ORACLE®
SOLARIS

Migration to Oracle Solaris Cluster Data Service for SAP NetWeaver

ORACLE®

Table of Contents

Introduction to Oracle Solaris Cluster Data Service for SAP NetWeaver

Oracle Solaris Cluster is a high availability cluster hardware and software product for the Oracle Solaris operating system. It is used to improve the availability of hardware and software services and business continuity. Oracle Solaris Cluster operates by having redundant storage, network switches, computers (known as cluster nodes), the Solaris Cluster framework software, and cluster agents for the applications to provide high availability. Applications are administrated and monitored in resource groups which consists of one or more resources. Resource groups can be configured as fail over or multiple masters, depending on the application requirement.

Oracle Solaris Cluster 3.3 3/13 supports all the SAP products based on SAP NetWeaver 7.0, 7.01, 7.02, 7.03, 7.10, 7.11, 7.20, 7.30, and 7.31 with SAP kernel version updated to at least 720/720_EXT patch level 300 or 721_EXT patch level 130. Oracle Solaris Cluster 3.3 3/13 also supports all the SAP products based on SAP NetWeaver 7.4 with at least SAP kernel 740 patch level 36 or SAP kernel 741/742 patch level 28.

Following is a brief description of the types of SAP systems:

» **ABAP system** – The ABAP system contains the SAP Application Server ABAP. An ABAP system is developed and extended using ABAP (Advanced Business Application Programming), which is a high level programming language. High availability SAP ABAP system consists of the following instances:
   » ABAP Central Services instance ASCSnn (nn is instance number between 00 and 99)
   » ERS - Enqueue Replication Server instance ERSnn
   » Database instance
   » PAS - Central Instance or Primary Application Server instance DVEBMGSnn
   » AAS - (Optional) Dialog Instance or Additional Application Server instance Dnn

» **Java system** – The SAP Java system contains the SAP Application Server Java, it is developed and extended using Oracle's object-oriented programming language Java. High availability SAP Java system consists of the following instances:
   » Central Services instance SCSnn
   » ERS - Enqueue Replication Server instance ERSnn
   » Database instance
   » PAS - Central Instance or Primary Application Server instance JCnn or Jnn
   » AAS - (Optional) Dialog Instance or Additional Application Server instance Jnn

» **ABAP+Java double stack system** – A double stack system has both NetWeaver Application Server ABAP and NetWeaver Application Server Java in one system. High availability SAP ABAP+Java system consists of the following instances:
   » ABAP Central Services instance ASCSnn
   » Centranl Services instance SCSnn
   » Database instance
   » ERS – two Enqueue Replication Server instances ERSnn
   » PAS - Central Instance or Primary Application Server instance DVEBMGSnn
   » AAS - (Optional) Dialog Instance or Additional Application Server instance Dnn

The Oracle Solaris Cluster Data Service for SAP NetWeaver (Agent `ORCLscsapnetw`) is fully qualified by Oracle and also certified by SAP on Oracle Solaris Cluster 4.x and Oracle Solaris Cluster 3.3 3/13. The agent provides administrative and monitoring functions for (A)SCS, ERS, PAS and AAS instances to ensure that these instances are highly available. The agent is also integrated with SAP HA script connector via `sapstartsrv`.

This document describes how to migrate from Solaris Cluster Data Service for SAP WebAS (Agent `SUNWscsapwebas`) to Solaris Cluster Data Service for SAP NetWeaver (Agent `ORCLscsapnetw`).

Prerequisite Check of Solaris Cluster

1. Update to Oracle Solaris 10 8/11 (Solaris 10 update 10) or later.
2. Update to Oracle Solaris Cluster 3.3 3/13 (Solaris Cluster 3.3 update 2) or later.
3. Install the Data Service for SAP NetWeaver.
4. Install the latest patches of Solaris 10 8/11 or later.
5. Install the latest patches of Solaris Cluster 3.3 3/13 or later.
6. Install the latest patch of Data Service for SAP NetWeaver: patch 149081 (SPARC) or patch 149082 (x86_64).

Prerequisite Check of SAP System

1. Logon as `<sid>adm` user on one node, check the SAP kernel version and patch level. The kernel release should be 720 and the patch number should be higher than 300. If not, update to SAP kernel 720 latest patch level. The `<sid>adm` user is the SAP system administration user, where `<sid>` is the lower case of the SAP system ID, which has three characters or numbers.

```
disp+work
```

2. Logon as `<sid>adm` user on the node where the SAP instance is running, and check the process `sapstartsrv` of each SAP instance. In the following command, <nr> means the SAP instance number, which can be any number between 00 and 99 and was decided during the SAP installation.

```
sapcontrol -nr <nr> -function GetProcessList
```

If the process `sapstartsrv` is not running, try to start it on the node where the SAP instance is running. Do not start the `sapstartsrv` process of a SAP instance on node other than the one where the SAP instance is running. Here <SID> means upper case of the SAP system ID.

```
sapcontrol -nr <nr> -function StartService <SID>
```

If the process `sapstartsrv` can not be started correctly, check the file `/usr/sap/sapservices`. Each instance should have one line in the `/usr/sap/sapservices` file. For example, the following file include entries of two SAP systems. The SAP system QE1 has instances SCS00, ERS10, J02, J03, and J04. The SAP system QE2 has instance DVEBMGS01. For more information refer to the SAP note 823941.

```
#!/bin/sh

LD_LIBRARY_PATH=/usr/sap/QE1/SCS00/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/QE1/SCS00/exe/sapstartsrv pf=/usr/sap/QE1/SYS/profile/QE1_SCS00_dink-1 -D
-u qe1adm

LD_LIBRARY_PATH=/usr/sap/QE1/ERS10/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/QE1/ERS10/exe/sapstartsrv pf=/usr/sap/QE1/ERS10/profile/QE1_ERS10_dink-3
-D -u qe1adm

LD_LIBRARY_PATH=/usr/sap/QE1/J02/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/QE1/J02/exe/sapstartsrv pf=/usr/sap/QE1/SYS/profile/QE1_J02_dink-4 -D -u
qe1adm

LD_LIBRARY_PATH=/usr/sap/QE1/J03/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/QE1/J03/exe/sapstartsrv pf=/usr/sap/QE1/SYS/profile/QE1_J03_pdink1 -D -u
qe1adm

LD_LIBRARY_PATH=/usr/sap/QE1/J04/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/QE1/J04/exe/sapstartsrv pf=/usr/sap/QE1/SYS/profile/QE1_J04_pdink2 -D -u
qe1adm

LD_LIBRARY_PATH=/usr/sap/QE2/DVEBMGS01/exe:$LD_LIBRARY_PATH; export
```

```
LD_LIBRARY_PATH; /usr/sap/QE2/DVEBMGS01/exe/sapstartsrv
pf=/usr/sap/QE2/SYS/profile/START_DVEBMGS01_dink-6 -D -u qe2adm
```

Note: Each instance has one line in the above file, do not split them in multiple lines.

3. Install SAP Host Agent on each cluster node. Make sure SAP host agent is located on local file system. Download the latest SAP Host Agent 7.20 from http://support.sap.com/swdc. In the navigation bar, choose → SAP Software Download Center → Support Packages and Patches → Browse our Download Catalog → SAP Technology Components → SAP HOST AGENT → SAP HOST AGENT 7.20 → Solaris <SPARC> or <X86_64>.

Logon to the cluster node with `root` user. Create a temporary directory (e.g. `/tmp/hostagent`). Change directory to the temporary directory and extract the downloaded `SAPHOSTAGENT<patchnr.>.SAR` file via SAPCAR. Install SAP Host Agent with command "`./saphostexec -install`". The SAP Host Agent will be installed in folder `/usr/sap/hostctrl`. Make sure it is on local file system on each cluster node. Refer to SAP note 1031096 for more information.

```
mkdir /tmp/hostagent
cd /tmp/hostagent
/usr/sap/<SID>/SYS/exe/run/SAPCAR -xfv <download_folder>/SAPHOSTAGENT<patchnr.>SAR

./saphostexec -install
```

The SAP Host Agent should be installed and started with above commands. After installation, you may also use following commands to stop/start the SAP Host Agent:

```
/usr/sap/hostctrl/exe/saphostexec -stop
/usr/sap/hostctrl/exe/saphostexec -start
```

4. Make sure the file `/etc/rc3.d/S90sapinit` does not exist. If it exists, rename it to other name. This is to ensure the `sapstartsrv` processes will not be started automatically during server reboot. Perform this step on all the related cluster nodes. The `sapstartsrv` processes will be started and stopped by the Solaris Cluster resources created later.

5. Double check to make sure the following files are identical on all the cluster nodes:

```
/etc/group
/etc/passwd
/etc/project
/etc/user_attr
/etc/shadow
```

6. Make sure the following files and folders identical as on the installation node:

```
/etc/services   (may be modified by SWPM/SUM)
/export/home    (may be modified by SWPM/SUM)
/usr/local/bin (for Oracle Database)
/var/opt/oracle (for Oracle Database)
/oracle/client  (for Oracle client, required by SAP instances)
/usr/sap/hostctrl    (SAP Host Agent, must be installed on local file system)
/usr/sap/sapservices*
/usr/sap/ccms
/usr/sap/tmp
/usr/sap/<SID>/SYS
```

7. Make sure the following files and folders are mounted on all the cluster nodes.

```
/sapmnt/<SID>
/usr/sap/trans
```

Delete the Existing Resources of HA WebAS

1. Log in to a cluster node with `root` user.

2. Disable the resources. Repeat this step for all the existing resources configured for the SAP instances, include the Dialog Instance, the Central Instance, the Enqueue Replication Server resource, the Enqueue Server resource, and the Message Server resource. The corresponding SAP instances should be stopped:

```
clrs disable <dialog_instance_resource>
clrs disable <central_instance_resource>
clrs disable <replicated_enqueue_server_resource>
clrs disable <enqueue_server_resource>
clrs disable <message_server_resource>
```

3. Delete the resources which were disabled in above step.

```
clrs delete <dialog_instance_resource>
clrs delete <central_instance_resource>
clrs delete <replicated_enqueue_server_resource>
clrs delete <enqueue_server_resource>
clrs delete <message_server_resource>
```

4. Unregister the resource types.

```
clresourcetype unregister SUNW.sapenq
clresourcetype unregister SUNW.saprepl
clresourcetype unregister SUNW.sapscs
clresourcetype unregister SUNW.sapwebas
```

5. Remove the strong negative affinity between the resource groups of (A)SCS and ERS instances. The Data Service for SAP NetWeaver uses different timing and different method to switch the enqueue replication server as the Data Server for SAP WebAS. The strong negative affinity is not required any more. The ERS instance resource group will be switched by the preempter resource which will be created later.

```
clrg show -p RG_affinities <replicated_enqueue_server_resource_group>
clrg set -p RG_affinities="" <replicated_enqueue_server_resource_group>
```

Recover the Modifications Made for HA WebAS

1. The HA WebAS requires the ERS instance number be the same as the (A)SCS instance number. But for the HA NetWeaver, the ERS instance number must be different as the (A)SCS instance number. Check the instance profile `/sapmnt/<SID>/profile/<SID>_ERS<nr>_<hostname>`, make sure the parameter `SAPSYSTEM` is other than the parameter SCSID. Give a new instance number to the ERS instance via changing the parameter `SAPSYSTEM`.

   For example, if the parameters are the same:

```
SAPSYSTEM=00
…
SCSID=00
```

Change the parameter SAPSYSTEM to a new instance number:

```
SAPSYSTEM=10
…
SCSID=00
```

2. Change the ERS instance name accordingly. Modify the folder name `/usr/sap/<SID>/ERS<nr>` and the file name `/sapmnt/<SID>/profile/<SID>_ERS<nr>_<hostname>`. Adjust the parameter `INSTANCE_NAME` as well.

For example, if you have changed the instance number from 00 to 10, then:

» Go to folder `/usr/sap/<SID>`, change the subfolder `ERS00` to `ERS10`

» Go to folder `/sapmnt/<SID>/profile`, change the file name `<SID>_ERS00_<hostname>` to `<SID>_ERS10_<hostname>`

» Change the parameter `INSTANCE_NAME` from `ERS00` to `ERS10` in the instance profile `/sapmnt/<SID>/profile/<SID>_ERS10_<hostname>`

3. Modify the start profile or instance profiles of the (A)SCS instance, make sure the message server is using `Restart_Program` and the enqueue server is using `Start_Program`. This is to ensure that the message server is restarted by `sapstartsrv`, and the enqueue server is *not* restarted by `sapstartsrv`. For example:

```
#-----------------------------------------------------------------------
# Start SAP message server
#-----------------------------------------------------------------------
_MS = ms.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_02 = local rm -f $(_MS)
Execute_03 = local ln -s -f $(DIR_EXECUTABLE)/msg_server$(FT_EXE) $(_MS)
Restart_Program_00 = local $(_MS) pf=$(_PF)
#-----------------------------------------------------------------------
# Start SAP enqueue server
#-----------------------------------------------------------------------
_EN = en.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_04 = local rm -f $(_EN)
Execute_05 = local ln -s -f $(DIR_EXECUTABLE)/enserver$(FT_EXE) $(_EN)
Start_Program_01 = local $(_EN) pf=$(_PF)
```

4. Check the instance profile of ERS, make sure the enqueue replication server is using `Restart_Program`.

```
#-----------------------------------------------------------------------
# Start enqueue replication server
#-----------------------------------------------------------------------
_ER = er.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_03 = local rm -f $(_ER)
Execute_04 = local ln -s -f $(DIR_EXECUTABLE)/enrepserver$(FT_EXE) $(_ER)
Restart_Program_00 = local $(_ER) pf=$(_PFL) NR=$(SCSID)
```

5. Modify the file `/usr/sap/sapservices`, make the entry for ERS instance is using the correct instance name.

```
LD_LIBRARY_PATH=/usr/sap/QE1/ERS10/exe:$LD_LIBRARY_PATH; export LD_LIBRARY_PATH;
/usr/sap/QE1/ERS10/exe/sapstartsrv
```

6. Double check with your SAP consultant to make sure the SAP instances are installed and configured correctly.

Make Sure SAP Instances Can Be Started/Stopped on Both Nodes

1. Logon as `<sid>adm` user and stop all the SAP instances if they are running. Stop all the `sapstartsrv` processes as well if they are running.

```
stopsap r3
sapcontrol -nr <nr> -function StopService
```

2. Double check that all the SAP processes have been stopped except the SAP Host Agent processes.

```
ps -ef | grep -i sap | grep -i <SID>
```

3. Identify the resource groups which have the logical host's resources. Switch the resource groups to one cluster node. Make sure the logical hosts are online and the file systems are available.

```
clrg switch -n <node1> scs-rg
clrg switch -n <node1> ers-rg
clrg switch -n <node1> pas-rg
clrg switch -n <node1> aas-rg
```

4. Make sure the database is running and available by SAP system. For ABAP-only or ABAP+Java system, this can be checked with following command as the `<sid>adm` user.

```
R3trans -d
```

5. Log in as the `<sid>adm` user on the node1. Start each `sapstartsrv` process of every SAP instance with following command.

```
sapcontrol -nr <nr> -function StartService <SID>
```

6. Start each SAP instance with following command in following sequence: (A)SCS, ERS, PAS, and AAS. The short term PAS is the primary application server instance DVEBMGSnn or JCnn or Jnn. AAS in the additional application server instance Dnn or Jnn.

```
sapcontrol -nr <nr> -function Start
```

You may also use the command `startsap r3 <instance>` to start each SAP instance.

7. Check the status of each SAP instance. Every listed process should be in GREEN status.

```
sapcontrol -nr <nr> -function GetProcessList
```

8. Log in to the SAP system using the SAP GUI or using the internet explorer. Verify that the SAP system is running.

9. Repeat steps 1 to 8 and start the SAP instances on node2.

Configure Oracle Solaris Cluster Resources with HA NetWeaver

This chapter is an example of a Java-only SAP system. The SAP instances and Solaris Cluster resource types are used as below:

» SCS00 – Standalone Central Services instance, which has the message server and the enqueue server. Solaris Cluster resource type `ORCL.sapcentr` is required.

» ERS10 – Enqueue Replication Server instance, which has the replicated enqueue server. Solaris Cluster resource type `ORCL.saprepenq` is required.

» J01 – Primary Application Server instance, which has the J2EE server and IGS watchdog processes. Solaris Cluster resource type `ORCL.sapdia` is required.

» J02 – Additional Application Server instance, which also has the J2EE server and the IGS watchdog processes. Solaris Cluster resource type `ORCL.sapdia` is required.

» Each SAP instance has a `sapstarsrv` process, which requires the Solaris Cluster resource type `ORCL.sapstartsrv`.

» Furthermore, after the (A)SCS instance fails over to the node where the ERS instance is running, the enqueue lock table will be handed over from the enqueue replication server to the enqueue server. Then the ERS instance should be sent to another cluster node. This requires the `ORCL.saprepenq_preempt` resource type.

For more information about the Oracle Solaris Cluster Data Service for SAP NetWeaver, see this Oracle Solaris Cluster Documentation.

1. Log in as the *sid*adm user on the nodes on which the SAP instance is running. Stop the SAP instances on all the nodes.

```
stopsap r3 J02 <logical_hostname>
stopsap r3 J01 <logical_hostname>
stopsap r3 ERS10 <logical_hostname>
stopsap r3 SCS00 <logical_hostanme>
```

2. Stop all the `sapstarsrv` processes.

```
sapcontrol -nr 02 -function StopService
sapcontrol -nr 01 -function StopService
sapcontrol -nr 10 -function StopService
sapcontrol -nr 00 -function StopService
```

3. Log in as the `root` user on both nodes and stop the SAP host agent processes.

```
root@vzstar1b:~# /usr/sap/hostctrl/exe/saphostexec -stop
root@vzstar2b:~# /usr/sap/hostctrl/exe/saphostexec -stop
```

4. Type `ps -ef | grep -i sap` command on all the nodes to ensure that the SAP process is not running.

5. The weak positive affinity of the `scs-rg` resource group to the `ers-rg` resource group should still be active. If not, set `rg_affinities` property for the `scs-rg` resource group. Type the following commands to declare weak positive affinity for the `ers-rg` resource group. This is to ensure the `scs-rg` resource group fails over to the node where the `ers-rg` resource group is running.

```
clrg set -p RG_affinities=+ers-rg scs-rg
clrg show -p RG_affinities scs-rg
```

6. Register the required Solaris Cluster resource types.

```
clrt register ORCL.sapstartsrv
clrt register ORCL.sapcentr
clrt register ORCL.saprepenq
clrt register ORCL.sapdia
clrt register ORCL.saprepenq_preempt
```

7. Create the resource for `sapstartsrv` of Central Service instance (A)SCSnn. In this example, the SAP system ID is BNZ and the instance is SCS00 on logical host star-11. The property `child_mon_level` is set to 5 to ensure the SAP instance is not restarted while restarting the `sapstartsrv` process. The property `resource_dependencies_offline_restart` is set to the file system resource to ensure the process `sapstartsrv` is restarted while the file system has been lost and back online.

```
# clrs create -d -g scs-rg -t ORCL.sapstartsrv \
> -p SID=BNZ \
> -p sap_user=bnzadm \
> -p instance_number=00 \
> -p instance_name=SCS00 \
> -p host=star-11 \
> -p child_mon_level=5 \
> -p resource_dependencies_offline_restart=hasp4sap \
> scs-startsrv-rs
```

8. (Optional) Set the tunable properties for the resource of `sapstartsrv` process of (A)SCSnn instance.

```
# clrs set \
> -p start_timeout=60 \
> -p stop_timeout=60 \
> -p probe_timeout=10 \
> -p thorough_probe_interval=10 \
> -p timeout_return=20 \
> scs-startsrv-rs
```

9. Create the resource of the (A)SCS instance. The property `retry_count` is set to 0 to ensure the (A)SCS instance is not restarted on the same node, but failover to the node where the ERS instance is running (together with the RG_affinity between the two resource groups). For the storage resource, if you are using scalable storage resources, or failover local file system, you can set the `resource_dependencies_offline_restart` property on storage resources. If you are using the failover global file system, set the `resource_dependencies` property instead. This is to avoid the (A)SCS and ERS instances are restarted together while the global file system is failed over.

```
# clrs create -d -g scs-rg -t ORCL.sapcentr \
> -p SID=BNZ \
> -p sap_user=bnzadm \
> -p instance_number=00 \
> -p instance_name=SCS00 \
> -p host=star-11 \
> -p retry_count=0 \
> -p resource_dependencies=hasp4sap,scs-startsrv-rs \
> scs-rs
```

10. (Optional) Set the tunable properties for the resource of (A)SCSnn instance.

```
# clrs set \
> -p start_timeout=60 \
> -p stop_timeout=60 \
> -p probe_timeout=10 \
> -p thorough_probe_interval=10 \
> -p yellow=20 \
> scs-rs
```

11. Create the resource for the `sapstartsrv` process of the ERS instance.

```
# clrs create -d -g ers-rg -t ORCL.sapstartsrv \
> -p SID=BNZ \
> -p sap_user=bnzadm \
> -p instance_number=10 \
> -p instance_name=ERS10 \
> -p host=star-12 \
> -p child_mon_level=5 \
> -p resource_dependencies_offline_restart =hasp4sap \
> ers-startsrv-rs
```

12. (Optional) Set the tunable properties for the resource of `sapstartsrv` process of ERS instance.

```
# clrs set \
> -p start_timeout=60 \
> -p stop_timeout=60 \
> -p probe_timeout=10 \
> -p thorough_probe_interval=10 \
> -p timeout_retrun=20 \
> ers-startsrv-rs
```

13. Create the resource for the ERS instance. Take care of the file system resource as the same for the (A)SCS resource.

```
# clrs create -d -g ers-rg -t ORCL.saprepenq \
> -p sid=BNZ \
> -p sap_user=bnzadm \
> -p instance_number=10 \
> -p instance_name=ERS10 \
> -p host=star-12 \
> -p resource_dependencies=hasp4sap,ers-startsrv-rs \
> ers-rs
```

14. (Optional) Set the tunable properties for the resource of ERS instance.

```
# clrs set \
> -p start_timeout=60 \
> -p stop_timeout=60 \
> -p probe_timeout=10 \
> -p thorough_probe_interval=10 \
> -p yellow=20 \
> ers-rs
```

15. Create the `preempter` resource of the ERS instance in the resource group of (A)SCS instance. After the (A)SCS resource group failed over to the node where the ERS instance is running, the replicated enqueue lock table was handed over from ERS to (A)SCS instance. The preempter is responsible for switching the ERS resource group to another node after the above procedure. The property `repenqres` indicates the resource of ERS instance (replicated enqueue server resource). The property `enq_instnr` indicates the (A)SCS instance number (enqueue server instance number).

```
# clrs create -d -g scs-rg -t ORCL.saprepenq_preempt \
> -p sid=BNZ \
> -p sap_user=bnzadm \
> -p repenqres=ers-rs \
> -p enq_instnr=00 \
> -p resource_dependencies_offline_restart=hasp4sapjava,scs-rs \
> preempter-rs
```

16. Create the resource for the `sapstartsrv` process of the Primary Application Server instance. It is the instance J01 in the following example. For other SAP system, the primary application server can be named as DVEBMGSnn, JCnn or Jnn according to the SAP system type and software release, where `nn` is the instance

number, which can be any number between 00 and 99. The property `child_mon_level` can be set to 5 if the SAP kernel 720 patch level is 401 or later.

```
# clrs create -d -g pas-rg -t ORCL.sapstartsrv \
> -p sid=BNZ \
> -p sap_user=bnzadm \
> -p instance_number=01 \
> -p instance_name=J01 \
> -p host=star-13 \
> -p child_mon_level=5 \
> -p resource_dependencies_offline_restart=hasp4sap \
> pas-startsrv-rs
```

17. (Optional) Set the tunable properties for the resource of `sapstartsrv` process of Primary Application Server instance.

```
# clrs set \
> -p start_timeout=60 \
> -p stop_timeout=60 \
> -p probe_timeout=10 \
> -p thorough_probe_interval=10 \
> -p timeout_retrun=20 \
> pas-startsrv-rs
```

18. Create the resource for the Primary Application Server instance (in the following example: J01).

```
# clrs create -d -g pas-rg -t sapdia \
> -p SID=BNZ \
> -p sap_user=bnzadm \
> -p instance_number=01 \
> -p instance_name=J01 \
> -p host=star-13 \
> -p resource_dependencies_offline_restart=hasp4sap \
> -p resource_dependencies=oracle-rs,listener-rs,scs-rs,pas-startsrv-rs \
> pas-rs
```

19. (Optional) Set the tunable properties for the resource of PAS instance.

```
# clrs set \
> -p start_timeout=300 \
> -p stop_timeout=120 \
> -p probe_timeout=10 \
> -p thorough_probe_interval=10 \
> -p yellow=20 \
> pas-rs
```

20. Repeat steps 16 through 19 for the Additional Application Server instance (in our example: the instance is J02) to create resources (in our example: `aas-startsrv-rs`, `aas-rs`) in the resource group (in our example: `aas-rg`).

21. Depending on the storage topology, add the required affinities and dependencies to the resource groups, to ensure that the resource group always has access to the necessary file system.

In the following example resource groups declare strong positive affinity for the `hasp` resource group.

```
clrg set -p RG_affinities+=++hasp4sap-rg ers-rg
clrg set -p RG_affinities+=++hasp4sap-rg scs-rg
clrg set -p RG_affinities+=++hasp4sap-rg pas-rg
clrg set -p RG_affinities+=++hasp4sap-rg aas-rg
```

22. Set the `Resource_project_name` parameter for the resources. The resource project name is the same as defined in the `/etc/project` and `/etc/user_attr` for the user `<sid>adm`.

```
clrs set -p Resource_project_name=BNZ scs-rs
clrs set -p Resource_project_name=BNZ ers-rs
clrs set -p Resource_project_name=BNZ pas-rs
clrs set -p Resource_project_name=BNZ aas-rs
```

23. Enable the resources and check the status of the resources. The resource group `ers-rg` should be switched by the preempter to another node than the `scs-rg` resource group is running.

```
clrs enable scs-startsrv-rs
clrs enable scs-rs
clrs enable preempter-rs
clrs status -g scs-rg

clrs enable ers-startsrv-rs
clrs enable ers-rs
clrs status -g ers-rg

clrs enable pas-startsrv-rs
clrs enable pas-rs
clrs status -g pas-rg

clrs enable aas-startsrv-rs
clrs enable aas-rs
clrs status -g aas-rg
```

(Optional) Enable Oracle Solaris Cluster HA Connector for `sapstartsrv`

Oracle Solaris Cluster agent for SAP NetWeaver (`ORCL.ha-netweaver`) has an option which enables Oracle Solaris Cluster HA Connector for `sapstartsrv`.

If you do not enable Oracle Solaris Cluster HA Connector with `sapstartsrv`, the Oracle Solaris Cluster will restart the resources or instances automatically when the SAP administrator stops the SAP instances with `stopsap`, `sapcontrol` or `SAP MMC`. The administrator must use Oracle Solaris Cluster commands to stop the SAP instances.

If you enable the Oracle Solaris Cluster HA Connector with `sapstartsrv`, the administrator can stop the instances using either the Oracle Solaris Cluster 4 commands or the SAP commands which include `stopsap/startsap` script, `SAP MMC`, and `sapcontrol` command.

Perform the following steps to enable the Oracle Solaris Cluster HA Connector for `sapstartsrv`:

1. Copy the following three parameters into the default profile `/sapmnt/<SID>/profile/DEFAULT.PFL` of the SAP system or to the instance profile files of each instance.

Note - The second parameter and its value should be entered in one line.

```
#
# SAP HA Script Connector
#
service/halib = /usr/sap/<SID>/SYS/exe/run/saphascriptco.so

service/halib_cluster_connector =
/opt/ORCLscsapnetw/saphacmd/bin/sap_orcl_cluster_connector

service/halib_debug_level = 1
```

2. Ensure that the *sid*adm user has the cluster administration privilege on all the nodes. If not, grant the cluster administration privilege to the *sid*adm user.

```
usermod -A solaris.cluster.admin <sid>adm

/* Or add an entry in /etc/usr_attr on all nodes. For example: */
bnzadm::::type=normal;auths=solaris.cluster.admin;project=BNZ
```

3. Restart the cluster resource groups for SAP instances to activate the changes. You will see `SAP_HA_OK` in the `/usr/sap/<SID>/<Instance>/work/sapstartsrv.log` log file.

```
clrg restart ers-rg
clrg restart scs-rg
clrg restart pas-rg
clrg restart aas-rg
```

Note: The resource group `scs-rg` should fail over to the node where `ers-rg` is running. The resource group `ers-rg` will be switched to another node afterwards.

4. Besides Solaris Cluster commands, you may now use the `startsap` or `stopsap` scripts, or the `sapcontrol` commands to start or stop the SAP instances. Alternatively, you can go to `http://<host>:5<nr>13` and start the SAP management console.

5. Log in as the *sid*adm user to the SAP management console. If the password does not work, perform the script with `root` user, this will set the SUID bit for the `sapuxuserchk` file of each instance.

```
/usr/sap/<SID>/SYS/exe/run/saproot.sh
```

You may also manually set up the SUID bit configuration. Log in as the `root` user. Change to the executable directory (`/usr/sap/<SID>/<instance>/exe`) that contains `sapuxuserchk` and type the following commands. Refer to SAP note 927637 for more information.

```
chown root:sapsys sapuxuserchk
chmod u+s,o-rwx sapuxuserchk
```

Troubleshooting

Problem: You cannot switch the primary application server instance on a Java system even when the Oracle database is running. The following error message is displayed:

```
root@vzstar1b:~# clrg switch -n vzstar2b pas-rg
clrg:  (C748634) Resource group pas-rg failed to start on chosen node and might
fail over to other node(s)
root@vzstar1b:~# Jun  4 11:35:17 vzstar1b SAPBNZ_01[29942]: Unable to open trace
file sapstartsrv.log. (Error 11 Resource temporarily unavailable)
[ntservsserver.cpp 3231]


vzstar2b.console
================
Jun  4 11:19:11 vzstar2b SAPBNZ_01[13504]: Unable to open trace file
sapstartsrv.log. (Error 11 Resource temporarily unavailable) [ntservsserver.cpp
3231]

Jun  4 11:20:07 vzstar2b SC[SUNWscor.oracle_server.monitor]:oracle-rg:oracle-rs:
Fault monitor detected error DBMS_ERROR: 99999 DEFAULT Action=RESTART : Monitor
detected death of Oracle background processes.

Jun  4 11:20:07 vzstar2b SC[SUNWscor.oracle_server.monitor]:oracle-rg:oracle-rs:
Restarting using scha_control RESOURCE_RESTART

Jun  4 11:20:59 vzstar2b SC[SUNWscor.oracle_server.start]:oracle-rg:oracle-rs:
Media error encountered, but Auto_end_bkp is disabled.

Jun  4 11:20:59 vzstar2b SC[SUNWscor.oracle_server.start]:oracle-rg:oracle-rs:
Could not start server
```

Workaround: This error occurs because of the following section which exists in the SAP instance profile `/usr/sap/<SID>/SYS/profile/<SID>_<Instance>_<hostname>` or in the SAP start profile `/usr/sap/<SID>/SYS/profile/START_<Instance>_<hostname>`.

```
#-----------------------------------------------------------------------
# Start J2EE database
#-----------------------------------------------------------------------
_DB = db.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_05 = immediate rm -f $(_DB)
Execute_06 = immediate ln -s -f $(DIR_CT_RUN)/startj2eedb $(_DB)
Start_Program_00 = immediate $(_DB)
exe/saposcol = $(DIR_CT_RUN)/saposcol
```

This section means that the `startj2eedb` script is always executed to check if the database is running. Since the `R3trans` program does not work for a Java system, it only checks if the Oracle processes are running on the host.

The `startj2eedb` script cannot get the correct database status when the database is running on another node. To solve this, comment out the line `Start_Program_xx` of the database in the profile.

```
#----------------------------------------------------------------------
# Start J2EE database
#----------------------------------------------------------------------
_DB = db.sap$(SAPSYSTEMNAME)_$(INSTANCE_NAME)
Execute_05 = immediate rm -f $(_DB)
Execute_06 = immediate ln -s -f $(DIR_CT_RUN)/startj2eedb $(_DB)
# Start_Program_00 = immediate $(_DB)
exe/saposcol = $(DIR_CT_RUN)/saposcol
```

Conclusion

Oracle Solaris Cluster provides high availability for SAP central services instance, enqueue replication server instance, database instance, as well as for primary application server (former central instance except message server and enqueue server) and additional application servers (former dialog instances). You may install each SAP system on its own Oracle Solaris zones to provide more isolation and protection.

To install highly available SAP on Oracle Solaris Cluster, you must have information about SAP products and Oracle Solaris Cluster.

About the Authors

This document is based on Xirui Yang's and Detlef Ulherr's experience of High Available SAP on Solaris Cluster. Xirui Yang is a Principal Software Engineer and expert in Oracle's ISV-Engineering team for SAP, specializing in HA technologies for SAP. Detlef Ulherr is a Principal Software Engineer for Oracle Solaris Cluster Agent products.

ORACLE®

Oracle Corporation, World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065, USA

Worldwide Inquiries
Phone: +1.650.506.7000
Fax: +1.650.506.7200

Integrated Cloud Applications & Platform Services

Migration to Oracle Solaris Cluster Data Service for SAP NetWeaver
November 2015
Authors: Xirui Yang and Detlef Ulherr