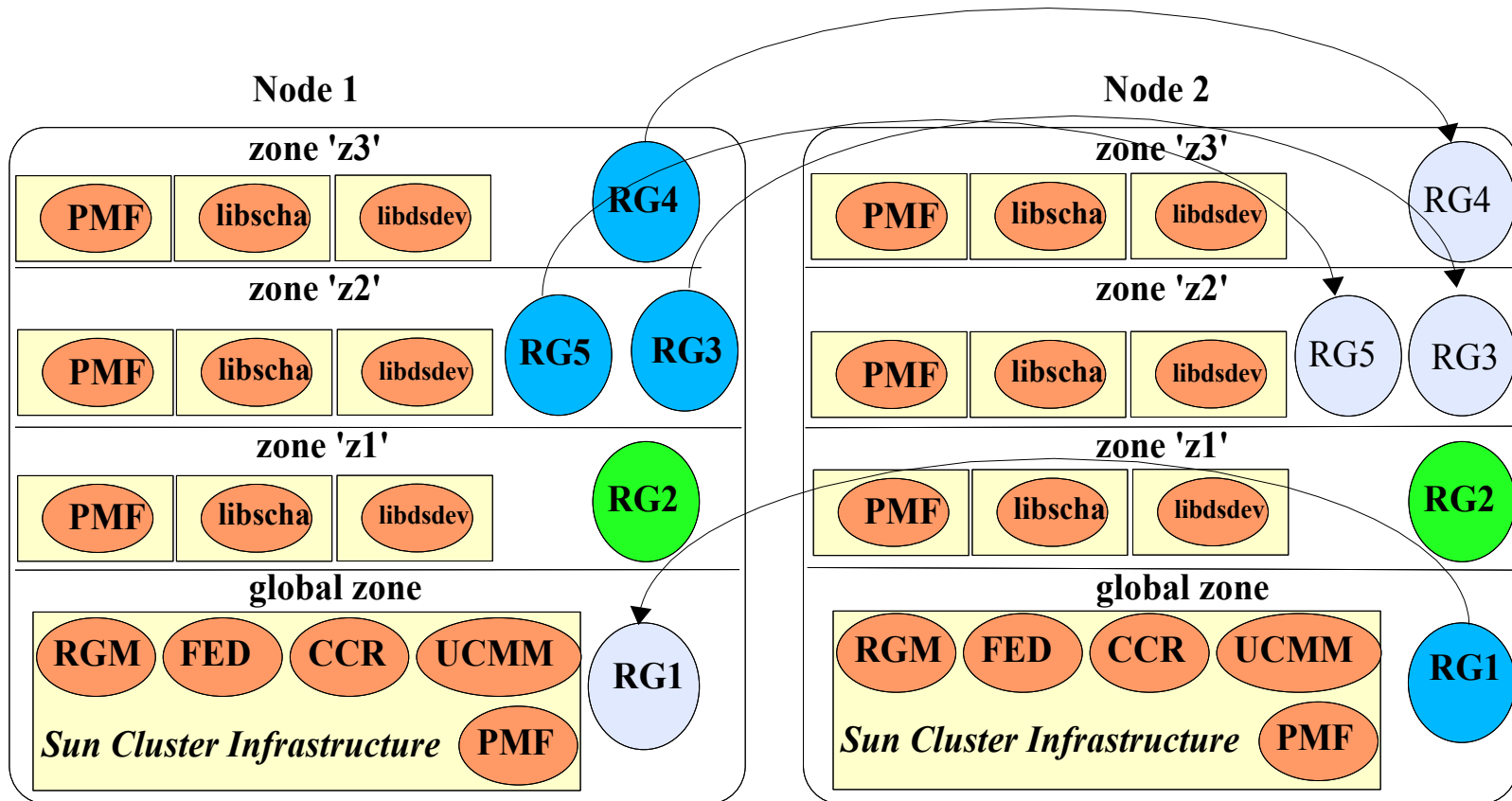
# Zone-Nodes Provide

- Application isolation
- Ability to exploit Sun Cluster agent to monitor application running within zone
- Ability to run most SC resource types (application and agent) unmodified in a non-global zone
- Ability to run multiple resource groups in the same zone that fail over independently
- Ability to dynamically create/destroy zones
  - > using the usual Solaris tools
  - > Automatic discovery by RGM

## Zone-Nodes Provide (cont.)

- Support for unbounded number of zones
- Support for resource groups to fail over between zones on the same node
  - > Does not really provide high availability
  - > Supports prototyping of cluster services
- Support for data services developed using Generic Data Service (GDS), Agent Builder, or Data Service Development Library (DSDL)

# Sun Cluster components in zones



# Zone isolation/security

- Zone isolation is incomplete
- User running in a non-global zone can “see” resource groups configured in other zones
- User running in a non-global zone cannot modify or affect behavior of RGs in other zones unless those RGs list the non-global zone in their Nodelist property
- Some admin commands are not permitted to run in non-global zone: RT register, RG create, ...
- Cluster administration is most easily done from the global zone
- Security to be enhanced by the “Clusterized Zones” (RAC in zones) project in a future release

# How to Use Zone-Nodes

- "Logical Nodename" **nodename : zonenumber**  
or **nodename**
- (old) **Nodelist=node1 , node2 , node3**
- (new) **Nodelist=**  
**node1 : zoneA , node2 : zoneA , node3 : zoneA**
- Also permitted:
  - > RG runs in different zone name per node:
    - **Nodelist=node1:zoneA,node2:zoneB,node3:zoneC**
  - > RG runs in multiple zones on single physical node:
    - **Nodelist=node1:zoneA,node1:zoneB,node1:zoneC**

# Zones support in Sun Cluster Manager

The screenshot shows the Sun Cluster Manager web interface in a browser window. The address bar shows the URL: `https://procks2:6789/SunClusterManager/index/Index`. The page title is "Sun Cluster Manager".

The left sidebar shows a tree view of the cluster configuration. The "Zones" folder under "procks1" is selected.

The main content area shows the configuration for the zone "procks1:zrocks1a". The "Status" tab is active, displaying the following information:

- Zone:** procks1:zrocks1a
- Status:** running

Below this, there is a section for "Resource Groups (2)". It includes a table with the following data:

Name	Auto-Recovery	Status	Type	Current Primaries
apache-server-rg		Online	Failover	procks1:zrocks1a
rocks-4-rg		Online	Failover	procks2:zrocks2a

# Data Services supported in non-global zones

- Combined with the HA Container Agent:
 

Apache Tomcat	MySQL
Samba	IBM WebSphere MQ
PostgreSQL	N1 Grid Service Provisioning System
  
- Using zone nodes with Sun Cluster 3.2
  - > all agents that are supported with the HA Container Agent, plus:
 

JES Application Server	JES Web Server
JES MQ Server	DNS
Apache	Kerberos
HA Oracle	Oracle E-Business Suite
Oracle 9iAS	GDS
IBM Websphere MQ Broker (pending IBM support confirmation)	
  
- Refer to the Config. Guide for the latest agent support info.

# Competition

- Veritas Cluster (VCS) offers a Container Agent
  - > VCS has HA Oracle and SAP container aware
- SC3.1 8/05 has similar functionality to VCS Container Agent
  - > Some of our agents are container aware in 3.1 8/05
- SC3.2 supersedes VCS with Zone Nodes and the HA Container Agent
  - > Starting with 3.2 many of our standard agents are Container aware (order of magnitude more than VCS)
  - > All our GDS custom agents can be Container aware
  - > Application failover between zones can be tested in a single node cluster for development purposes



# Developing Application Agents on Solaris Cluster

# Introduction

- Solaris Cluster 3.2 has an extensive portfolio of supported applications
  - > Agents available on the Solaris Cluster DVD or download
  - > Most JES applications ship SC Agents
- APIs and tools available for custom Agents
- Talk outline
  - > Application Characteristics
  - > SC Resource Management Model
  - > Available APIs
  - > Solaris Cluster Agent Builder Tool
  - > Hands on exercise developing a custom agent

# Application Characteristics

- Crash tolerance
  - > Be able to restart correctly after an unclean shutdown
  - > Sometime requires cleanup of lock/pid files
- Independence from server hostname
  - > *THAT* changes with a failover!
  - > New feature in SC32 to override application hostname resolution
    - export LD\_PRELOAD=/usr/cluster/lib/libschost.so.1
    - export SC\_LHOSTNAME="myHAhostname"
    - man libschost.so.1(1) for details
  - > Should be able to co-exist with multi-homed hosts
- Multi-hosted data
  - > Application should not hard code data paths
  - > Sometimes symbolic links can be used as work-around

# Resource Management Model

- Key concepts
  - > Resource Type (RT): Is a representation of an HA entity
    - Example: HA-Oracle RT, a HA Filesystem
  - > Resource: Is a specific instance of a RT
    - Example: Oracle HR database
  - > Resource Group(RG): Is a collection of resources
    - Example: A RG containing
      - A failover filesystem resource
      - A failover IP Address (aka LogicalHostname resource)
      - A failover Oracle database instance
  - > Resources can have *dependencies* between them
    - Facilitates proper startup/shutdown sequencing
    - Dependencies can have various *flavors* such as *strong/weak/restart*
    - Works **across** different cluster nodes
- Implemented by Resource Group Manager (RGM)

# Example of a Failover RG

Name: Oracle-rg

Maximum primaries: 1

Type: LogicalHostname

Hostname: ora-1

Type: HAStoragePlus

Name: hafs1

FilesystemMountPoints: /global/ora-1

Type: SUNW.Oracle\_server

Oracle\_home: /global/ora-1/oracle/

Oracle\_SID: D001

Resource\_dependencies: hafs1

# Developing Resource Types

## (aka HA Agents)

- Manages an applications such as Oracle, Apache, NFS etc.
- Implements *callback methods*
  - > To Start, Stop and Monitor the application
- Manages specific properties needed to manage the applications
  - > eg, value of ORACLE\_HOME, timeout value for a specific task
  - > Optionally implement methods to VALIDATE and UPDATE these properties
- Supplies a Resource Type Registration (RTR) file to specify the above information

# Available APIs

- Sun Cluster High Availability API (SCHA API)
  - Querying properties: *scha\_resource\_get(1HA)*
  - Taking action on failures: *scha\_control(1HA)*
  - Managing status: *scha\_resource\_setstatus(1HA)*
  - Available in C and CLI form
- PMF (Process Monitoring Facility)
  - To manage application processes
  - Quick restart after failures
  - Guaranteed stop
  - CLI interface *pmfadm(1M)*
- To run arbitrary commands under a timeout
  - *hatimerun(1M)*

# Available APIs (contd...)

- **Data Services Development Library (DSDL)**
  - Brings together SCHA API, PMF and hatimerun
  - Provides an integrated fault monitor model
    - Local restarts after application failures, repeated failures lead to inter node failover
    - Provides APIs for application fault monitoring
      - `scds_fm_net_connect(3HA)`
      - `scds_simple_probe(3HA)`
      - `scds_timerun(3HA)`
  - Available only in C
- **Generic Data Service (GDS)**
  - Layered on top of DSDL
  - Allows developers to plug in simple scripts and create RTs
  - Customers love it because there is very little code to own and maintain


# Solaris Cluster Agent Builder

- GUI based code generation tool for RTs
- You only specify how to start your app
  - > Uses Sun Cluster Process Monitoring Facility (PMF) to monitor the application processes
  - > Reliable application shutdown
  - > Tunable local restart and failover decisions
  - > Can perform simple TCP handshake level monitoring
  - > Optionally specify a monitor script for detailed application health checks
- Can generate Agent in ksh, C and GDS
- Creates a Solaris package for the Agent

# Sun Cluster Agent Builder

## Create Screen

File Edit



**Vendor Name:**

**Application Name:**

**RT Version:**

**Working Directory:**

Scalable
  Failover
  Network Aware

**Type of the generated source for the Resource Type**
 C
  ksh
  GDS

Step 1 of 2:

**Output Log**

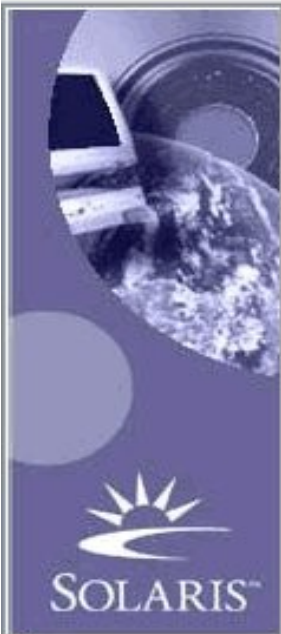
```

Creating a failover resource type.
Creating the rtconfig file ...done.
Cloning and modifying Makefile ...done.
Cloning and modifying README.apache ...done.
Cloning and modifying prototype ...done.
    
```

# Sun Cluster Agent Builder

## Configure Screen

File Edit



**Start Command (or file):**

Browse ...

**Timeout (in Secs):**

**Stop Command (optional):**

Browse ...

**Timeout (in Secs):**

**Validate Command (optional):**

Browse ...

**Timeout (in Secs):**

**Probe Command (optional):**

Browse ...

**Timeout (in Secs):**

Step 2 of 2:

Configure

<<Previous

Next>>

Exit

Output Log

```

Creating a failover resource type.
Creating the rtconfig file ...done.
Cloning and modifying Makefile ...done.
Cloning and modifying README.apache ...done.
Cloning and modifying prototype ...done.
    
```



# Developing Application Agents on Solaris Cluster

## Exercise

# Agent Development Exercise

- Double check that Apache is setup correctly
  - Start it on BOTH nodes with */usr/apache2/bin/apachectl start*
    - Check */var/apache2/logs/error\_log* in case of failures
  - Start web browser and connect with apache
  - Stop apache
- Start scdsbuilder and create a custom Apache Agent
  - Remember to set your DISPLAY shell variable
  - Suggest using GDS, Failover and Network aware
  - Use your own start, stop, validate and probes
  - Package would be created in working directory/pkg
  - Pkgadd it on all cluster nodes

# Exercise contd...

- Deploy your Agent
  - Run `/ws/galileo/tools/bin/labinfo -v $your-cluster-name`
    - Look under the headings “Failover ADDRESSES” and “Shared ADDRESSES” to find out available HA hostnames
    - Make sure the HA address you are going to use is not configured already
  - Run `/opt/$pkgname/util/startapache -h <logical-hostname> -p 80/tcp`
- Test your Agent
  - Kill the Apache processes, they should be restarted
  - Reboot the node running Apache, it should fail over to the other node



# Data Service Configuration Wizard

# Feature Overview

- Configurator for Data Services
- Wizards for
  - HA Oracle
  - Oracle Real Application Cluster (RAC)
  - SAP Web Application Server
  - Apache Webserver
  - NFS
  - Logical Host, Shared Address, HA StoragePlus
- Command Line and Browser-based

# Command Line Based Wizards

```
*** Main Menu ***
```

```
Please select from one of the following options:
```

- 1) Quorum
- 2) Resource groups
- 3) Data Services
- 4) Cluster interconnect
- 5) Device groups and volumes
- 6) Private hostnames
- 7) New nodes
- 8) Other cluster tasks
  
- ?) Help with menu options
- q) Quit

```
Option: 3
```

# Command Line Based Wizards

```
*** Data Services Menu ***
```

```
Please select from one of the following options:
```

- 1) Apache Web Server
- 2) Oracle
- 3) NFS
- 4) Oracle Real Application Clusters
- 5) SAP Web Application Server
- 6) Highly Available Storage
- 7) Logical Hostname
- 8) Shared Address
  
- ?) Help
- q) Return to the Main Menu


```
Option: 
```

# Browser Based Wizards

APPLICATIONS | VERSION
REFRESH | LOG OUT | HELP

User: root Server: pslayer4

## Sun™ Cluster Manager



Sun™ Microsystems, Inc.

slayer

**Tasks**

- ▶ Nodes
- ▼ Resource Groups
  - ▶ global\_locafs-rg
  - ▶ kerberos-rg
  - ▶ qfsmnds-rg
  - ▶ rac-framework-rg
  - ▶ scalmnt-rg
- ▶ Storage
- ▶ Private Interconnects
- ▶ IPMP Groups
- ▶ Quorum
- ▶ SNMP Modules

### Tasks

#### Configure Data Services for Applications

- Apache Web Server i
- Oracle i
- NFS i
- SAP Web Application Server i
- Oracle Real Application Clusters i

#### Configure Sun Cluster Resources

- Highly Available Storage i
- Logical Hostname i
- Shared Address i

# What it does

- Simplify the typical configuration
- Detailed knowledge of Solaris Cluster RGM concepts is NOT a pre-requisite for configuration.

# What it does not do

- Support all configuration possibilities
- Hide resource group and resource concepts
- Manage an existing data service

# How ?

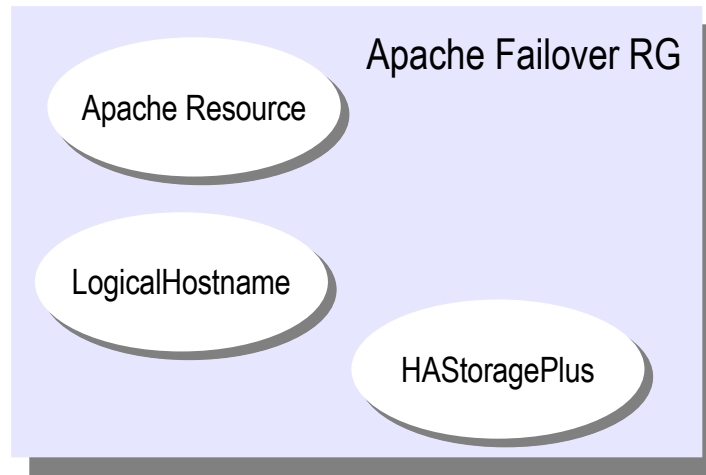
- Present a list of possible options, wherever possible
- Automate configuration on all the nodes of the dataservice
- Auto-generation of resource and RG names
- Auto-selection of resource group, based on user choices
- Rollback configuration changes on failure

# Configuration Choices

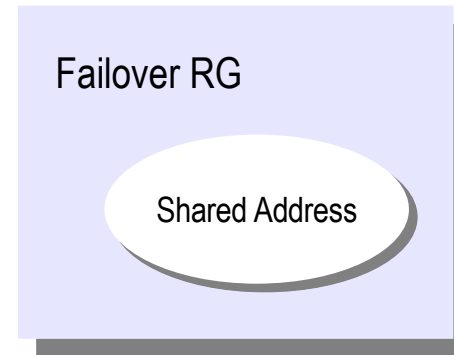
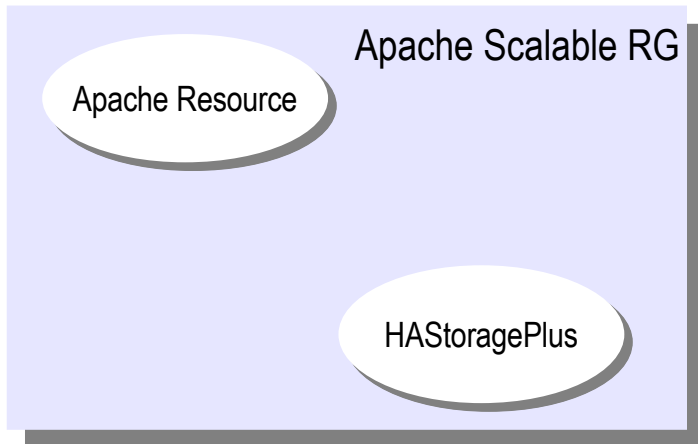
- Enforces a highly available storage resource for configurations
- Option to use existing resources or create new ones for storage and networking resources

# Apache Configuration

## Failover Config



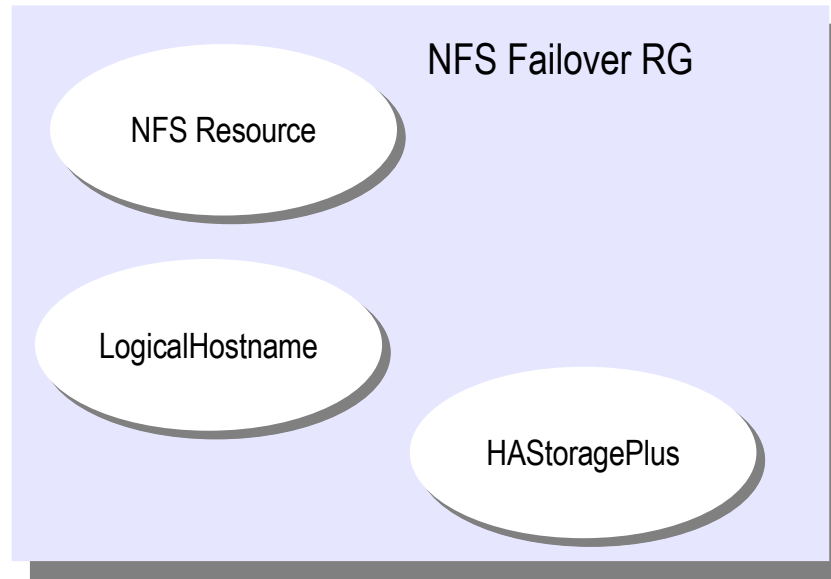
## Scalable Config



Resource Group Dependency

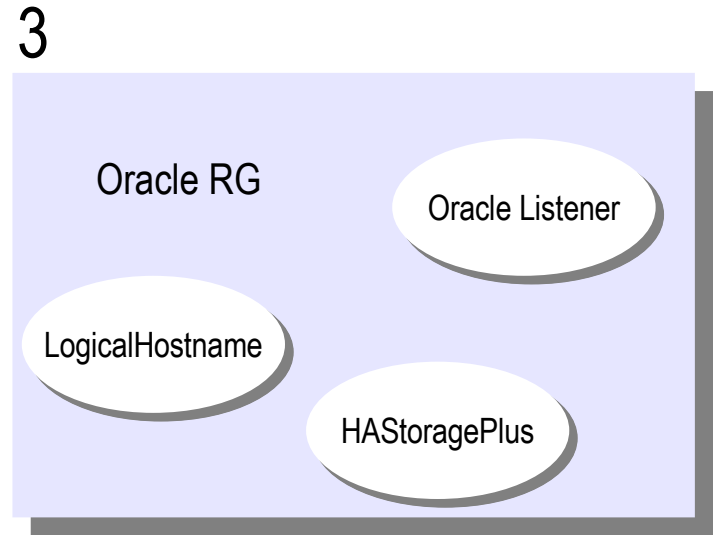
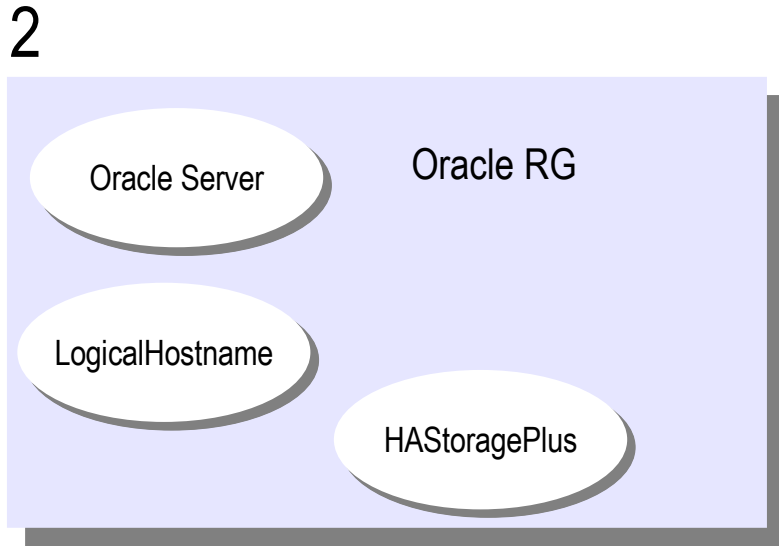
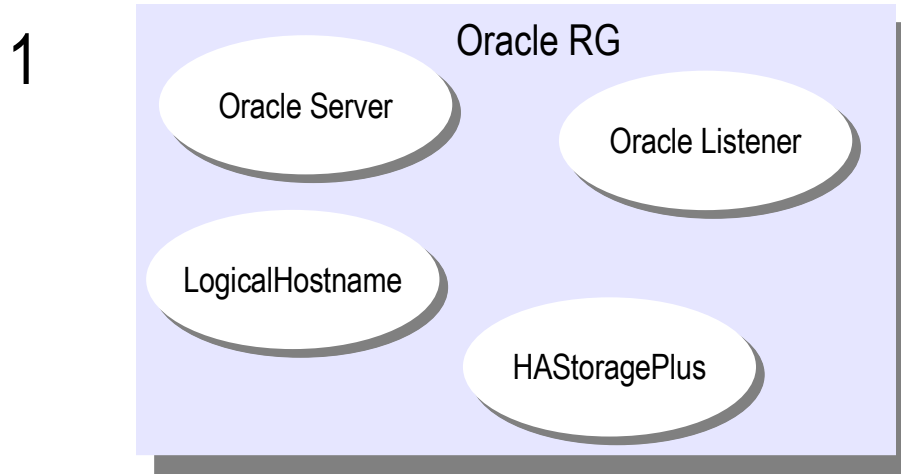
Resource Dependency

# NFS Configuration



Resource Dependency

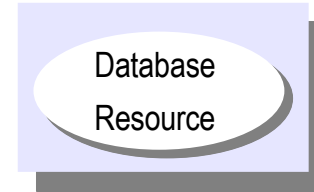
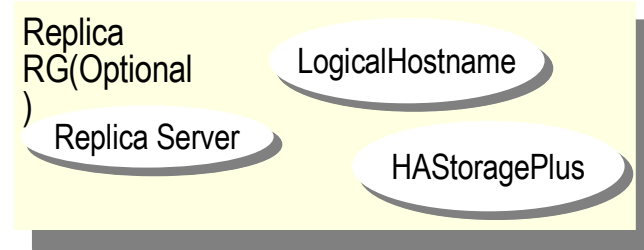
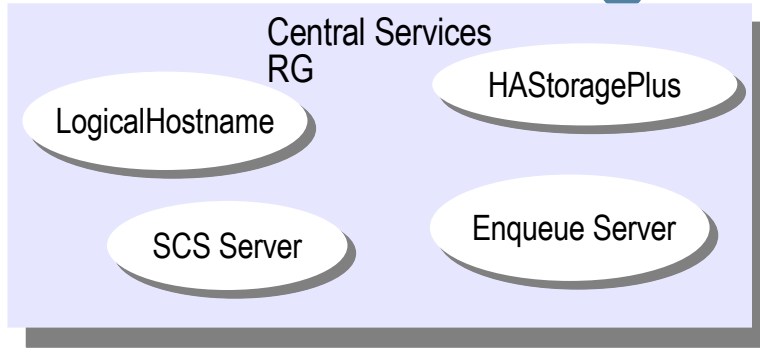
# Oracle Configuration



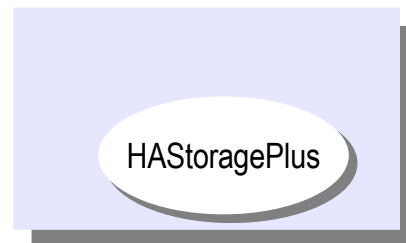
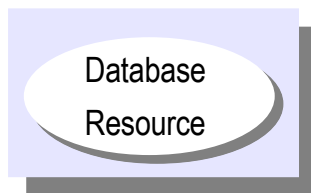
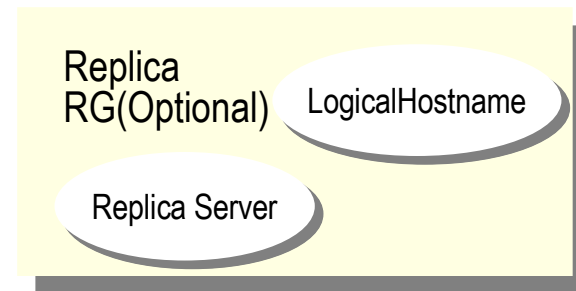
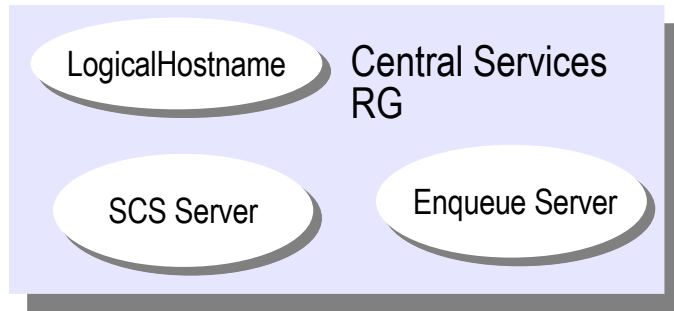
Resource Dependency

# SAP WebAS Configuration

1



2



Strong Negative Resource Group Affinity

Weak Positive Resource Group Affinity

Resource Dependency

# Oracle RAC Configuration

- RAC Framework
- Storage Resources for RAC
  - > All storage schemes supported SC3.2 are supported by the wizards
- Database Resources for RAC
  - > Oracle RAC version 9i
  - > Oracle RAC version 10gR2
- All 6 sample configurations mentioned in RAC guide and more.

# Log Files

- Command Log
  - > `/var/cluster/logs/commandlog`



# Solaris Cluster for Oracle RAC

# Agenda

- Why Solaris Cluster solutions for Oracle Deployments
- Summary of Solaris Cluster offerings for Oracle
- Solaris Cluster Advanced Edition for Oracle RAC
- RAC Manageability features in SC3.2

# Solaris Cluster Solutions for Oracle

## Solaris Cluster HA Oracle Agent

Non-RAC solution for:

- Single instance DB
- Support for Oracle in Solaris containers

Oracle DB

HA Oracle Agent

SC

Solaris

## Solaris Cluster Oracle RAC Agent

Basic RAC solution for:

- RAC on raw devices (ASM or Veritas)
- Customers using RAC on NAS
- Support for 8 RAC nodes

RAC

RAC Agent

SC

Solaris

## Solaris Cluster Advanced Edition for Oracle RAC

Sun-only advanced solution for:

- Customers who want ease of shared file system as well as clustered volume manager
- Support for 8 RAC nodes

RAC

RAC Agent

QFS

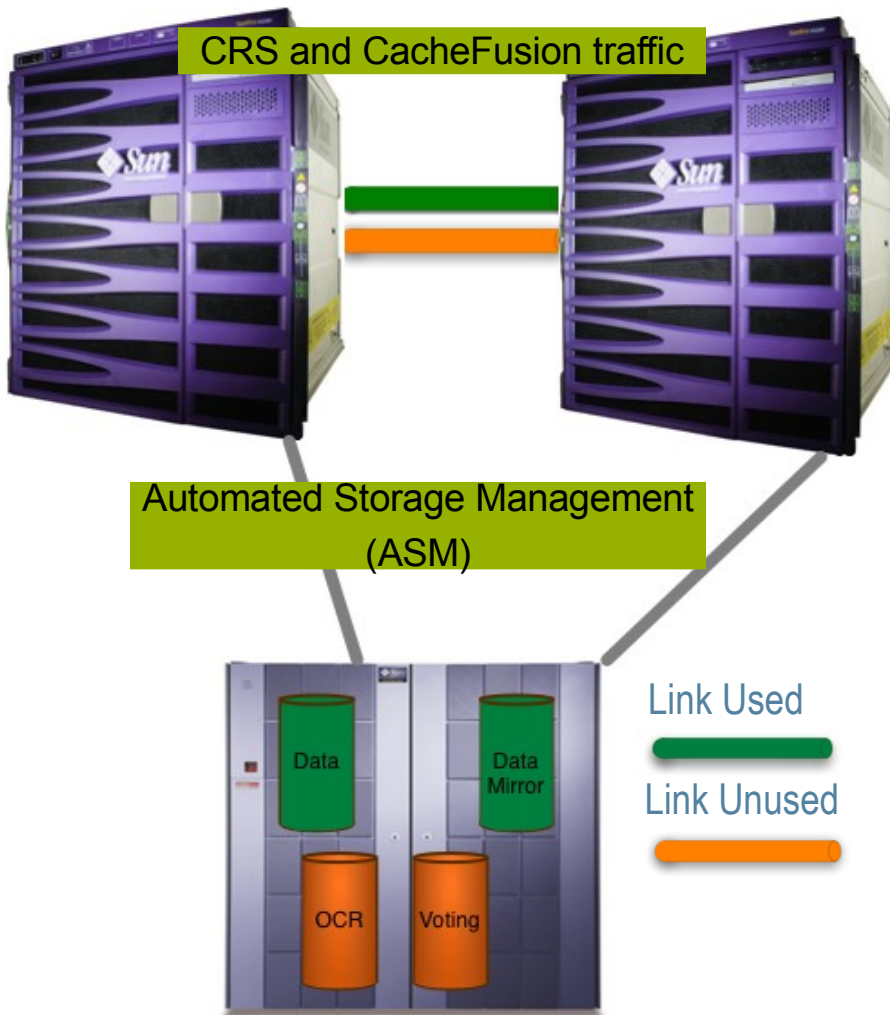
Advanced Edition

SC-SVM

SC

Solaris

# Why 10g CRS alone is not enough



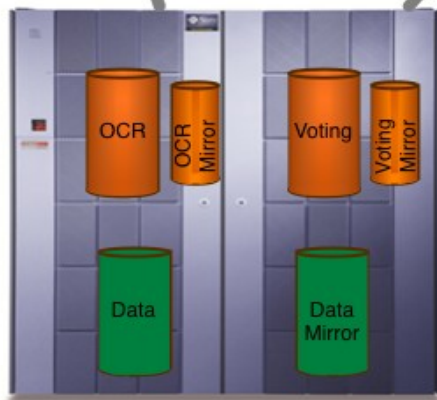
- Split-brain a real possibility and serious threat for data integrity
  - > CRS does not have IO fencing
- Redundancy for OCR and Voting disks not straight forward
  - > ASM mirrors data only not OCR or Voting disks
  - > 3 devices to protect single Voting disk failure
- Redundant interconnects not used by CRS; no failover if link fails
  - > IPMP-based solutions have limitations

# Value Proposition (1/2)



Filesystem with Volume Manager

Virtualized access to shared storage via DID Naming



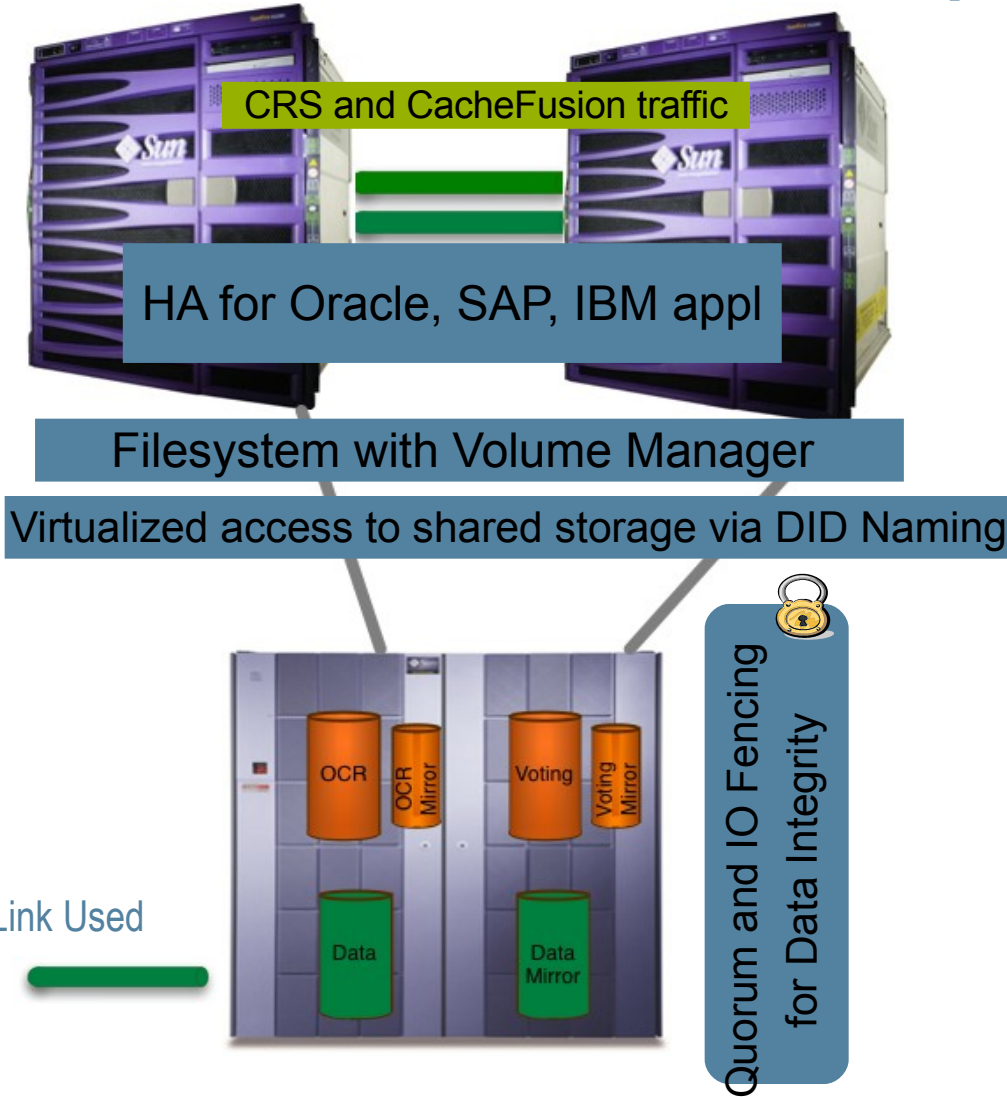
Quorum and IO Fencing  
for Data Integrity

- Heartbeats in kernel and more resilient to load, faster node failure detection
- Device-level IO Fencing prevents data corruption
- All interconnect links are used with automatic failover built-in
  - > Up to 6 links supported
  - > Under a single pseudo interface supporting all types of IP traffic
  - > Specialized interconnects with RSM/SCI now and RDS/IB under development
- CacheFusion traffic is striped over private interconnects resulting in higher throughput and lower latency for clients

Link Used

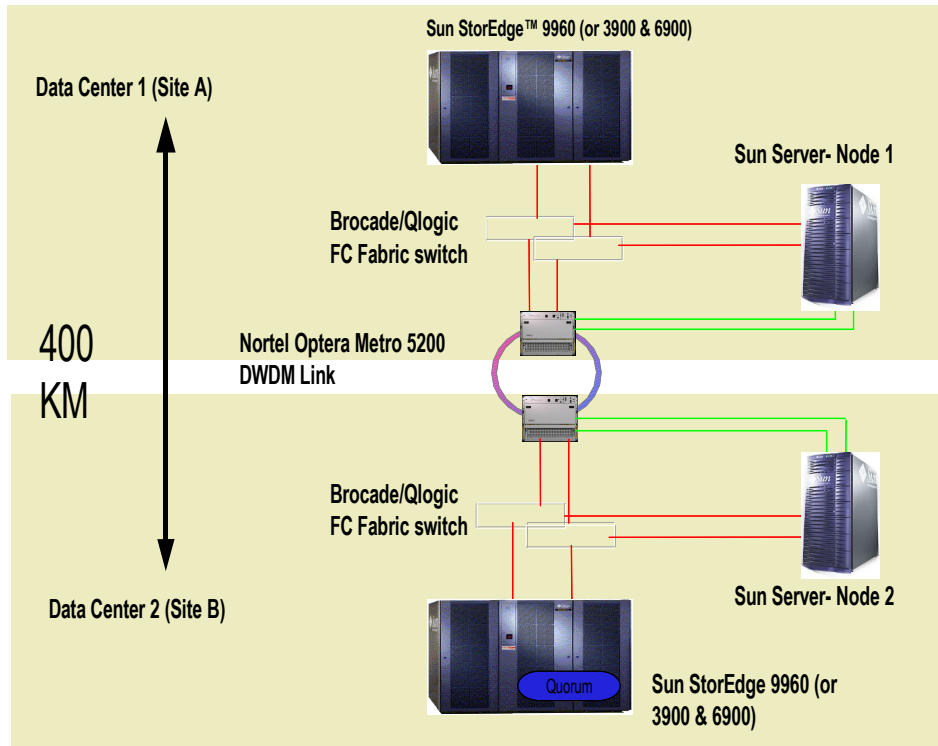


# Value Proposition (2/2)



- > DID (device ID) driver provides unique name
- > Simpler redundancy for Voting and OCR disks
- > No conflict in cluster membership
  - CRS integration with SC
- > HA for heterogeneous enterprise applications
- > ORACLE\_HOME and CRS\_HOME on shared storage
- > Automatic NTP services setup to maintain time
- > Support for 9i RAC
- > Continuous qualification testing for new HW

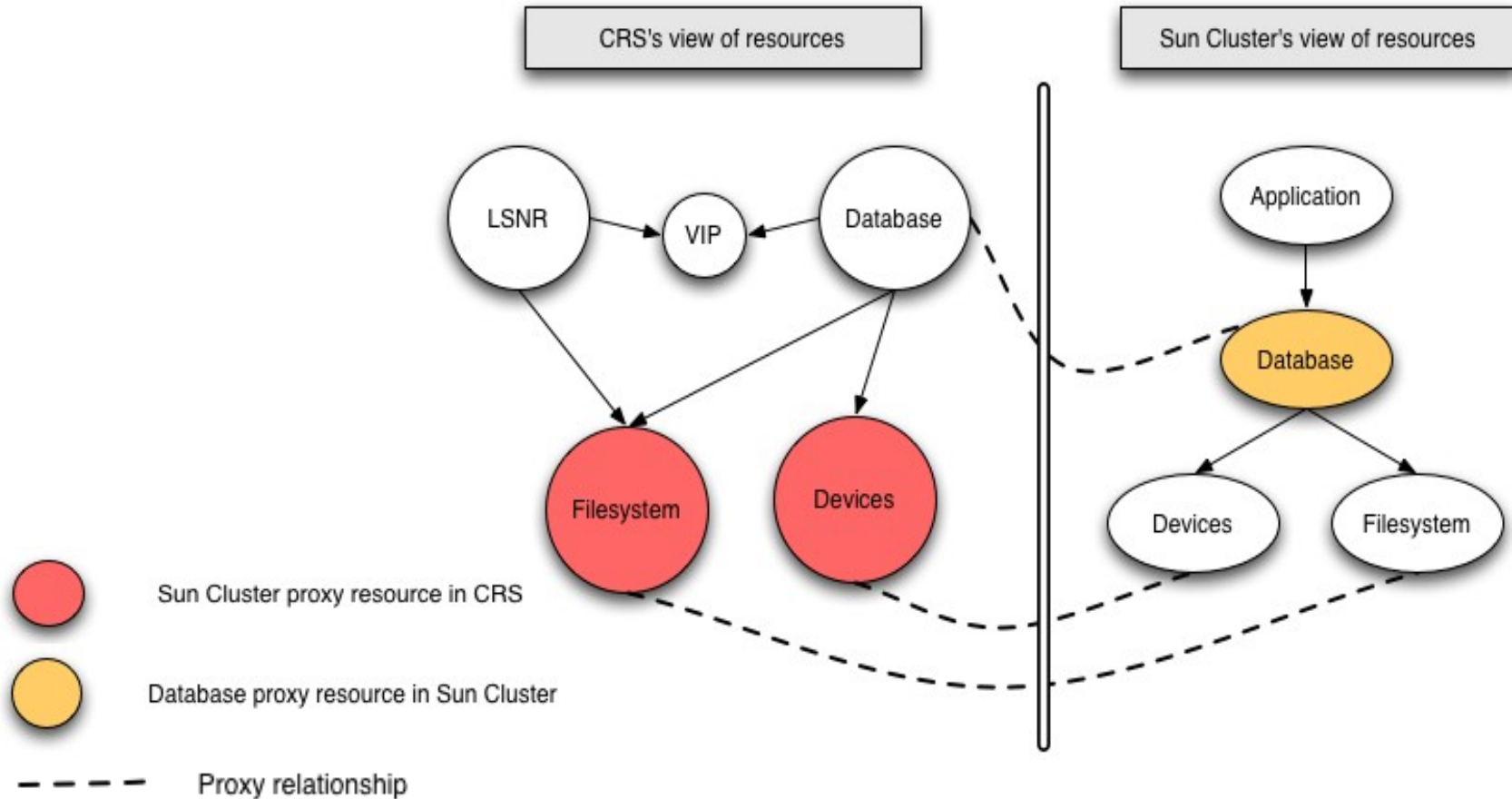
# Business Continuity for RAC



- Long Distance Clusters: active/active hundreds of km
  - > Supports inter-site cluster with Oracle RAC
- Unlimited distance with Sun Cluster Geo Edition
  - > RAC 9i with TrueCopy now
  - > Under development:
    - 10g RAC with TrueCopy, EMC SRDF
    - Data Guard

# RAC manageability - Improved 10gR2 integration

Sun Cluster and CRS Integration



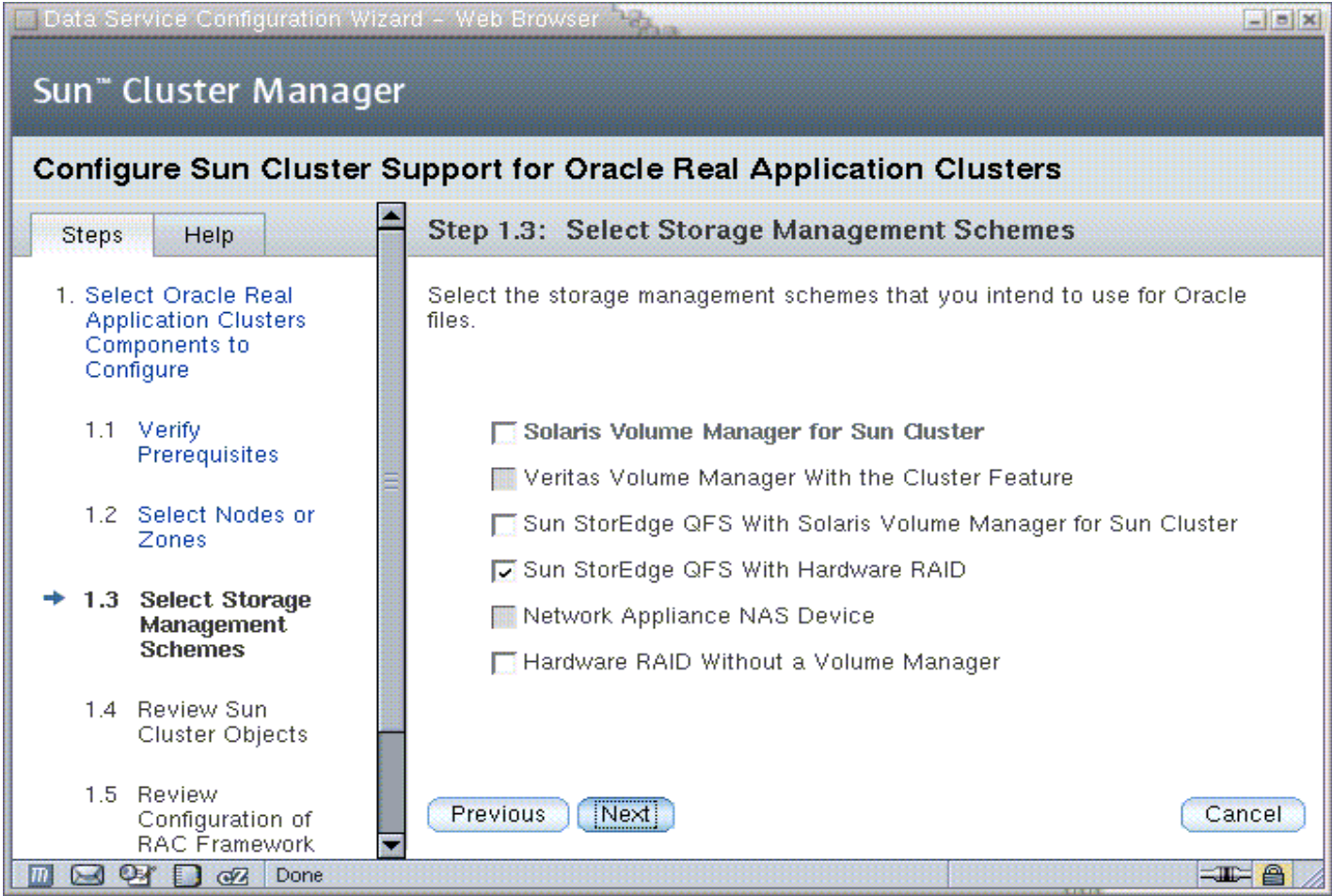
# RAC manageability – Overview (1)

- Direct Management of Oracle 9i RAC DB as a single scalable Resource Group
  - > Scalable resource type for 9i RAC DB server instance
  - > Scalable resource type for 9i RAC listener
    - Availability of specific instances on a per-node basis
    - Same fault monitoring capabilities as in HA-Oracle agent
- Proxy Management of Oracle 10gR2 RAC DB via a single scalable Resource Group
  - > Scalable proxy resource type for 10gR2 RAC DB instance
  - > Use 10gR2 public interfaces (SRVCTL, CRS CLI, FAN) to interact with Oracle cluster management functionalities
    - Availability of specific instances on a per-node basis
    - Rely on CRS fault monitoring and restart capabilities

# RAC manageability – Overview (2)

- Enabling 10gR2 CRS to take into account storage availability
  - > CRS resource (implementation provided by Solaris Cluster SW) to proxy the SC resource for shared diskgroup or shared mount points
- RAC framework Resource Group evolution
  - > Reconfiguration only involves modules which have corresponding resource created (rac\_udlm, rac\_svm, etc.)
    - The presence of the RG and its resources is mandatory
  - > New SUNW.crs\_framework resource type to coordinate node shutdown
    - Applicable only to 10g RAC
    - Not involved in reconfiguration mechanism

# Configuration Wizard for RAC



# Summary

	Feature	Function	Benefit
1	IO Fencing	Fence off shared storage from a bad/ill-behaved node	Guarantees data integrity and does not allow split-brain scenario
2	Redundancy of interconnects via private interfaces	Provide a logical network interface (with a separate network address) to the applications	Protect against single point of failure of NICs or Switches
3	Application traffic striping and transparent failover of interconnects	Stripe network traffic over multiple interconnects and handle failures transparently	Much better network utilization & higher application throughput
4	Choice of specialized interconnects	Leverage high-bandwidth low latency interconnects for all applications not just Oracle	More choice. Enables higher throughputs
5	Uniform device path names	Same name on all nodes for each shared LUN	Managing shared LUNs becomes much easier
6	Node Time synchronization	Synchronizes time on all cluster nodes	Allows inter-process communication across the nodes. Time synchronization is a key requirement for Oracle RAC
7	Shared QFS as a filesystem	A cluster filesystem for RAC	Improves manageability of files and data without compromising on the performance.  Also, provides more redundancy options for Oracle CRS OCR and Voting disks.  OCR and Voting can reside on the filesystem.
8	Support for 9i and 10g	Easier migration from RAC 9i environment to 10g.	CRS is available only with 10g. Better investment & knowledge protection from a 9i environment
9	Large application (agents) portfolio	Eliminates additional work in making applications (SAP, IBM etc.) HA	Greater value out-of-box
10	HA Oracle or RAC	With Sun cluster customers can choose the HA option that best meets their requirements	More choice
11	Tight integration with Solaris	Better fault management and availability due to kernel code. Heartbeats, more reliable and tunable	Higher Availability compared to VCS or CRS. Harder to <i>break</i> Oracle with Sun Cluster on Solaris
12	Guarantee of a tested, certified and supported configuration	Test every configuration	Customer does not have to worry about quality or support

# Solaris Cluster Advanced Edition

- Proven and mature Sun Cluster monitoring and failover
- A high performance filesystem to improve manageability
- Robust Solaris Volume Manager for data mirroring and availability
- Supports up to 8-node RAC configurations.



# **HA ZFS: ZFS as a highly available file system**

# Agenda

- ZFS Overview
- HAStoragePlus Overview
- ZFS and HAStoragePlus
- Differences Between ZFS and Other File Systems with HAS+
- Documentation
- Questions
- Exercises

# ZFS Overview

- “Blank page” file system
  - > Throw out old assumptions about file systems
  - > Start fresh without encumbrances
- Easy administration
  - > Scales to 1000s of file systems
  - > Properties per file system
    - inheritable
    - includes mount point, share information
- Very high data integrity
  - > Detects “silent” data corruption
  - > Self healing when redundant (mirror or RAID-Z)

# HAStoragePlus Overview

- Manages dependencies between applications and storage
  - > Dependencies described by administrator
- Co-locates application with file system
  - > Or disk group or metaset, for other file systems
- Manages file system fail-over

# ZFS and HAStoragePlus

- Provides fail-over capability for ZFS file systems
- Fail-over unit is the zpool
  - > All file systems in a zpool fail over together
- Much like other file systems managed by HAS+

# Differences Between ZFS and Other File Systems with HAS+

- Easy to administer
  - > No entries required in /etc/vfstab
  - > ZFS “legacy” mount points disallowed
- Extension properties
  - > Use FilesystemMountPoints for UFS or VxFS
  - > Use Zpools property to configure ZFS pools
- Managed at the “zpool” level
  - > Multiple file systems within a zpool managed together
  - > Separate zpools required for independent fail-over
    - Somewhat at odds with space-sharing design of ZFS

# Differences Between ZFS and Other File Systems with HAS+ (2)

- New HAS+ Property “Zpools”
  - > Lists one or more zpools managed by a given HAS+ resource
  - > All listed zpools fail over together
- Only fail-over resource groups allowed
  - > ZFS cannot be used with a scalable resource group
- ZFS “share” property not compatible with HA-NFS
  - > Use existing HA-NFS mechanisms, instead

# Documentation

- Man pages
  - > SUNW.HAStoragePlus(5HA)
  - > zpool(1M)
  - > zfs(1M)
  - > Sun Cluster Data Services Planning and Administration Guide for Solaris OS
    - <http://docs.sun.com/app/docs/doc/819-2974>
    - Specific to HAStoragePlus and ZFS
      - <http://docs.sun.com/app/docs/doc/819-2974/6n57pdk11?q=zfs&a=view>

## Exercise-1 (creating rg with ZFS)

1. Create a zpool named “hazpool” with a few file systems. (use `zpool(1M)` and `zfs(1M)`).
2. Create a failover resource-group with a resource of type `SUNW.HAStoragePlus` and configure the pool “hazpool” to the resource in `Zpools` extension property. (use `clrg(1CL)`, `clrs(1CL)`)
3. Bring the resource group on line.
4. Check for file systems on the node where resource group is on line.

## Exercise-2 (Checking switchover and failover)

1. Do resource group switchover from one node to another and check for file systems on both nodes.
2. Reboot/Panic the node where resource group is on line. Check for file systems on the node where the resource group is failed over.

# (Sample) Solutions to exercises

## **zpool/zfs creation**

```
phost1# zpool create hazpool mirror c1t1d0 c2t1d0
```

```
phost1# zfs create hazpool/home
```

```
phost1# zfs list
```

## **Resource group creation**

```
phost1# clrg create hasp-rg
```

```
phost1# clrt register SUNW.HAStoragePlus
```

```
phost1# clrs create -g hasp-rg -t SUNW.HAStoragePlus \  
-p Zpools=hazpool hasp-rs
```

## **Bring resource group on line**

```
phost1# clrg online -eM hasp-rg
```

```
phost1# zfs list
```

# (Sample) Solutions to exercises

## **ZFS switchover**

```
phost1# clrg switch -n phost2 hasp-rg  
phost1# zfs list
```

## **ZFS failover**

```
phost2# zfs list  
phost2# uadmin 1 1  
phost1# zfs list
```



# **Business Continuity with Sun Cluster Geographic Edition 3.2**



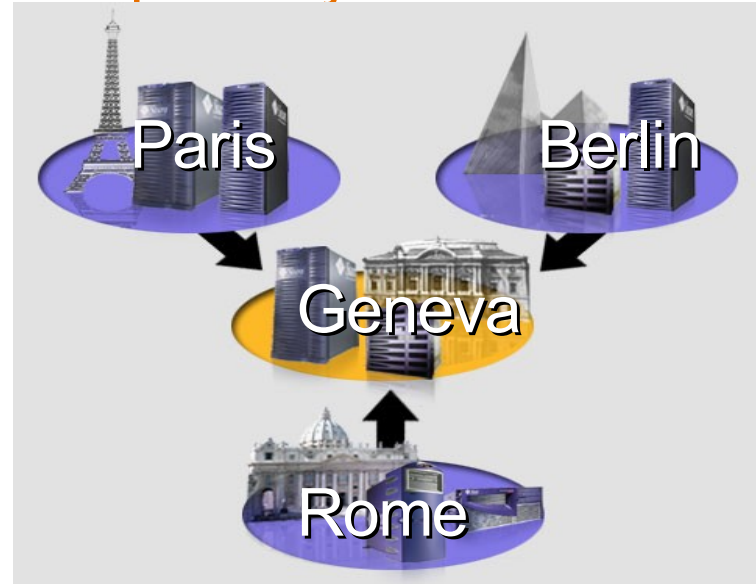
# Sun Cluster Geographic Edition

## Multi-Cluster and Multi-Site capability

### N+1 Multi-site Support

One site backs up multiple cities

One way Data Replication



### Active – Active Configuration

Each site backs up the other

Bi-directional Data Replication



# Sun Cluster Geographic Edition

## Architecture

- Layered Extension of standard Sun Cluster
  - > Built on Sun Cluster 3.2, for Solaris™ 9 (SPARC) and 10 (SPARC,x64)
- Builds on same hierarchical concepts
  - > Partnership
    - **Pair of clusters**
  - > Protection Group
    - **Combination of Resource Groups and Replication resources**
    - **PG configuration propagated between partners**
    - **One “Big Red Button” switch-over of PG**
  - > Heartbeat for connection monitoring
    - **Built-in standard TCP/IP heartbeat, optional plug-ins**
    - **Alerts by email, SNMP trap, custom script**
  - > Action script
    - **Can be used to fail-over name server entries**



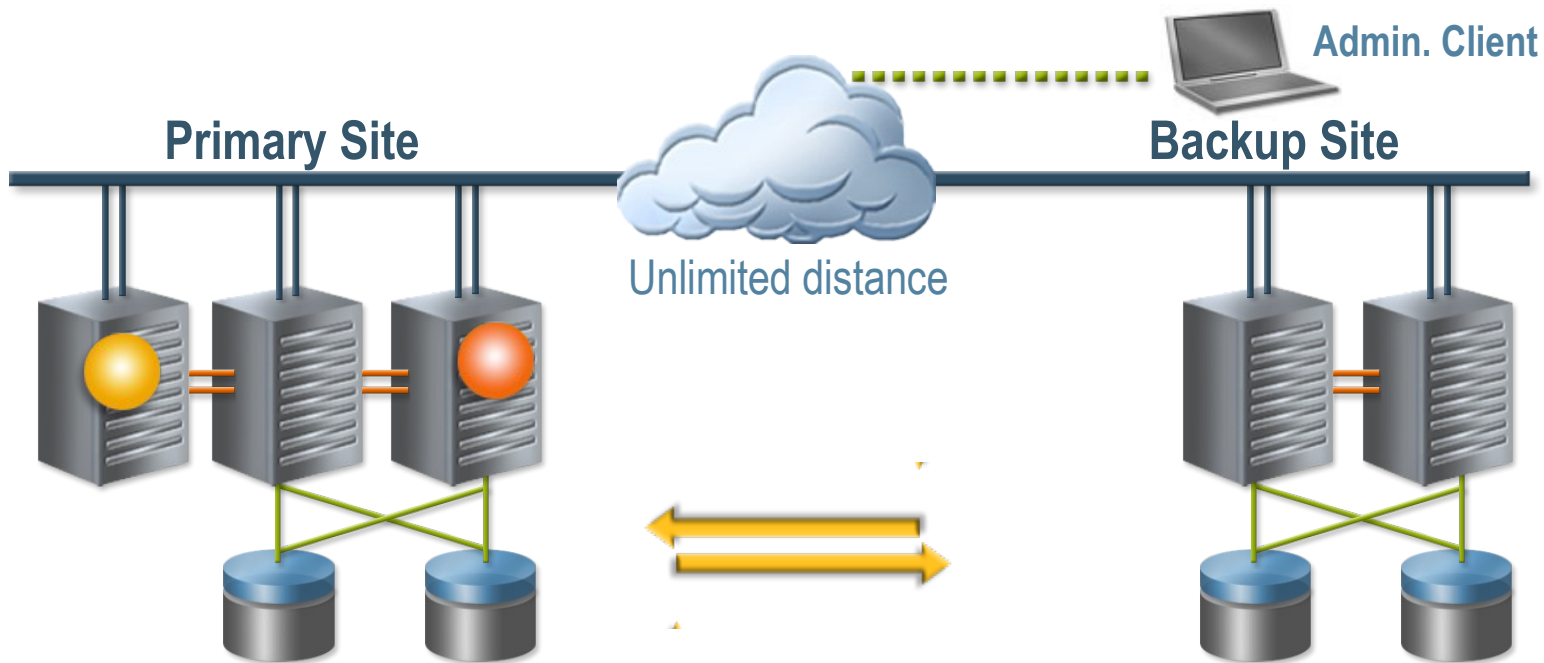
# Sun Cluster Geographic Edition

## Objects and modes of operation

- Protection Group (PG) is the unit of a switchover/takeover
  - > SC3.2 Resource Group(s) + Storage replication object (optional)
- Planned Switchover
  - > Shuts down PG on one cluster
  - > Reconfigures replication
  - > Restarts PG on new cluster
  - > No data loss.
- Emergency Takeover
  - > Will start PG on new cluster even if partner is unreachable
  - > Attempts to reach partner cluster to synchronize
  - > Brings replicated copy of data online on new cluster
  - > Very simple command, e.g. “**geopg takeover *pgname***”
  - > Some risk of lost data updates, if storage was not fully synchronized

# Sun Cluster Geographic Edition

## Configuration options



### Data Replication

- Sun StorageTek Availability Suite (AVS) with any Sun Cluster supported storage
- StorEdge 99x0 TrueCopy with Sun StorEdge 9910/9960/9970/9980
- EMC SRDF

solaris 9 and 10
 ULTRASPARC  
 solaris 10
 AMD

# Sun Cluster Geographic Edition

## Data Replication Technologies

- Sun StorageTek Availability Suite (AVS)
  - > Host-based
    - Replication performed by host system
  - > Can be used in Synchronous or Asynchronous mode
- Sun StorEdge 99x0 TrueCopy
  - > Storage-based
    - Replication performed by storage array
  - > No special Geographic Edition requirements
  - > Can be used in Synchronous or Asynchronous mode
- EMC<sup>2</sup> SRDF
  - > Similar characteristics to TrueCopy
  - > Some limitations when used in Asynchronous (SRDF/A) mode

# Sun Cluster Geographic Edition

## Management

- Partnership management
  - > Command Line Interface
    - **geops, geopg, geoadm, geohb**
  - > Sun Cluster Manager
    - **Browser-based tool**
    - **Hierarchical management model**
    - **Same tool as used for individual clusters**
- Partner Loss notification module
  - > Configurable, based on Sun Cluster Events
  - > Three types of notification action
    - **Email alert**
    - **SNMP Trap**
    - **Action script**

# Sun Cluster Geographic Edition

## Command-Line Interface

- CLI

- > Modelled on new Sun Cluster 3.2 CLI
- > Partnership operations
  - **geops add-trust -c <remotecluster>**
  - **geops create -c <remotecluster> pshipname**
  - **geops join-partnership remotecluster pshipname**
  - **geops update pshipname**
- > Protection group operations
  - **geopg create pgname -s pshipname -o localrole -d datareptype**
  - **geopg update pgname**
  - **geopg takeover pgname**
- > Framework control
  - **geoadm start**
  - **geoadm status**

# Sun Cluster Geographic Edition

## Application support

- General rule – Geographic Edition supports any application that is supported by Sun Cluster 3.2
- Applications may have replication issues
  - > Some may require Synchronous replication
  - > Some scalable applications (e.g. Oracle RAC) cannot work with host-based replication
- Sun is testing principal applications
  - > Oracle HA and RAC 10g
  - > SAP
  - > etc.

# Sun Cluster Geographic Edition

New features in SCGE 3.2, compared with SCGE 3.1 8/05

- AVS 4.0 support for Solaris 10 (SPARC and x64)
- EMC<sup>2</sup> SRDF support
- Improved heartbeat performance
- GUI improvements
- Easier setup (**geops add-trust**)
- SNMP notifications
- Common Agent Container (“Cacao”) 2.0

# Sun Cluster Geographic Edition

## Futures

- As qualifications
  - > SVM support with external replication
  - > TrueCopy for x64
  - > Oracle 10g with TrueCopy
  - > Match SC3.2 support for Solaris Zones
- Planned for next release
  - > Oracle Data Guard
  - > SMI-S Copy Services control

# Sun Cluster – Geographic Edition

## Demo description

- Oracle client GUI running on external desktop
  - > Connects to the application on the primary cluster
- GUI writes records into the database which include physical hostname of Oracle host
- Oracle 'switched over' to secondary cluster
  - > If primary site is lost to disaster, secondary site 'takes over' the service
- After the switchover, the new primary cluster will have all the records of the replicated data, and new data can still be added.

# Backup Slides

# AVS Tuning

## Asynchronous mode

- Based on IO queues
  - > Maintains write ordering
  - > Multiple threads used to empty queues
  - > Threads use synchronous kernel RPC
- Very dependent on application write pattern
  - > Sequential writes coalesced
  - > Random writes require 1 RPC per write
- Tuning dependent on link data rate and latency (RTT)
  - > Synchronous RPC:  $1/\text{RTT}$  IOPs per thread per second
  - > Use multiple threads to match to link data rate
- TCP buffer/window size also important
  - > Optimal size is  $2 * \text{RTT} * \text{datarate}$
  - > Careful: buffer allocation is **per connection**

# Sun Cluster Geographic Edition

## AVS data replication (SNDR)

- AVS 3.2.1 on S9 (SPARC), AVS 4.0 on S10 (SPARC, x64)
- Host Based, IP connectivity
  - > A node in the cluster which owns the PG is AVS primary
  - > Corresponding node in partner cluster is AVS secondary
- Volume manager support
  - > Today: Solaris VM (SPARC, x64), VxVM 5.0 (SPARC)
  - > Q1CY07: VxVM 4.1 (x64)
- Can be used in Synchronous or Asynchronous mode
  - > Synchronous
    - Disk write performance limited by network connectivity
  - > Asynchronous
    - Network does not affect local write performance
    - Tuning is critical for asynchronous replication performance

# Sun Cluster Geographic Edition

## TrueCopy™ data replication

- Storage Based
  - > Cluster which owns the PG has primary array (PVOL)
  - > Corresponding storage unit in partnership is secondary (SVOL)
- Volume manager support
  - > Today: VxVM 5.0 (SPARC)
  - > Q1CY07: VxVM 4.1 (x64), SVM (SPARC, x64)
- No special Geographic Edition requirements
  - > Standard SunCluster support
- Can be used in Synchronous or Asynchronous mode
  - > As for AVS

# Sun Cluster Geographic Edition

## EMC<sup>2</sup> SRDF data replication

- Storage Based
  - > Cluster which owns the PG has primary array (R1)
  - > Corresponding storage unit in partner cluster is secondary (R2)
- Volume manager support
  - > Today: VxVM 5.0 (SPARC)
  - > Q1CY07: SVM (SPARC)
- No special Geographic Edition requirements
  - > Standard SunCluster support
  - > SRDF not yet available on x64
- Can be used in Synchronous or Asynchronous (/A) mode
  - > **NB:** SRDF/A does not permit role swap, so only Takeover is allowed
  - > “Domino” mode cannot be used.



**More informations ?**

# For further information

- Check out Solaris Cluster blogs - <http://blogs.sun.com/SC>
- Discuss Solaris Cluster @ Sun Developer Network (SDN)  
<http://forum.java.sun.com/forum.jspa?forumID=842>
- Check out <http://www.sun.com/cluster>
- Download Solaris Cluster software @  
<http://www.sun.com/download/products.xml?id=4581ab9e>
- Get trained in Solaris Cluster @  
<http://www.sun.com/training/catalog/server/cluster.xml>
- SC Documentation @ <http://docs.sun.com>
  - > SC 3.2 Documentation Center  
<http://docs.sun.com/app/docs/doc/820-0335/>



# Solaris Cluster Roadshow End

