IBM IDS and High Availability on Sun Cluster 3.0 Update 1

Abstract

We describe the implementation and design of IBM IDS highly available environments on the Sun Cluster 3.0 Update 1 (SC3.0) platform. Included is a detailed description of the IBMids HA agent for SC3.0, and recommendations for highly available strategies. Practical considerations regarding design, implementation, testing and performance work with SC3.0 are also discussed.

Overview

It is presumed that the reader is familiar with the motivation for the implementation of a highly available (HA) solution and with the basic terminology in this field. The IBMids agent was developed and tested on Sun Cluster 3.0 update 1 and Solaris 8 maintenance update 5 with associated patches. In addition, it is assumed that the reader has some experience with IBM IDS 9.x or 7.x and with SC3.0. However, some crucial overview material will be presented in the following sections in order to prepare the reader for the installation of the IBMids agent, and while we discuss the agent in terms of IBM IDS 9.x many of the topics remain the same for IBM IDS 7.x, but where there are major differences they will be noted.

IBM IDS

IBM IDS 9.x offers world-leading extensibility features. It also provides the mission-critical availability, reliability, scalability and data-intensive transaction management capabilities a growing number of organizations rely on to succeed in e-business. And because business logic can be embedded in the database and fully integrated into IBM Informix Dynamic Scalable Architecture™, IBM IDS 9.x can solve business problems few other systems can. What's more, IBM IDS 9.x can provide uniform access to client information (regardless of where it resides) via leading distributed database and virtual table capabilities - allowing queries to encompass a wide range of databases, from flat files to other IBM IDS and non-IBM IDS databases. For more information please visit www.ibm.com/software/data/informix/ids.

Sun Cluster 3.0

Sun Cluster 3.0 is the latest release of the Sun Microsystems clustering product delivering highly available scalable service in the Solaris8 operating environment. We will present a brief product summary; the reader is urged to refer to the sources listed herein for the definitive product documentation and specifications.

Hardware Requirements

The current list of hardware supported by SC3.0 is available at www.sun.com/software/cluster/detailed.html.

Software Requirements

Sun Cluster 3.0 Update 1 requires Solaris 8 Maintenance Update 4 or better on all nodes that will participate in the cluster. SC3.0 also recommends using a volume manager: either Solstice DiskSuite or VERITAS Volume Manager 3.0.4 or later.

Failover

SC3.0 provides high availability by enabling application failover. The state of each individual node is periodically monitored. The cluster software automatically relocates a cluster-aware application from a failed primary node to a designated secondary node. When a failover occurs, clients may see a brief interruption in service and may need to

reconnect after the failover has finished. However, clients are not aware of the physical server from which they are provided the application and data.

By allowing other nodes in a cluster to automatically host workload when the primary node becomes nonfunctional, SC3.0 can significantly reduces downtime and increases productivity by providing high availability service to all users.

Multihost Disks

SC3.0 requires multihost disk storage: disks that can be connected to more than one node at a time. In the SC3.0 environment, multihost storage allows disk devices to become highly available. Disk devices resident on the multihost storage can tolerate single node failures since there still exists a physical path to the data via the alternate server node.

Multihost disks can be accessed globally through a primary node; this node is said to master the disk. If client requests are accessing the data through one node and that node fails, the requests are switched over to use another node that has a direct connection to the same disks.

A volume manager provides for mirrored or RAID-5 configurations for data redundancy of the multihost disks. Currently, SC3.0 supports Solstice DiskSuite and VERITAS Volume Manager as volume managers. Combining multihost disks with disk mirroring and striping protects against both node failure and individual disk failure.

Global Devices

SC3.0 uses global devices to provide cluster-wide, highly available access to any device in a cluster, from any node, without regard to where the device is physically attached. All disks are included in the global namespace with an assigned device id (did) and are configured as global devices. Therefore, the disks themselves are visible from all cluster nodes.

Cluster File Systems/Global File Systems

A cluster file system, also referred as a global file system, is a proxy between the kernel on one node and the underlying file system volume manager running on a node that has a physical connection to the disk(s). Cluster file systems are dependent on global devices with a physical connection to one or more nodes. The cluster file system is independent of the underlying file system and volume manager. Currently, cluster file systems can be built on UFS using either Solstice DiskSuite or VERITAS Volume Manager. The data only becomes available to all nodes if the file systems on the disks are mounted globally as a cluster file system.

Device Groups

In SC3.0, all multihost disks must be under control of the Sun Cluster framework. Disk groups, managed by either Solstice DiskSuite or VERITAS Volume Manager, are first created on the multihost disk. Then, they are registered as Sun Cluster disk device groups. A disk device group is a type of global device. Multihost device groups are highly available. Disks are accessible through an alternate path if the node currently mastering the device group fails. The failure of the node mastering the device group does not affect access to the device group except for the time required to perform the recovery and consistency checks. During this time, all requests are blocked (transparently to the application) until the system makes the device group available.

Resource Group Manger (RGM)

The cluster facility known as the Resource Group Manager, or RGM, provides the mechanism for high availability. The RGM runs as a daemon on each cluster node and automatically starts and stops resources on selected nodes according to pre-configured policies. The RGM allows a resource to be highly available in the event of a node failure or reboot by stopping the resource on the affected node and starting it on another. The RGM also automatically starts and stops resource-specific monitors that can detect resource failures and relocate failing resources onto another node. It can also monitor other aspects of resource performance.

Data Services

The term data service is used to describe a third-party application that has been configured to run on a cluster rather than on a single server. A data service includes the application software and SC3.0 software that starts, stops, and monitors the application.

SC3.0 supplies data service methods that are used to control and monitor the application within the cluster. These methods run under the control of the RGM, which uses them to start, stop, and monitor the application on the cluster nodes. These methods, along with the cluster framework software and multihost disks, enable applications to become highly available data services. As highly available data services, they can prevent significant application interruptions after any single failure within the cluster. The failure could be to a node, an interface component, or to the application itself.

The RGM also manages resources in the cluster, including instances of an application and network resources (logical hostnames and shared addresses).

Resource Types, Resource Groups and Resources

A resource type consists of a software application to be run on the cluster, control programs used as callback methods by the RGM to manage the application as a cluster resource, and a set of properties that form part of the static configuration of a cluster. The RGM uses resource type properties to manage resources of a particular type. A

resource inherits the properties and values of its resource type. It is an instance of the underlying application running on the cluster. Each instantiation requires a unique name within the cluster.

Each resource must be configured in a resource group. The RGM brings all resources in a group online and offline together on the same node. When the RGM brings a resource group online or offline, it invokes callback methods on the individual resources in the group.

The nodes on which a resource group is currently online are referred to as its primaries or its primary nodes. A resource group is mastered by each of its primaries. Each resource group has an associated Nodelist property, set by the cluster administrator, which identifies all potential primaries or masters of the resource group.

IBM IDS Installation and Configuration in a Sun Cluster 3.0 Environment

The installation of IBM IDS in a SC3.0 environment closely mirrors that which is documented in the relevant *Installation Guide for Informix Dynamic Server* guide.

Cluster File System considerations

At this stage, it is presumed that SC3.0 is configured according to the information given in the Sun Cluster 3.0 U1 Installation Guide. Note that it is recommended that at least one cluster file system be defined and available, as it will be used to host the informix user's home directory and informix binaries directory (INFORMIXDIR) for each IBM IDS instance to be installed.

Please review the Sun Cluster 3.0 U1 Data Services Installation and Configuration Guide

with particular emphasis on the topic entitled *Determining the Location of Application Binaries*.

Briefly, there are two distinct binary location strategies: they can either exist on a global file system or they can exist on each physical node on a local file system. It is recommended that the IBM IDS binaries be placed on a global file system for the following reasons:

- Access to the binaries will be highly available (special steps will need to be taken to ensure that access to the binaries be highly available if they are locally installed and mounted)
- Installation of binary images need only be performed once (per set of cluster nodes which share the global binary installation mount points)
- There is no chance of having mismatched binary code levels across the nodes of the cluster
- Nodes which are added to the cluster automatically can access the IBM IDS binary.

While the cluster file system is not required to be used as the instance home directory mount point, its use is strongly recommended; otherwise, there are potential availability issues with respect to the instance home directory which must be considered should cluster file system not be utilized for this task.

User and Group Creation Considerations

On each physical node that will participate in the cluster, an informix group and user account needs to be created for the Informix DBA account.

If NIS or NIS+ are in use, groups and users must be created on the NIS server prior to executing the install programs. If using local server authentication, repeat the following steps on all the nodes in the cluster. The users and groups must be defined identically across all nodes in the cluster; ensure that the user id, group id, user home directory, account passwords and user shell are consistent across all nodes in the cluster.

For example, use the following Solaris command to create the informix group on each server in a local server authentication environment:

```
groupadd -g 1001 informix
```

Use the following Solaris command to create the informix user account, without creating home:

```
useradd -g informix -u 1001 -d /global/informix -m -s /bin/ksh informix
```

Note that for the these commands, we presume that /global/informix is the mount point of a cluster file system which will be utilized to host the INFORMIXDIR directory. Please ensure that the account passwords are consistent across all nodes in the cluster (use the Solaris `passwd` command to set them if necessary).

Miscellaneous Considerations

At this point, you are advised to consult the appropriate *Installation Guide for Informix Dynamic Server* for specifics on installation and initializations. One other important point, unchanged from a typical IBM IDS installations on Solaris install, is to ensure that the correct kernel parameters have been applied at each physical node in the cluster. These parameters can be found in the release notes delivered with the IBM IDS software.

Installation of IBM IDS Binary

It is recommended that the software be installed in a global file system that is mounted on both the primary and backup system. The database server should be installed with the standard procedures prescribed for that version of the server on the primary machine and initialized like a normal installation. Once that is complete, depending on the IBM IDS version being installed the following should be done.

IBM IDS 7.x: The same procedure should be followed on the backup system. Although a savvy DBA or system administrator may be able to create just the necessary links for the install to work correctly on the second machine, rerunning the installer on the second system won't affect the primary system and makes sure that all necessary steps have been completed on both machines.

IBM IDS 9.2x: On the backup system run the RUNASROOT.xxxxx files in the same order as the primary machine.

IBM IDS 9.3x: No further actions necessary.

Configuration of a Particular IBM IDS Instance

Become superuser on all nodes.

Configure the `/etc/nsswitch.conf` file as follows so that the Sun Cluster HA for IBM IDS data service starts and stops correctly if a switchover or failover occurs.

On each node that can master the logical host that runs the Sun Cluster HA for IBM IDS data service, include one of the following entries for group in the `/etc/nsswitch.conf` file.

```
group:
group: files
group: files [NOTFOUND=return] nis
group: file [NOTFOUND=return] nisplus
```

Refer to the Sun Cluster 3.0 U1 Installation Guide for further nsswitch.conf entries required by SC3.0

The Sun Cluster HA for IBM IDS data service uses the `su` *user command to start and stop the database node. The user is typically* `informix.` The network information name service might become unavailable when a cluster node's public network fails. Adding one of the preceding entries for group ensures that the `su(1M)` command does not refer to the NIS/NIS+ name services if the network information name service is unavailable.

Configure the cluster file system for the Sun Cluster HA for IBM IDS data service.

If raw devices contain the databases, configure the global devices for raw-device access. See the Sun Cluster 3.0 U1 Installation Guide *for information on how to configure global devices.*

When you use the Solstice(TM) DiskSuite volume manager, configure the IBM IDS software to use UNIX file system (UFS) logging on mirrored meta devices or raw-mirrored meta devices. See the Solstice DiskSuite documentation for more information on how to configure raw-mirrored meta devices.

On the primary node create the directory to be used for the software install on a global file system, as such:

```
cd /global/informix/
mkdir ids930uc1
chown informix:informix ids921fc5
chmod 755 ids921fc5
```

The IBM IDS software should be extracted into the previously created directory and the standard procedures for software installation should be followed for a normal installation with some minor exceptions that can be done in parallel. Add a hostname alias, like dbserver, on each system in the `/etc/hosts` file as shown below.

Example: `/etc/hosts` for a primary system named suninfo:

```
#
# Internet host table
#
127.0.0.1         localhost
192.168.0.201     pelican
192.168.0.202     suninfo     loghost dbserver
```

and a backup system named pelican:

```
#
# Internet host table
#
127.0.0.1         localhost
192.168.0.201     pelican     loghost dbserver
192.168.0.202     suninfo
```

Next make the appropriate entry in `/etc/services` for the service name to be used in the sqlhosts file on each machine, for example:

Example: `/etc/services` entry:

```
online      1526/tcp
```

Then configure the onconfig and sqlhosts files with the appropriate information for the instance based upon those entries. This only needs to be done on one machine since the software is installed on a global file system. Examples of these files are in Appendix A.

Once it has been verified that the instance can be started and stopped with the normal procedures, there is one more step to perform on the backup system. Execute the RUNASROOT.xxxxx in the same order as the primary machine to create the necessary links for the server. Next you can attempt to create the stores database (or an empty test database if you prefer). Create the database on the path of the global device which you plan to use for storage of the actual production database. Ensure that the create database command completes successfully. Proceed to remove the test database via the drop database command. The instance is now ready to be made HA!

Sun Cluster 3.0 IBMids Agent

The IBMids agent is a software product consisting of the methods required to start, stop, monitor IBM IDS in a SC3.0 environment. These methods are implemented as shell scripts, which allows for maximum flexibility in the hands of a skilled user. To allow for ease of installation on each node of the cluster, move the IBMids.tar file into /global/informix, in preparing to install the IBMids agent. After extracting the tar file there will be a directory named agent in /global/informix. On each machine change directory to /global/informix/agent and run "pkgadd -d . IBMids". After the pkgadd command completes the following directory structure should now appear beneath the `/opt/IBMids/` directory on all nodes.

```
bin/
etc/
man/
README.ids
util/
```

If it does not, ensure that the IBMids agent has been installed correctly.

Cluster Topology

The IBMids agent fully supports all three, cluster topology classes supported by SC3.0. These are as follows:

- Clustered Pairs / Idle Standby
- N+1 (or STAR)
- Pair+M

We will discuss examples in the use of each cluster topology class later in this paper. Note however, that the nomenclature is somewhat fluid (that is, there are a variety of names used interchangeably in the HA industry, we will try to conform to those just stated).

Logical Hostname/IP Failover

A logical hostname together with the IP address it maps to, must be associated with a particular IBMids instance. Client programs will access the IBM IDS database instance using this logical hostname instead of the physical hostname of a server in the cluster. This logical hostname is the entry point to the cluster and it shields the client program from addressing the physical servers directly. That is, this logical hostname/IP address is what is configured on the clients (via the setnet32 command or the ODBC DM). This logical hostname is configured as a logical hostname resource. This logical hostname resource must be added to the same resource group as the instance resource. In the case of a failure, the entire resource group including the instance and the logical hostname will be failed over to the backup server. This floating IP setup provides high availability IBM IDS service to client programs. Ensure that this hostname maps to an IP address, and that this name-to-IP address mapping is configured on all nodes in the cluster. More information on configuration for public IP addresses can be found in Sun Cluster 3.0 U1 Installation Guide. Following is an `/etc/host` file with the logical hostname entry for the example systems.

Example: `/etc/hosts` with logical hostname entry for suninfo:

```
#
# Internet host table
#
127.0.0.1         localhost
192.168.0.201     pelican
192.168.0.202     suninfo     loghost
192.168.0.203     dbserver
```

and backup system pelican:

```
#
# Internet host table
#
127.0.0.1         localhost
192.168.0.201     pelican     loghost
192.168.0.202     suninfo
192.168.0.203     dbserver
```

Logical Hostname Considerations

In the case of IBM IDS, the highly available logical hostname/IP must be collocated (that is, located within the same SC3.0 resource group) as the instance itself. This will guarantee that the logical hostname/IP address will always be local to the IBM IDS instance. Otherwise, should they not exist within the same SC3.0 resource group, the possibility exists that a particular physical node may host the logical hostname/IP address, while a different physical node may host the IBM IDS instance itself. Note that you may choose to assign no highly available hostname/IP address for a particular IBM IDS instance. This may be done in the case where there are no TCP/IP clients.

Agent utilities

There are three utilities (startids, stopids, removeids) supplied, located in `/opt/IBMids/util`, that are used to control the installation/start, stop and removal of a basic IBMids agent in a Sun Cluster 3.0 environment. These scripts are designed to help with the initial setup of an IBMids agent but are limited in their capabilities. Note that while there exist a number of other components to the package, it is only these three that are designed to be called directly by the individual user. A man style document is provided in the man directory for each of the three supplied methods in `/opt/IBMids`. **Note:** The `/opt/IBMids/man` path should be added to the MANPATH environment variable.

Startids utility

This method will register and selectively start the appropriate resources and resource groups for a specified instance. It will normally attempt to online the resource group unless the -x option is used, which is recommended for an initial installation. Commonly, this will be the first script called, as it will perform all necessary steps to prepare IBM IDS for SC3.0 control including prompting for the IBM IDS specific information and validating the user's responses before configuring the agent. To create an IBMids resource type for the example systems the following startids command can be used:

```
cd /opt/IBMids/util
./startids -h dbserver -p 1526/tcp -n nafo0@suninfo,nafo0@pelican -x
```

Alternatively the following SC commands could be run from the command line:

```
# Register resource type <IBM.ids>
scrgadm -a -t IBM.ids

# Creating failover resource group <ids-harg>
scrgadm -a -g ids-harg -h suninfo,pelican

# Creating logical host resource <dbserver>
scrgadm -a -L -g ids-harg -l dbserver -n nafo0@suninfo,nafo0@pelican

# Creating resource <ids-hars> for the resource type <IBM.ids>
scrgadm -a -j ids-hars -g ids-harg -t IBM.ids \
      -y scalable=false \
      -y Port_list=1526/tcp \
      -y Network_resources_used=dbserver \
      -x Confdir_list="" \
      -x START_TIMEOUT=300 \
      -x STOP_TIMEOUT=300 \
      -x PROBE_TIMEOUT=30 \
      -x INFORMIXDIR=/global/informix/iif921fc5 \
      -x INFORMIXSERVER=cluster_shm \
      -x ONCONFIG=onconfig.cluster \
      -x INFORMIXSQLHOSTS=/global/informix/iif921fc5/etc/sqlhosts

# Bring the resource group <ids-harg> online
scswitch -Z -g ids-harg
```

Stopids utility

This method will execute required SC3.0 commands in order to bring a Highly Available IBM IDS instance offline. It will not remove any resources or resource groups from the cluster. To stop the instance for the example systems the following stopids command can be used:

```
/opt/IBMids/util/stopids -h dbserver
```

Alternatively the following SC commands could be run from the command line:

```
#Offline resource group <ids-harg> on all nodes ...
scswitch -z -g ids-harg -h ""

#Disable the resource <ids-hars> ...
scswitch -n -j ids-hars

#Disable the network resource <dbserver> ...
scswitch -n -j dbserver
```

Removeids utility

This method will execute the required commands in order to remove the IBMids agent from the SC3.0 cluster. It will remove the resources and groups created by the startids command. Essentially, this method is the inverse of startids, and will be generally called if the database instance is no longer required to be HA. To remove the IBMids resource type, resource groups and resources for the example systems the following removeids command can be used:

```
/opt/IBMids/util/removeids -h dbserver
```

Alternatively the following SC commands could be run from the command line:

```
#Offline the resource group <ids-harg> ...
scswitch -z -g ids-harg -h ""

#Disable the resource <ids-hars> ...
scswitch -n -j ids-hars

#Remove the resource <ids-hars> ...
scrgadm -r -j ids-hars

#Remove the resource type <IBM.ids> ...
scrgadm -r -t IBM.ids

#Disable the network resource <dbserver> ...
scswitch -n -j dbserver

#Remove the resource <dbserver> ...
scrgadm -r -j dbserver

#Unmanage the resource group <ids-harg> ...
scswitch -u -g ids-harg

#Remove the resource group <ids-harg> ...
scrgadm -r -g ids-harg
```

Agent Setup Example

We will create a highly available IBM IDS instance in the following example. This is how our example cluster appears in diagrammatic form.

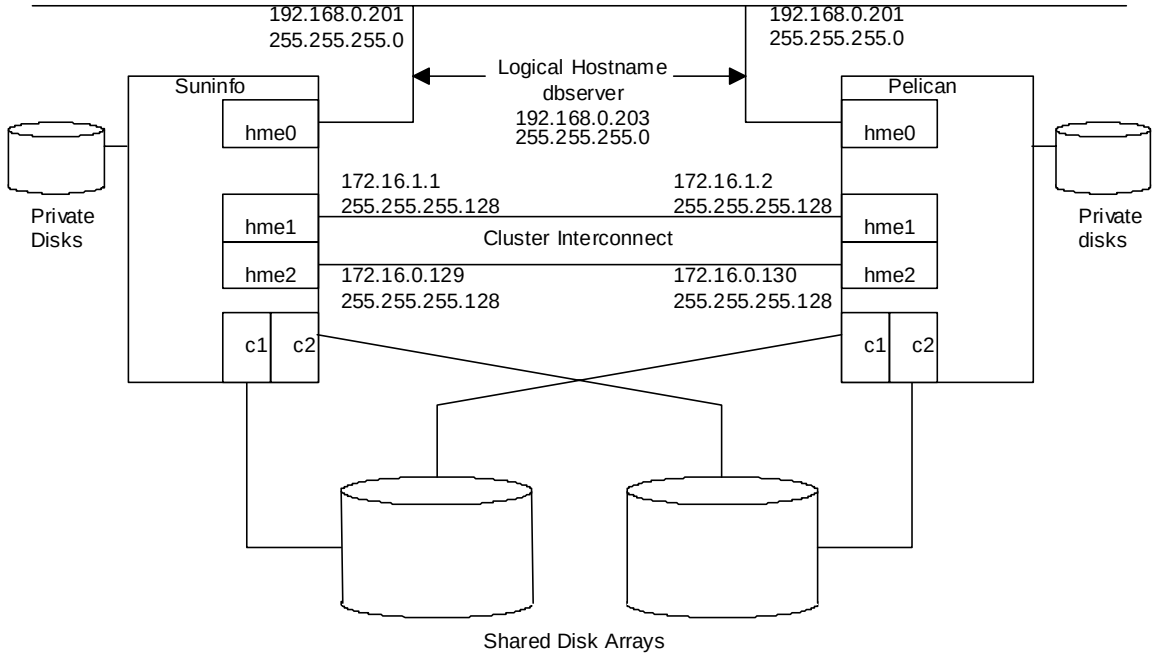


Figure 1

In terms of the existing SC3.0 infrastructure, we'll use the scstat, scrgadm and scdidadm commands to display the information for the given diagrammed cluster: (blank lines removed to save space)

```
# scstat -p
------------------------------------------------------------------
-- Cluster Nodes --
                    Node name           Status
                    ---------           ------
  Cluster node:     suninfo             Online
  Cluster node:     pelican             Online
------------------------------------------------------------------
-- Cluster Transport Paths --
                    Endpoint            Endpoint            Status
                    --------            --------            ------
  Transport path:   suninfo:hme2        pelican:hme2        Path online
  Transport path:   suninfo:hme1        pelican:hme1        Path online
------------------------------------------------------------------
-- Quorum Summary --
  Quorum votes possible:      3
  Quorum votes needed:        2
  Quorum votes present:       3
-- Quorum Votes by Node --
```

```
                    Node Name           Present Possible Status
                    ---------           ------- -------- ------
  Node votes:       suninfo             1        1       Online
  Node votes:       pelican             1        1       Online
-- Quorum Votes by Device --
                    Device Name         Present Possible Status
                    -----------         ------- -------- ------
  Device votes:     /dev/did/rdsk/d4s2  1        1       Online
------------------------------------------------------------------
-- Device Group Servers --
                         Device Group        Primary             Secondary
                         ------------        -------             ---------
  Device group servers:  ids-ds1             suninfo             pelican
  Device group servers:  dbspace-ds1         suninfo             pelican
  Device group servers:  sbspace-ds1         suninfo             pelican
-- Device Group Status --
                              Device Group        Status
                              ------------        ------
  Device group status:        ids-ds1             Online
  Device group status:        dbspace-ds1         Online
  Device group status:        sbspace-ds1         Online
------------------------------------------------------------------
------------------------------------------------------------------
# scrgadm -p
Res Type name:                 SUNW.LogicalHostname
  Res Type description:        Logical Hostname Resource Type
Res Type name:                 SUNW.SharedAddress
  Res Type description:        HA Shared Address Resource Type
# scdidadm -L
1        suninfo:/dev/rdsk/c0t0d0       /dev/did/rdsk/d1
2        suninfo:/dev/rdsk/c0t1d0       /dev/did/rdsk/d2
3        suninfo:/dev/rdsk/c0t6d0       /dev/did/rdsk/d3
4        pelican:/dev/rdsk/c1t3d0       /dev/did/rdsk/d4
4        suninfo:/dev/rdsk/c1t3d0       /dev/did/rdsk/d4
5        pelican:/dev/rdsk/c1t4d0       /dev/did/rdsk/d5
5        suninfo:/dev/rdsk/c1t4d0       /dev/did/rdsk/d5
6        pelican:/dev/rdsk/c1t5d0       /dev/did/rdsk/d6
6        suninfo:/dev/rdsk/c1t5d0       /dev/did/rdsk/d6
7        pelican:/dev/rdsk/c2t3d0       /dev/did/rdsk/d7
7        suninfo:/dev/rdsk/c2t3d0       /dev/did/rdsk/d7
8        pelican:/dev/rdsk/c2t4d0       /dev/did/rdsk/d8
8        suninfo:/dev/rdsk/c2t4d0       /dev/did/rdsk/d8
9        pelican:/dev/rdsk/c2t5d0       /dev/did/rdsk/d9
9        suninfo:/dev/rdsk/c2t5d0       /dev/did/rdsk/d9
10       pelican:/dev/rdsk/c0t0d0       /dev/did/rdsk/d10
11       pelican:/dev/rdsk/c0t1d0       /dev/did/rdsk/d11
12       pelican:/dev/rdsk/c0t6d0       /dev/did/rdsk/d12
```

You will notice logical disks 1-3 and 10-12 look like the same devices but have different logical device names. These are local devices and not part of the shared arrays.

Example

In this example we will create a simple highly available IBM IDS instance along with an associated logical hostname/IP address. Our IBM IDS instance name (DBSERVERNAME) is `dbserver_tcp` with `dbserver_shm` as a DBSERVERALIAS and the logical hostname in /etc/hosts is `dbserver`. In the following examples the use of the -v option is to show the actual scrgadm commands that are run by the utilities and the occasional use of "`|grep .`" is just to remove blank lines to shorten output.

First, use the startids utility to register the instance with the Sun Cluster 3.0 infrastructure (we will use the convention of boldface for input entered by the user):

```
suninfo # cd /opt/IBMids/util
./startids -h dbserver -p 1526/tcp -n nafo0@suninfo,nafo0@pelican -x -v
Registering resource type <IBM.ids>...
scrgadm -a -t IBM.ids
done.
Creating failover resource group <ids-harg>...
scrgadm -a -g ids-harg
done.
Creating logical host resource <dbserver>...
scrgadm -a -L -g ids-harg -l dbserver -n nafo0@suninfo,nafo0@pelican
done.

Enter the directory path for the Informix software:
/global/informix/ids930uc1

Select your onconfig file from the list below.
1) onconfig.930
2) onconfig.demo
3) onconfig.std
4) Other
Enter the number for your onconfig file: 1

Found file "sqlhosts" in "/global/informix/ids930uc1/etc".
Do you want to use this file for communications info (y/n)? y

INFORMIXSERVER is set to "dbserver_tcp" in "onconfig.930".
Is this the servername that you want to use(y/n)? y

The default for START_TIMEOUT is 300 seconds.
Do you want to use this value(y/n)? y

The default for STOP_TIMEOUT is 300 seconds.
Do you want to use this value(y/n)? y

The default for PROBE_TIMEOUT is 30 seconds.
Do you want to use this value(y/n)? y

The Following values will be used to install the Informix agent.
INFORMIXDIR=/global/informix/ids930uc1
INFORMIXSERVER=dbserver_tcp
ONCONFIG=onconfig.930
INFORMIXSQLHOSTS=/global/informix/ids930uc1/etc/sqlhosts
```

```
START_TIMEOUT=300
STOP_TIMEOUT=300
PROBE_TIMEOUT=30
Are these values what you want to use(y/n)? y
Creating resource <ids-hars> for the resource type <IBM.ids>...
scrgadm -a -j ids-hars -g ids-harg -t IBM.ids \
     -y Port_list=1526/tcp \
     -y Network_resources_used=dbserver \
     -x Confdir_list="" \
     -x START_TIMEOUT=300 \
     -x STOP_TIMEOUT=300 \
     -x PROBE_TIMEOUT=30 \
     -x TRACE_AGENT=false \
     -x INFORMIXDIR=/global/informix/ids930uc1 \
     -x INFORMIXSERVER=dbserver_tcp \
     -x ONCONFIG=onconfig.930 \
     -x INFORMIXSQLHOSTS=/global/informix/ids930uc1/etc/sqlhosts
done.
Not starting service.  To start, enter the following command.
scswitch -Z -g ids-harg
```

Use the scstat command to determine the status of the cluster.

```
suninfo # scstat -p|grep .
-------------------------------------------------------------------
-- Cluster Nodes --
                    Node name           Status
                    ---------           ------
  Cluster node:     suninfo             Online
  Cluster node:     pelican             Online
-------------------------------------------------------------------
-- Cluster Transport Paths --
                    Endpoint            Endpoint            Status
                    --------            --------            ------
  Transport path:   suninfo:hme2        pelican:hme2        Path online
  Transport path:   suninfo:hme1        pelican:hme1        Path online
-------------------------------------------------------------------
-- Quorum Summary --
  Quorum votes possible:      3
  Quorum votes needed:        2
  Quorum votes present:       3
-- Quorum Votes by Node --
                    Node Name           Present Possible Status
                    ---------           ------- -------- ------
  Node votes:       suninfo             1        1       Online
  Node votes:       pelican             1        1       Online
-- Quorum Votes by Device --
                    Device Name         Present Possible Status
                    -----------         ------- -------- ------
  Device votes:     /dev/did/rdsk/d4s2  1        1       Online
-------------------------------------------------------------------
-- Device Group Servers --
                         Device Group        Primary             Secondary
                         ------------        -------             ---------
```

```
  Device group servers:  ids-ds1             suninfo             pelican
  Device group servers:  dbspace-ds1         suninfo             pelican
  Device group servers:  sbspace-ds1         suninfo             pelican
-- Device Group Status --
                              Device Group        Status
                              ------------        ------
  Device group status:        ids-ds1             Online
  Device group status:        dbspace-ds1         Online
  Device group status:        sbspace-ds1         Online
--------------------------------------------------------------------
-- Resource Groups and Resources --
            Group Name          Resources
            ----------          ---------
 Resources: ids-harg            dbserver ids-hars
-- Resource Groups --
            Group Name          Node Name           State
            ----------          ---------           -----
     Group: ids-harg            suninfo             Offline
     Group: ids-harg            pelican             Offline
-- Resources --
            Resource Name       Node Name           State     Status Message
            -------------       ---------           -----     --------------
  Resource: dbserver            suninfo             Offline   Offline
  Resource: dbserver            pelican             Offline   Offline
  Resource: ids-hars            suninfo             Offline   Offline
  Resource: ids-hars            pelican             Offline   Offline
--------------------------------------------------------------------
```

The result of the startids processing is that the appropriate IBM IDS resources and resource groups are created and registered with SC3.0.

Note the naming convention of the resources and resource groups and their structure. The start, stop and remove utilities have a rigid structure and are provided for reference rather than agent maintenance. While they can be used to create and maintain a simple agent installation, they will probably not be sufficient for most agent setups.

After verifying that the agent registered properly you can start the instance with the `startids` command.

```
suninfo # /opt/IBMids/util/startids -h dbserver -v
Bringing the resource group <ids-harg> online...
scswitch -Z -g ids-harg
done.
```

The result of the online processing is that the ids-harg resource group and its associated resources are online and under the control of Sun Cluster 3.0. You should see the resources brought online for the appropriate node (for example, on the physical hostname suninfo, you should see that it hosts the logical hostname resource dbserver, as well as the ids-hars resource). There are two more supplied utilities which we will now discuss: `stopids` and `removeids`.

At some point you may wish to offline an IBM IDS instance in order to perform some maintenance tasks on the system where you may need to bring the database engine down for an extended period of time. Directly issuing the onmode -uky command will be ineffective, as SC3.0 will interpret the absence of resources caused by the successful completion of the onmode command as a failure and attempt to restart the appropriate database instance resources. Instead, to accomplish this task, you must offline the resources as follows.

```
suninfo # /opt/IBMids/util/stopids -h dbserver -v
Offlining resource group <ids-harg> on all nodes ...
scswitch -z -g ids-harg -h ""
done.
Disabling the resource <ids-hars> ...
scswitch -n -j ids-hars
done.
Disabling the network resource <dbserver> ...
scswitch -n -j dbserver
done.
```

We can verify the command's completion with scstat:

```
 # scstat -p|grep .
------------------------------------------------------------------
-- Cluster Nodes --
                    Node name           Status
                    ---------           ------
  Cluster node:     suninfo             Online
  Cluster node:     pelican             Online
------------------------------------------------------------------
-- Cluster Transport Paths --
                    Endpoint            Endpoint            Status
                    --------            --------            ------
  Transport path:   suninfo:hme2        pelican:hme2        Path online
  Transport path:   suninfo:hme1        pelican:hme1        Path online
------------------------------------------------------------------
-- Quorum Summary --
  Quorum votes possible:      3
  Quorum votes needed:        2
  Quorum votes present:       3
-- Quorum Votes by Node --
                    Node Name           Present Possible Status
                    ---------           ------- -------- ------
  Node votes:       suninfo             1        1       Online
  Node votes:       pelican             1        1       Online
-- Quorum Votes by Device --
                    Device Name         Present Possible Status
                    -----------         ------- -------- ------
  Device votes:     /dev/did/rdsk/d4s2  1        1       Online
------------------------------------------------------------------
-- Device Group Servers --
                         Device Group        Primary             Secondary
                         ------------        -------             ---------
  Device group servers:  ids-ds1             pelican             suninfo
```

```
  Device group servers:  dbspace-ds1         pelican             suninfo
  Device group servers:  sbspace-ds1         pelican             suninfo
-- Device Group Status --
                              Device Group        Status
                              ------------        ------
  Device group status:        ids-ds1             Online
  Device group status:        dbspace-ds1         Online
  Device group status:        sbspace-ds1         Online
------------------------------------------------------------------
-- Resource Groups and Resources --
            Group Name          Resources
            ----------          ---------
 Resources: ids-harg            dbserver ids-hars
-- Resource Groups --
            Group Name          Node Name           State
            ----------          ---------           -----
     Group: ids-harg            suninfo             Offline
     Group: ids-harg            pelican             Offline
-- Resources --
            Resource Name       Node Name           State     Status Message
            -------------       ---------           -----     --------------
  Resource: dbserver            suninfo             Offline   Offline -
LogicalHostname offline.
  Resource: dbserver            pelican             Offline   Offline
  Resource: ids-hars            suninfo             Offline   Offline
  Resource: ids-hars            pelican             Offline   Offline
------------------------------------------------------------------
```

The result of this action, as you can see from the scstat output, is that all resources are now offlined. No resources associated with the instance ids-harg should be active on either node, and process monitoring has stopped. At this point SC3.0 will not take any action to protect this instance should it be brought online manually (that is, via oninit) and a failure were to occur. Note that at this point trying to start the instance will fail since the logical hostname is offlined. A change to the /etc/hosts file would be necessary to start the database instance. Alternately the following command could be used to take the database resource into an unmanaged state:

```
scswitch -n -M -j ids-hars
```

Note: Care must be taken for any changes made in this state, for example changes to /etc/hosts that may affect the logical host resource while it is still enabled. And any changes that are made to the database instance are reflected in the appropriate extension properties.

Let us assume that it is decided to remove this instance permanently from SC3.0 monitoring and control. For this task, you may use the removeids method. Note that this method merely interfaces with SC3.0 in order to perform these tasks, the instance itself is not dropped nor removed.

```
suninfo # /opt/IBMids/util/removeids -h dbserver -v
Offlining the resource group <ids-harg> ...
scswitch -z -g ids-harg -h ""
done.
Disabling the resource <ids-hars> ...
scswitch -n -j ids-hars
done.
```

```
Removing the resource <ids-hars> ...
scrgadm -r -j ids-hars
done.
Removing the resource type <IBM.ids> ...
scrgadm -r -t IBM.ids
done.
Disabling the network resource <dbserver> ...
```

```
scswitch -n -j dbserver
done.
Removing the resource <dbserver> ...
scrgadm -r -j dbserver
done.
Unmanaging the resource group <ids-harg> ...
scswitch -u -g ids-harg
done.
Removing the resource group <ids-harg> ...
scrgadm -r -g ids-harg
done.
```

Should there be any difficulty with the registration, unregistration, onlining or offlining process, you may need to consult the system error log (typically in `/var/adm/messages`).

Another option for debugging (for advanced users), is to call the startdb, stopdb and probedb methods directly and observe their behavior. To do this type of testing the logical hostname must be onlined but not the rest of the resource group.

As stated before these three utilities are provided more for reference than for cluster manipulation. The use of `the scrgadm and scswitch` commands provide for greater flexibility in configuring the agent. For instance multiple IBM IDS instances could be started in one resource group, or multiple resource groups with multiple logical hostnames being created for each instance. The layout depends on the requirements of the application driving the server.

Customizing SC3.0 resource properties for IBM IDS

The IBMids agent is registered with SC3.0 as a defined resource type. Each IBM IDS instance is registered with SC3.0 as a resource. For resource properties provided by SC3.0, you are able change specific attributes via the Sun Cluster administrative commands.

The resource type properties and the resource properties are declared in the Resource Type Registration (RTR) file. The resource property declarations follow the resource type declarations in the RTR file. Entries begin with an open curly bracket and end with a closed curly bracket.

You can change resource properties by registering the resource with a modified RTR file`/opt/IBMids/etc/IBM.ids`, but the preferred method is through the SC3.0 administrative command `scrgadm` which can be used to add a new or modify an existing configuration.

Start_timeout, stop_timeout and probe_timeout extension properties

The <method>_timeout properties set the value in seconds after which the SC3.0 RGM concludes invocation of the method has failed. The MIN value for the <method>_timeout properties is set to 30 seconds. This prevents administrators from setting shorter time-outs, which do not improve switchover/failover performance, and can lead to undesired RGM actions (false fail overs, node reboot, or setting the resource to a `STOP_FAILED` state, requiring operator intervention).

Setting too-short method time outs leads to a decrease in overall availability of the IBM IDS data service.

Monitor_retry_count and monitor_retry_interval extension properties

The number of retries (`retry_count`) to be done within a `retry_interval` before the SC3.0 RGM concludes that the resource cannot be successfully started on the current node. The `retry_interval` is minutes as opposed to seconds for other time variable.

IBM IDS environment variable extension properties

The IBM IDS instance specific environment variables INFORMIXDIR, INFORMIXSERVER, INFORMIXSQLHOSTS, ONCONFIG, DB_LANG, SERVER_LOCALE, CLIENT_LOCALE along with LD_LIBRARY_PATH are all stored as extension properties for each resource associated with an IBM IDS instance. The INFORMIXDIR, INFORMIXSERVER, INFORMIXSQLHOSTS and ONCONFIG extension properties are mandatory for all instances of the IBM IDS agent, while the others are provided for instance specific needs. Each of these properties can be changed when the agent instance is disabled, and care must be taken when changing an IBM IDS instance that these properties are also changed accordingly.

The property values can be added (-a) or changed (-c) using the SC3.0 `scrgadm` command.

For a complete description on customizing the resource properties, please refer to the Sun Cluster 3.0 U1 Data Services Installation and Configuration Guide.

Cluster Verification Testing

An important aspect of a highly available cluster is testing. The purpose of this testing is to gain some confidence that the highly available cluster will function as envisioned for various failure scenarios. What follows is a set of scenarios that are recommended at a minimum for cluster testing and verification. Note that to ensure that the cluster continues to function as expected, it is recommended that time be taken to ensure these tests are run on whenever a configuration change is made to the cluster instance.

Test 1

In this test, we would like to perform Sun Cluster 3.0 management commands, to ensure that the ids-harg instance can be controlled correctly. First, verify that the instance is accessible from the clients (or locally), and that various database commands complete successfully (for example, create database). Take the ids-hars resource offline, and take the SC3.0 resource offline via the following:

```
scswitch -n -j ids-hars
scswitch -n -j dbserver
```

Observe that the ids-hars resource appear offline on all nodes in the cluster, and the logical hostname dbserver is likewise. From the perspective of a client of this instance, the existing connections are closed.

Test 2

To return the resources to their previous states, bring them online with the following SC3.0 commands in the reverse order that they were offlined:

```
scswitch -e -j dbserver
scswitch -e -j ids-hars
```

IBM IDS clients can now reconnect to the server.

Test 3

Test the failover of the IBM IDS instance and associated resources from suninfo to pelican. At this point, the cluster is again at its initial state. With the following command you can move the resources contained within the resource group ids-harg over to the backup node:

```
scswitch -z -g ids-harg -h pelican
```

When complete the relevant resources will be online on the backup node. You should now see the ids-hars and dbserver resources along with the ids-harg resource group are now online on the backup machine, that is, pelican. Verify that this is the case by executing the scstat -p command.

Test 4

Here, test the failover capabilities of the Sun Cluster 3.0 software itself. Bring the resources back into their initial state (that is, have the ids-harg resource group hosted on suninfo). this can be accomplished via:

```
scswitch -z -g ids-harg -h suninfo
```

Once the instance and its associated resources are hosted on the suninfo machine, perform a power off operation on that physical node. This will cause the internal heartbeat mechanism to detect a physical node failure, and the ids-harg resource group will be restarted on the surviving node, pelican.

Verify that the results are identical to those seen in Test 3 (that is, the ids-harg resource group should be hosted on pelican and the clients should behave similarly in both cases).

Test 5

Bring the cluster back into its initial state. In this test, verify that the software monitoring is working as expected. To perform this test, you may issue commands as follows:

```
ps -ef | grep oninit
```

Get the PID of process group leader, which will be the oninit process whose PPID is 1.

```
kill -11 <PID>
```

This may take a few moments as this will cause to server to assert fail. The Sun Cluster 3.0 monitor should detect that a required process is not running and attempt to restart the instance on the same node. Verify that this in fact does occur. The client connections should experience a brief delay in service while the restart process continues.

Note that there are a large number of distinct testing scenarios that can be executed, limited only by your resources and imagination. The above set are meant as a minimum that should be run to test the correct functioning of the cluster.

Discussion of Cluster Topology

The IBMids agent fully supports all three cluster topology classes supported by Sun

Cluster 3.0. These are as follows:

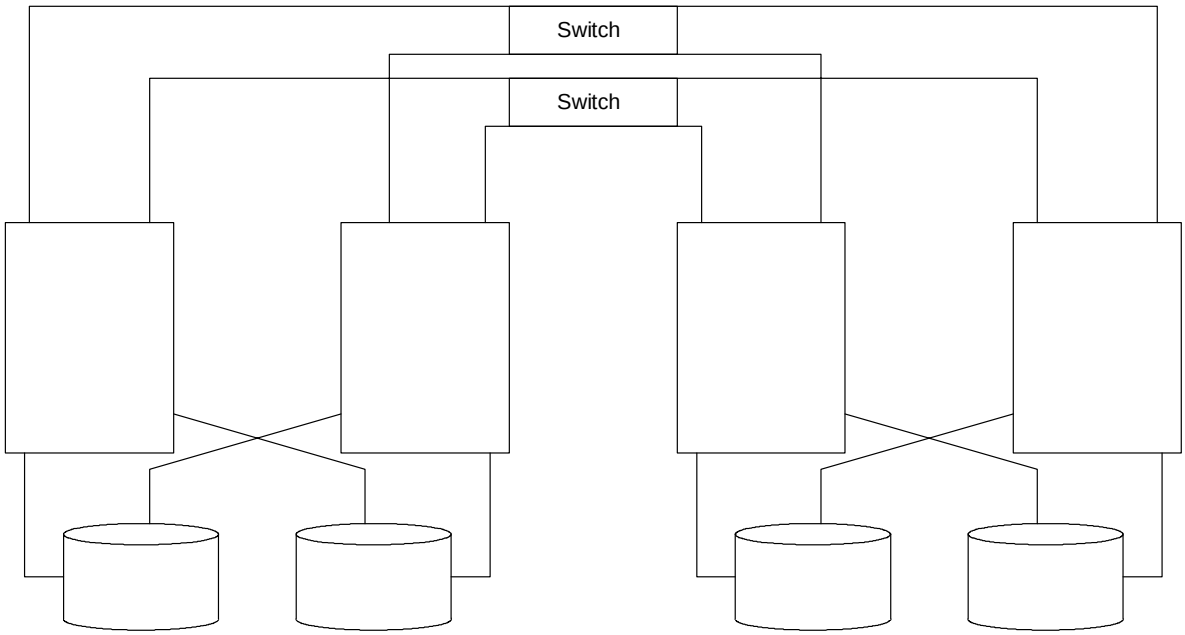
- Clustered Pairs / Idle Standby
- N+1 (or STAR)
- Pair+M

Idle Standby

Idle standby is the simplest highly available cluster topology, as seen previously in **Figure 1**. In this scenario, the primary machine is hosting the production database instance and associated resources. A second machine is idle, and available to host the production database instance and associated resources should a failure occur on the primary machine. Note that the second machine can be idle, or it can be running a workload (perhaps another IBM IDS instance) in order to maximize use of resources.

Clustered Pairs

In the mutual take over case, you envision a cluster of N nodes as N/2 pairs of nodes. Node number n is responsible for failover support of node number n+1 (and vice versa), node number n+2 is responsible for failover support of node number n+3 (and vice versa), and so on until you reach the Nth node. Note that this scenario requires that N be an even number. The advantages of this configuration are that in the normal (that is, non-failure case), all machines are hosting database resources and are performing productive work. The primary disadvantage is that in the failure case (that is, during the time period after one of the hardware resources has failed but before this hardware resource has been repaired), there is one node that is required to support on average twice the workload of any other physical node. However, should failures be relatively rare, and their duration short, this is a reasonable scenario.

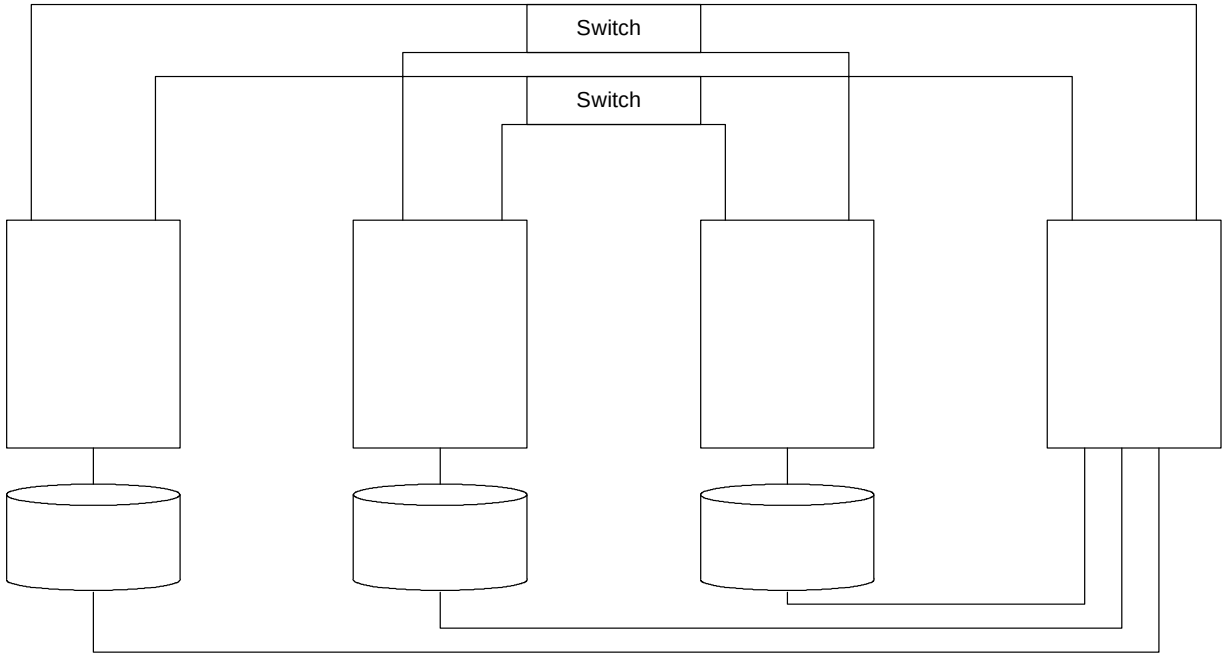


Clustered Pairs (Mutual Takeover)

Figure 2

N+1 Takeover

This case relies on an N node cluster, with one defined node as the standby for all N nodes. The advantage of this scenario is that there is no performance degradation during the failure time (that is, during the time period after one of the hardware resources has failed but before this hardware resource has been repaired). The primary disadvantage is that approximately 1/(N+1) of the aggregate physical computing resource lies unused during the normal operation (and presumably overwhelmingly likely and common) case.

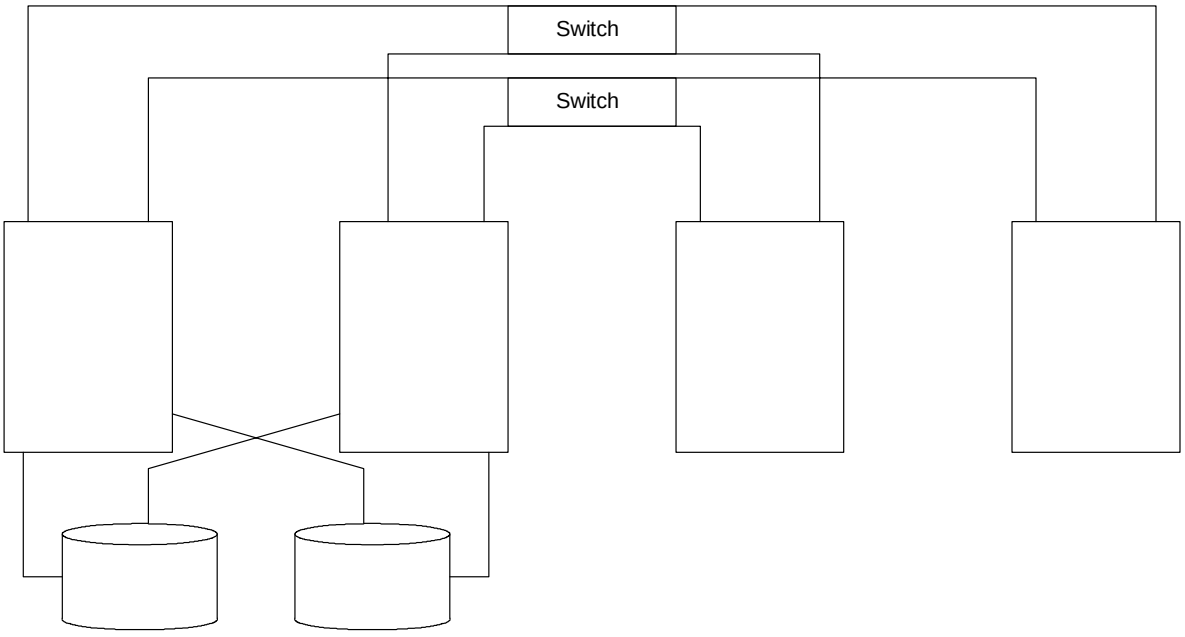


N + 1 (Star)

Figure 3

Pair + M (N + M)

This case relies on an N node cluster, with M defined nodes as the hot standby for each of the N nodes. The prime advantage of this configuration is that the environment is fully redundant, in the sense that up to N - 1 node failures can be tolerated while still maintaining full database access (subject of course to increased query response times due to capacity constraints when there are less than N nodes in the cluster). In this way, IBM IDS used in conjunction with SC3.0 ensures full database software redundancy and is most appropriate for environments requiring the highest degree of availability. The primary disadvantage of this configuration is potentially non-local accesses to data on the global device; this will be discussed later in the paper.



Pair + M

Figure 4

Optimizing Cluster File System (CFS) Performance

Whenever possible, collocate the IBM IDS instance with the corresponding data and use locally hosted global file systems. If the a global device is dual-hosted in the cluster, let the same node primary the resource group and its corresponding device group, therefore the active I/O path will be local access. During a failover, the backup node (secondary) will be promoted to the status of a primary for resource and device group which have been switched over.

The IBM IDS instance is strongly recommended to be collocated with its corresponding data in order to achieve the best performance in disk intensive scenarios.

Note that this collocation is not significant for the instance home directory since it tends to be subjected to an extremely small I/O bandwidth by the database server.

As a general guideline, Pair+M topology should be avoid, because the additional M nodes will likely be configured such that access to data will be remote through one of the global I/O paths provided by the primary of the dual-hosted device. The disk I/O performance will be impacted by the overhead associated with acquiring the date remotely.

Sun HAStorage data service is an agent that can be used in your configuration to control the primary for a resource group device. It makes sure that when a resource group is moved to another node that the associated device groups are switched to primary on that node. For more information on HAStorage and configuration please refer to the Synchronizing the Startups Between Resource Groups and Disk Device Groups. To add this agent to our example systems you would do the following:

```
scrgadm -a -t SUNW.HAStorage
scrgadm -a -j ids-hastorage -g ids-harg -t SUNW.HAStorage -x \
      ServicePath=ids-ds1,dbspace-ds1,sbspace-ds1 -x AffinityOn=True
scswitch -e -j hastorage-1
scrgadm -c -j ids-hars -y Resource_Dependencies=ids-hastorage
```

Now whenever a failover/switchover occurs the device groups will follow the resource group to a new primary machine.

Recommended CFS mount options

The required mount options for cluster file systems is as follows:

```
global,logging
```

If a files system is being used solely for database storage and does not contain any binary files that will be executed the following option should also be used:

```
forcedirectio
```

Typical entries in the `/etc/vfstab` file would be as follows for the Informix binaries filesystem:

```
/dev/vx/dsk/scdg2/vol05 /dev/vx/rdsk/scdg2/vol05 /global/scdg2/scdg2 ufs
2 yes global,logging
```

and for a database storage file system:

```
/dev/vx/dsk/scdg2/vol05 /dev/vx/rdsk/scdg2/vol05 /global/scdg2/scdg2 ufs
2 yes global,logging,forcedirectio
```

The global mount option simply identifies this mount point as a Cluster File System (CFS). The logging option is always used implicitly in use for a CFS file system, regardless of whether it is directly specified on the mount line, however, for clarity, it is recommended that the option be explicitly stated. The forcedirectio option only applies to CFS, and essentially allows file system I/O to bypass Solaris kernel page cache. This option is especially beneficial for IBM IDS file system I/O, as IDs performs all required I/O caching and derives no further benefit from additional caching in the Solaris kernel. The reduction in CPU path length can be significant with the use of direct I/O, and its use is encouraged.

There is one additional CFS mount option which merits mention. The syncdir option only applies to global UFS mounts. Whether it is utilized depends on the relative importance placed on error semantics versus absolute performance. Briefly, the syncdir option is relevant in the case when a file is in the process of being extended. If it is set, notification of a file system full will be returned even if a failover of the primary occurs during the write call itself. This is ideal error semantics behavior, however, this must be weighed against the performance degradation, which can be significant in the case of writes which occur on the non-primary node.

The recommendation is that, should all I/O occur only local to the primary host, that this option be enabled. Otherwise, due to the performance penalty and the exceedingly small probability of a file system full occurring during a diskgroup failover, it is recommended that this option not be enabled.

Finally, for customers desiring the maximum performance possible with a global available device, the use of raw partitions is recommended.

Failover Time and How to Minimize It

To reduce the service interruption time, it is important to understand the discrete components which make up failover, and those are:

- time for SC3.0 to detect and respond to failure
- time for IBM IDS to recover from failure
- time for TCP/IP clients to detect hostname switch over and reconnect

In particular, you can influence (and reduce) the amount of time required for IBM IDS failure recovery with the following methods. An example of setting this parameter to allow this time window to be no longer than, for example, ten seconds follows.

```
ndd -set /dev/tcp tcp_ip_abort_cinterval 10000
```

Also, you may reduce the time required for IBM IDS database restart recovery. You may employ the standard techniques, which are discussed in the Administrators Guide for Informix Dynamic Server and the Performance Guide for Informix Dynamic Server.

Client Issues

Applications which are relying on a highly available IBM IDS instance must be able to reconnect in the event of a fail over. Since the hostname and IP address of a logical host remain the same, there is no need to connect to a different hostname or re-catalog the database.

Consider a cluster with two machines and one IBM IDS instance. The IBM IDS instance will normally reside on one of the machines in the cluster. Clients of the HA instance will connect to the logical IP address (or hostname) of the logical host associated with the HA instance.

According to an HA client, there are two types of fail overs. One type occurs if the machine which is hosting the HA instance crashes. The other type is when the HA instance is given an opportunity to shut down gracefully.

If a machine crashes and takes down the HA instance, both existing connections and new connections to the database will hang. The connections hang because there are no machines on the network that have the IP address which the clients were using for the database. If the database is shutdown gracefully, a "`onmode -uky`" forces off the connections to the database and then shuts down. In this case, existing connections to the database will receive an error because they were forced off.

During the fail over, the logical IP address associated with the database is off-line, either because the SC3.0 software took it off-line or because the machine which was hosting the logical host crashed. At this point any new connections to the database will hang for a short period of time.

The logical IP address associated with the database is eventually brought up on another machine before IBM IDS is started. At this stage, a connection to the database will not hang, but will receive a communication error because IBM IDS has not yet been started

on the system. Clients which were still connected to the database will also begin receiving communication errors at this stage. Although the clients still believe they are connected, the new machine just started hosting the logical IP address and has no knowledge of any existing connections. The connections are simply reset and the client receives a communication error. After a short time, IBM IDS will be started on the machine and a connection to the database will succeed. At this point, the database may be inconsistent and the clients may have to wait for it to be recovered.

Even though this sounds like a complicated and time consuming set of events, it is actually quite simple. When designing an application for an HA environment, it is not necessary to write special code for the stages where the database connections hang. The connections only hang for a short period of time while the Sun Cluster software moves the logical IP address. Any data service running on Sun Cluster will experience the same hanging connections during this stage. No matter how the database comes down, the clients will receive an error and will simply need to try to reconnect until successful.

From the client's perspective, it is as if the HA instance went down and was then brought back up on the same machine. In a controlled fail over, it would appear to the client that it was forced off and that it could later reconnect to the database on the same machine. In an uncontrolled fail over, it would appear to the client that the database server crashed and was soon brought back up on the same machine.

The Use of Keep-alives

On the server side, using TCP keep-alives protects the server from wasting system resources for a down (or network-partitioned) client. If those resources are not cleaned up (in a server that stays up long enough), eventually the wasted resources grow without bound as clients crash and reboot.

On the client side, using TCP keep-alives enables the client to be notified when a network address resource has failed over or switched over from one physical host to another. That transfer of the network address resource breaks the TCP connection.

However, unless the client has enabled the keep-alive, it would not necessarily learn of the connection break if the connection happens to be quiescent at the time.

In Solaris, the tcp_keepalive feature institutes a series of probe messages from the host to the client. The host sends a series of messages at specified intervals to the client. If no response returns, the host closes the channel. The default interval is 7.2 million milliseconds (2 hours), as well as in many other implementations. You can use `ndd(1M)` to change the value of `tcp_keepalive_interval`. In order for its value to be set every time when system boots, you can add following entry in `/etc/init.d/inetinit` script. The entry for setting `tcp_keepalive_interval` for one hour is as below.

```
/usr/sbin/ndd -set /dev/tcp tcp_keepalive_interval 3600000
```

It is recommended that the value of `tcp_keepalive_interval` should not be set below 15 minutes. In Windows NT/2000 environment, the loss of client/server connection is also determined by using keep-alive probes. The following registry entries

control keep alive probe packet parameters on computers running Windows environment. Changing registry parameters affects all TCP/IP stream connections on the system.

```
KeepAliveInterval
KeepAliveTime
TcpMaxDataRetransmissions
```

The above entries can be found in following registry location:

```
\HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\TCPIP\Parameters
```

Client Retry

From the client's perspective, a failover appears to be a crash of the logical host followed by a fast reboot. In a controlled fail over, it would appear to the client that it was forced off and that it could later reconnect to the database on the same machine. In an uncontrolled fail over, it would appear to the client that the database server crashed and was soon brought back up on the same machine. During a failover, client applications is recommended to check the connection status (time out or communication errors as discussed earlier in this section), and to retry with new connections. Applications can also be implemented to notify the user that a long retry is in progress and enable the user to choose whether to continue.

Summary

We have discussed the design, implementation and verification of a highly available IBM IDS cluster in a SC3.0 environment. This combination produces an outstanding database management system with a high degree of availability, reliability and performance.

Appendix

Example sqlhosts

```
#**************************************************************************
#
#                     INFORMIX SOFTWARE, INC.
#
#                        PROPRIETARY DATA
#
#     THIS DOCUMENT CONTAINS TRADE SECRET DATA WHICH IS THE PROPERTY OF
#     INFORMIX SOFTWARE, INC.  THIS DOCUMENT IS SUBMITTED TO RECIPIENT IN
#     CONFIDENCE.  INFORMATION CONTAINED HEREIN MAY NOT BE USED, COPIED OR
#     DISCLOSED IN WHOLE OR IN PART EXCEPT AS PERMITTED BY WRITTEN AGREEMENT
#     SIGNED BY AN OFFICER OF INFORMIX SOFTWARE, INC.
#
#     THIS MATERIAL IS ALSO COPYRIGHTED AS AN UNPUBLISHED WORK UNDER
#     SECTIONS 104 AND 408 OF TITLE 17 OF THE UNITED STATES CODE.
#     UNAUTHORIZED USE, COPYING OR OTHER REPRODUCTION IS PROHIBITED BY LAW.
#
#
#   Title:  sqlhosts
#   Sccsid: @(#)sqlhosts      9.X   ##/##/##  ##:##:##
#   Description:
#           sqlhosts file for IBM IDS documentation.
#
#**************************************************************************

dbserver_shm      onipcshm    dbserver    no-op
dbserver_tcp      ontlitcp    dbserver    online
```

Example onconfig

```
#*****************************************************************************
#
#                          INFORMIX SOFTWARE, INC.
#
#  Title:       onconfig.std
#  Description: Informix Dynamic Server Configuration Parameters
#
#*****************************************************************************

# Root Dbspace Configuration

ROOTNAME        rootdbs930      # Root dbspace name
ROOTPATH        /global/dbspace/rootdbs930 # Path for device containing root
dbspace
ROOTOFFSET      0               # Offset of root dbspace into device (Kbytes)
ROOTSIZE        30000           # Size of root dbspace (Kbytes)

# Disk Mirroring Configuration Parameters

MIRROR          0               # Mirroring flag (Yes = 1, No = 0)
MIRRORPATH                      # Path for device containing mirrored root
MIRROROFFSET    0               # Offset into mirrored device (Kbytes)

# Physical Log Configuration

PHYSDBS         rootdbs930      # Location (dbspace) of physical log
PHYSFILE        2000            # Physical log file size (Kbytes)

# Logical Log Configuration

LOGFILES        6               # Number of logical log files
LOGSIZE         2000            # Logical log size (Kbytes)

# Diagnostics

MSGPATH         /global/informix/ids930uc1/online.log # System message log
file path
CONSOLE         /dev/console    # System console message path
ALARMPROGRAM    /global/informix/ids930uc1/etc/log_full.sh # Alarm program
path
TBLSPACE_STATS  1               # Maintain tblspace statistics

# System Archive Tape Device

TAPEDEV         /dev/null       # Tape device path
TAPEBLK         16              # Tape block size (Kbytes)
TAPESIZE        10240           # Maximum amount of data to put on tape
(Kbytes)

# Log Archive Tape Device

LTAPEDEV        /dev/null       # Log tape device path
```

```
LTAPEBLK        16              # Log tape block size (Kbytes)
LTAPESIZE       10240           # Max amount of data to put on log tape
(Kbytes)

# Optical

STAGEBLOB                       # Informix Dynamic Server staging area

# System Configuration

SERVERNUM       1               # Unique id corresponding to a OnLine
instance
DBSERVERNAME    dbserver_tcp    # Name of default database server
DBSERVERALIASES dbserver_shm    # List of alternate dbservernames
NETTYPE         tlitcp,1,,NET   # Configure poll thread(s) for nettype
NETTYPE         ipcshm,1,,CPU   # Configure poll thread(s) for nettype
DEADLOCK_TIMEOUT        60      # Max time to wait of lock in distributed
env.
RESIDENT        0               # Forced residency flag (Yes = 1, No = 0)

MULTIPROCESSOR  0               # 0 for single-processor, 1 for multi-
processor
NUMCPUVPS       1               # Number of user (cpu) vps
SINGLE_CPU_VP   0               # If non-zero, limit number of cpu vps to one

NOAGE           0               # Process aging
AFF_SPROC       0               # Affinity start processor
AFF_NPROCS      0               # Affinity number of processors

# Shared Memory Parameters

LOCKS           2000            # Maximum number of locks
BUFFERS         5000            # Maximum number of shared buffers
NUMAIOVPS                       # Number of IO vps
PHYSBUFF        32              # Physical log buffer size (Kbytes)
LOGBUFF         32              # Logical log buffer size (Kbytes)
CLEANERS        1               # Number of buffer cleaner processes
SHMBASE         0x0A000000L     # Shared memory base address
SHMVIRTSIZE     8000            # initial virtual shared memory segment size
SHMADD          8192            # Size of new shared memory segments (Kbytes)
SHMTOTAL        0               # Total shared memory (Kbytes). 0=>unlimited
CKPTINTVL       300             # Check point interval (in sec)
LRUS            8               # Number of LRU queues
LRU_MAX_DIRTY   60              # LRU percent dirty begin cleaning limit
LRU_MIN_DIRTY   50              # LRU percent dirty end cleaning limit
TXTIMEOUT       300             # Transaction timeout (in sec)
STACKSIZE       32              # Stack size (Kbytes)

# System Page Size
# BUFFSIZE - OnLine no longer supports this configuration parameter.
#            To determine the page size used by OnLine on your platform
#            see the last line of output from the command, 'onstat -b'.


# Recovery Variables
# OFF_RECVRY_THREADS:
```

```
# Number of parallel worker threads during fast recovery or an offline
restore.
# ON_RECVRY_THREADS:
# Number of parallel worker threads during an online restore.

OFF_RECVRY_THREADS      10      # Default number of offline worker threads
ON_RECVRY_THREADS       1       # Default number of online worker threads

# Data Replication Variables
DRINTERVAL      30              # DR max time between DR buffer flushes (in
sec)
DRTIMEOUT       30              # DR network timeout (in sec)
DRLOSTFOUND     /global/informix/ids930uc1/etc/dr.lostfound  # DR lost+found
file path

# CDR Variables
CDR_EVALTHREADS 1,2     # evaluator threads (per-cpu-vp,additional)
CDR_DSLOCKWAIT  5       # DS lockwait timeout (seconds)
CDR_QUEUEMEM    4096    # Maximum amount of memory for any CDR queue (Kbytes)
CDR_NIFCOMPRESS 0       # Link level compression (-1 never, 0 none, 9 max)
CDR_SERIAL              0       # Serial Column Sequence
CDR_DBSPACE                     # dbspace for syscdr database
CDR_QHDR_DBSPACE    # CDR queue dbspace (default same as catalog)
CDR_QDATA_SBSPACE   # CDR queue smart blob space
CDR_QDATA_SBFLAGS   0   # Log/no-log (default no log)


# Backup/Restore variables
BAR_ACT_LOG   /global/informix/ids930uc1/bar_act.log  # ON-Bar Log file - not
in /tmp please
BAR_DEBUG_LOG  /global/informix/ids930uc1/bar_dbug.log # ON-Bar Debug Log -
not in /tmp please
BAR_MAX_BACKUP  0
BAR_RETRY       1
BAR_NB_XPORT_COUNT 10
BAR_XFER_BUF_SIZE 31
RESTARTABLE_RESTORE     ON
BAR_PROGRESS_FREQ       0

# Informix Storage Manager variables
ISM_DATA_POOL   ISMData
ISM_LOG_POOL    ISMLogs

# Read Ahead Variables
RA_PAGES                        # Number of pages to attempt to read ahead
RA_THRESHOLD                    # Number of pages left before next group

# DBSPACETEMP:
# OnLine equivalent of DBTEMP for SE. This is the list of dbspaces
# that the OnLine SQL Engine will use to create temp tables etc.
# If specified it must be a colon separated list of dbspaces that exist
# when the OnLine system is brought online.  If not specified, or if
# all dbspaces specified are invalid, various ad hoc queries will create
# temporary files in /tmp instead.
```

```
DBSPACETEMP                     # Default temp dbspaces

# DUMP*:
# The following parameters control the type of diagnostics information which
# is preserved when an unanticipated error condition (assertion failure)
occurs
# during OnLine operations.
# For DUMPSHMEM, DUMPGCORE and DUMPCORE 1 means Yes, 0 means No.

DUMPDIR         /tmp            # Preserve diagnostics in this directory
DUMPSHMEM       1               # Dump a copy of shared memory
DUMPGCORE       0               # Dump a core image using 'gcore'
DUMPCORE        0               # Dump a core image (Warning:this aborts
OnLine)
DUMPCNT         1               # Number of shared memory or gcore dumps for
                                # a single user's session

FILLFACTOR      90              # Fill factor for building indexes

# method for OnLine to use when determining current time
USEOSTIME       0       # 0: use internal time(fast), 1: get time from
OS(slow)

# Parallel Database Queries (pdq)
MAX_PDQPRIORITY 100    # Maximum allowed pdqpriority
DS_MAX_QUERIES         # Maximum number of decision support queries
DS_TOTAL_MEMORY        # Decision support memory (Kbytes)
DS_MAX_SCANS 1048576   # Maximum number of decision support scans
DATASKIP               # List of dbspaces to skip

# OPTCOMPIND
# 0 => Nested loop joins will be preferred (where
#      possible) over sortmerge joins and hash joins.
# 1 => If the transaction isolation mode is not
#      "repeatable read", optimizer behaves as in (2)
#      below.  Otherwise it behaves as in (0) above.
# 2 => Use costs regardless of the transaction isolation
#      mode.  Nested loop joins are not necessarily
#      preferred.  Optimizer bases its decision purely
#      on costs.
OPTCOMPIND      2      # To hint the optimizer

DIRECTIVES      1    # Optimizer DIRECTIVES ON (1/Default) or OFF (0)

ONDBSPACEDOWN   2      # Dbspace down option: 0 = CONTINUE, 1 = ABORT, 2 =
WAIT
OPCACHEMAX      0      # Maximum optical cache size (Kbytes)

# HETERO_COMMIT (Gateway participation in distributed transactions)
# 1 => Heterogeneous Commit is enabled
# 0 (or any other value) => Heterogeneous Commit is disabled
HETERO_COMMIT   0
```

```
SBSPACENAME            # Default smartblob space name - this is where blobs
                       # go if no sbspace is specified when the smartblob is
                       # created. It is also used by some datablades as
                       # the location to put their smartblobs.
SYSSBSPACENAME         # Default smartblob space for use by the Informix
                       # Server. This is used primarily for Informix Server
                       # system statistics collection.

BLOCKTIMEOUT    3600   # Default timeout for system block
SYSALARMPROGRAM    /global/informix/ids930uc1/evidence.sh    # System Alarm
program path

# Optimization goal: -1 = ALL_ROWS(Default), 0 = FIRST_ROWS
OPT_GOAL        -1

ALLOW_NEWLINE   0       # embedded newlines(Yes = 1, No = 0 or anything but
1)

#
# The following are default settings for enabling Java in the database.
# Replace all occurrences of /usr/informix with the value of $INFORMIXDIR.

#VPCLASS        jvp,num=1       # Number of JVPs to start with

JVPJAVAHOME     /global/informix/ids930uc1/extend/krakatoa/jre/       # JRE
installation root directory
JVPHOME          /global/informix/ids930uc1/extend/krakatoa # Krakatoa
installation directory

JVPPROPFILE     /global/informix/ids930uc1/extend/krakatoa/.jvpprops # JVP
property file
JVPLOGFILE      /global/informix/ids930uc1/jvp.log   # JVP log file.

JDKVERSION    1.3              # JDK version supported by this server

# The path to the JRE libraries relative to JVPJAVAHOME
JVPJAVALIB    /lib/sparc/

# The JRE libraries to use for the Java VM

JVPJAVAVM     hpi:server:verify:java:net:zip:jpeg

# use JVPARGS to change Java VM configuration
#To display jni call
#JVPARGS        -verbose:jni

# Classpath to use upon Java VM start-up (use _g version for debugging)

# JVPCLASSPATH
/usr/informix/extend/krakatoa/krakatoa_g.jar:/usr/informix/extend/krakatoa/jd
bc_g.jar
JVPCLASSPATH
/global/informix/ids930uc1/extend/krakatoa/krakatoa.jar:/global/informix/ids9
30uc1/extend/krakatoa/jdbc.jar
```