Supplementary Document to:

Invincible Supply Chain

Reference Architecture for Mission-Critical SAP® Advanced Planning & Optimization

Implementation Guide for SAP HotStandby liveCache with PowerHA 7.1.1

Publication Date 2/ August 2012

IBM

About This Document

This implementation document is related to the case study document on mission critical SCM system. The case study document "The Invincible Supply Chain" illustrates a top to bottom reference architecture design for a mission critical SAP supply chain system including the design overview for the HotStandby liveCache. The case study document, along with the administration manual for HotStandby can be found at the following website:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

This implementation document focuses primarily on the implementation of the HotStandby liveCache on the IBM SAN Volume Controller.

Table of Contents

1 Preface

1.1 Document Scope

This document is intended as an implementation guide based on the design for SAP liveCache HotStandby depicted in the primary document “The Invincible Supply Chain”.
This document covers the installation path using both the PowerHA 7.1.1 installation automation tools: Wizard and Smart Assist as well as a manual installation method. The installation was tested with GA PowerHA 7.1.1 with SP1 and the first official efix for SP1. This document is the result of the rigorous regression testing prior to the first customer installation and a number of small issues were found which require work around actions - these are detailed here. These issues will be removed in later versions of the product.

The document is provided for teams implementing a highly available SAP SCM APO system with PowerHA 7.1.1 HotStandby liveCache.

1.2 Special Notices

1.3 Version Information

This is version 2.0 of this document.

1.4 Authors of this Document

- Katharina Probst, SME HA, Böblingen Lab, IBM Germany
- Marc-Stephan Tauchert, SAP Solutions Architect, SAP Growth Program, Germany
- Carol Davis, Consulting IT Specialist, IBM SAP International Competence Center

2 Introduction to SAP liveCache HotStandby

The HotStandby solution documented here is based on an API provided by SAP which drives the HotStandby functionality. This section provides a quick review of the SAP HotStandby API.

2.1 SAP liveCache with HotStandby design and requirements

The following information over the implementation of SAP MaxDB with HotStandby is thanks to the SAP Labs in Berlin where SAP MaxDB and SAP liveCache are developed. A few comments are added to connect the general design for SAP MaxDB with HotStandby to implementation done for SAP liveCache on IBM System Storage. The focus in this document is on SAP liveCache with HotStandby as a component of SAP Advanced Planning & Optimization.

2.1.1 SAP MaxDB/SAP liveCache with HotStandby

A HotStandby differs from a conventional failover HA solution in several ways. In a HotStandby solution, both databases are running in parallel in an active/passive partnership. The standby is maintained in a continuous restart state which allows it to maintain synchronization with the production database. This is done by reapplying log records from all transactions to the standby database. The STANDBY status of the database, between ADMIN and ONLINE makes it possible for the standby to switch to active mode in a very short time, and maintain data consistency by completing all transactions.

The design of the liveCache HotStandby relies on the functionality of the storage subsystem. The requirements are the ability to generate a full stand-alone read-write split mirror[1] of the SAP liveCache data and concurrent access by both active and standby server to the database log volumes.

The concurrent log volumes are written by the active SAP liveCache, and read by the standby. This is the mechanism for insuring that all data is synchronized in the standby. There is an ongoing communication between the active and standby SAP liveCache to keep the standby informed of the most current log record and log volume position.

In the case of a failure of the active SAP liveCache, the standby commits any outstanding transactions in the log, takes control of the log (switches to write mode) and becomes the active SAP liveCache. The SAP liveCache database instances are peers and provide rotating standby. When the failed primary server is reactivated, it will become the standby.

SAP liveCache with HotStandby relies on the operating system to provide a cluster solution to detect the failure of the active SAP liveCache, switch the standby server to active status, and initiate the transfer of the SAP liveCache service IP address from the failed server to the server assuming active status.

With HotStandby, the time needed for starting the database instance and building the memory structures is saved. This is important in the case of SAP liveCache due to the very large memory structures which must be initialized while starting. Additionally, the time needed for restoring log information is reduced to nearly nothing.

[1] This is a full data copy that can be modified on its own behalf – not a snapshot or partial copy that retain dependencies on the source copy.

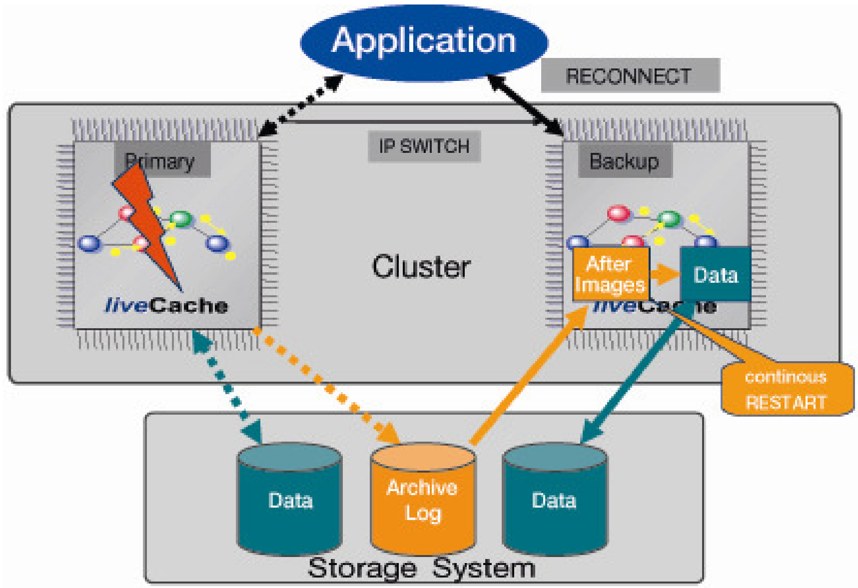


Figure 1: Overview of SAP liveCache HotStandby crash recovery provided by SAP

Basis of SAP MaxDB and liveCache with HotStandby
The HotStandby solution provided by SAP is valid for both SAP MaxDB and SAP liveCache. SAP liveCache uses the same database technology and in this document these two are often used interchangeably when referring to the HotStandby.

The solution, as supported in this HotStandby implementation, consists of two physically separate servers with physically shared storage for the liveCache log and a method of reliable data replication used to resynchronize the liveCache instances.

Each server must have its own unique network address. Additionally a reliable IP address must be maintained to provide client access to the current active liveCache server. This is an IP alias which is moved between the two servers as required. The cluster software (PowerHA in this case) is used to detect a situation necessitating failover and to perform the operations needed to redirect client connections (IP alias takeover) to the active liveCache server.

The SAP liveCache with HotStandby implementation is based on two or more separate database servers (the IBM solution documented here supports a cluster pair) that access a shared storage system. The control logic of the storage integration is implemented by a storage library (named libHSS). The data-volumes for each database are separate and the log volume is shared.

Between master and standby instances, a synchronization channel is established which is only needed to transfer synchronization information (such as the last write position in the log volume) but not for data transfer. The bandwidth for this link can be quite small.

The LOG and DATA volumes have special requirements
The access type to the LOG volumes is read-write to the master, and read-only to the standby server. Access is concurrent. The IBM solution does not restrict the standby server to read-only but relies on the logic of the takeover to ensure that only one instance is actively writing to the LOG volume.

A fast mirror of the DATA volumes (so called split mirror or snapshot) is necessary. This allows the standby server DATA volumes to be established using the current image of the master server's DATA volumes and vice versa (if master/standby roles are switched). The IBM solution presented here uses the FlashCopy functionality of the IBM storage servers to generate the split mirror. This functionality is triggered by the liveCache through a storage library (libHSS).

After a mirror of the DATA volumes is established, separation must be possible which will allow both the master and the standby servers to mount their DATA volumes for read/write. The IBM FlashCopy functionality establishes a logical copy within seconds, which can then be used as a completely stand-alone and totally consistent copy. The actual physical coping of data blocks continues in background while the new copy is already read/write capable.

The SAP MaxDB runtime is extended by the API functions to allow mirror establishing, mirror separation, and read-only/read-write switching. These routines have a separate layer that abstracts the storage system (RETHSS_API). This API is the basis of the shared library integration of SAP liveCache with IBM System Storage.

The SAP application programming interface (API) allows storage solution providers to integrate their storage functionality with the HotStandby control mechanisms of the SAP liveCache. This API exports the SAP liveCache logic that is then mapped to the functionality of the target storage server which fulfills the necessary requirements. The end result of this API integration is a shared library which is then made available to SAP liveCache and enables the HotStandby.

Beside the storage integration, the integration between the APO application and the cluster is also provided by this SAP solution. This aspect will be illuminated in section 3.5.

liveCache configuration for HotStandby
The liveCache configuration parameters are shared between all HotStandby database servers. They consist of the normal set of parameters for SAP liveCache databases and some extended parameters for the HotStandby solution. The parameters of liveCache database instances are read only once during startup. This means that the configuration file cannot be dynamically modified. However, HotStandby nodes can be added if the master is running.

The relevant liveCache parameters:
HS_NODE_00X:
The SAP MaxDB (or SAP liveCache) runtime has an additional routine that allows the SAP MaxDB kernel to identify itself on the basis of this variable. This local node name must be a valid network name as it is used by the master instance to establish the synchronization channel to the standby instance. Therefore each must be unique and assigned to a separate machine. In newer liveCache versions the name is changed to HotStandbyNodeNameNNN. The value will be searched by using the output of **uname –n** on UNIX® systems. The default master node is stored in HS_NODE_001 by convention.
In this solution, there is a single standby server and so there are three IP addresses: node1 (HS_NODE_001), node2 (HS_NODE_002) and the IP alias (OFFICIAL_NODE).

OFFICIAL_NODE:
The official node name is used for client access to the master node – the virtual IP used for the service IP alias. It is used by all clients to access the master. Each instance will use the OFFICIAL_NODE for storing the official hostname in the SAP MaxDB system tables. Hence, it must be shared over all instances.
The OFFICIAL_NODE must not match any of the HotStandbyNodeNameNNN entries, as this is the IP alias used for the service address, and therefore cannot be bound to a node.

HS_STORAGE_DLL:
The name of the storage access library (libHSS) which implements the RTEHSS_API.

HotStandbySyncInterval:
Defines how often the master sends synchronization information to the standby, instructing the standby to continue with log recovery. The default value is 50 seconds.

All other parameters are common, especially the volume names and sizes, the logical name of the liveCache instance and the cache size.

Example taken from kernel output on the HotStandby cluster:

Parameter names prior version 7.7	**Parameter names from version 7.7 onwards**
HS_STORAGE_DLL=libHSSibm2145	HotStandbyStorageDLLPath libHSSibm2145.so
OFFICIAL_NODE=LCHLCIP	OfficialNodeName LCHL2IP
HS_SYNC_INTERVAL=50	HotStandbySyncInterval 50
HS_NODE_001=IS03D11	HotStandbyNodeName001 IS03D11
HS_NODE_002=IS04D11	HotStandbyNodeName002 IS04D11

The names with uppercase letters are old parameter names used prior version 7.7. The old names can still be used for compatibility reasons.

Below are some of the commands that can be used at DBMCLI level to define the HotStandby. The SAP GUI database manager tool can be used to implement the HotStandby setup directly as well. The implementation in section 5 will describe the details. See the section 13.1 "Related documents and sources of further information" for more details on HotStandby commands.

Commands for Managing Hot Standby Systems

Commands for Experts

DBM Command	Description
hss_addstandby	Defining a standby instance
hss_copyfile	Copying database files in the cluster
hss_enable	Defining a database instance as the master instance
hss_execute	Executing DBM commands in a standby instance
hss_getnodes	Displaying the hot standby parameters
hss_removestandby	Deleting a standby instance

Figure 2: HotStandby commands

2.1.2 Supported SAP liveCache and SAP SCM versions

HotStandby support for SAP MaxDB/SAP liveCache began with version 7.5. Information on SAP liveCache versions belonging to certain SAP SCM versions can be found at: https://service.sap.com/pam.
This case study was done on base of SAP liveCache 7.7 Build 17 and above.

NOTE: The recommended version for AIX is **7.7.07 build 39 and above**. liveCache version 7.7 is supported from SCM 5.1 and SCM 7.0.

2.1.3 SAP liveCache and SAP APO transaction LC10

The position of SAP liveCache in an SAP SCM system, rather than as a SAP MaxDB database introduces some additional complexity for the failover cluster solution. liveCache is controlled from the SAP APO transaction LC10 and is not a standalone like database.

From this transaction it is started, stopped, and initialized. These activities also trigger reports in the SAP APO system which release temporary locks in the SAP liveCache and perform other cleanup/synchronization activities. It is therefore not recommended to start a SAP liveCache instance from the cluster without knowing the status it is expected to be in from the view of the application. Hence the cluster is required to maintain some knowledge of the application status – whether it is in status started or stopped.

SAP APO does provide a type of user exit or hook in the routine which starts and stops SAP liveCache. Having this hook implemented, information on the action being executed by SAP APO is passed to the cluster. If the hook is not found, it is assumed that this is not a HotStandby and no linkage is made with the cluster support.

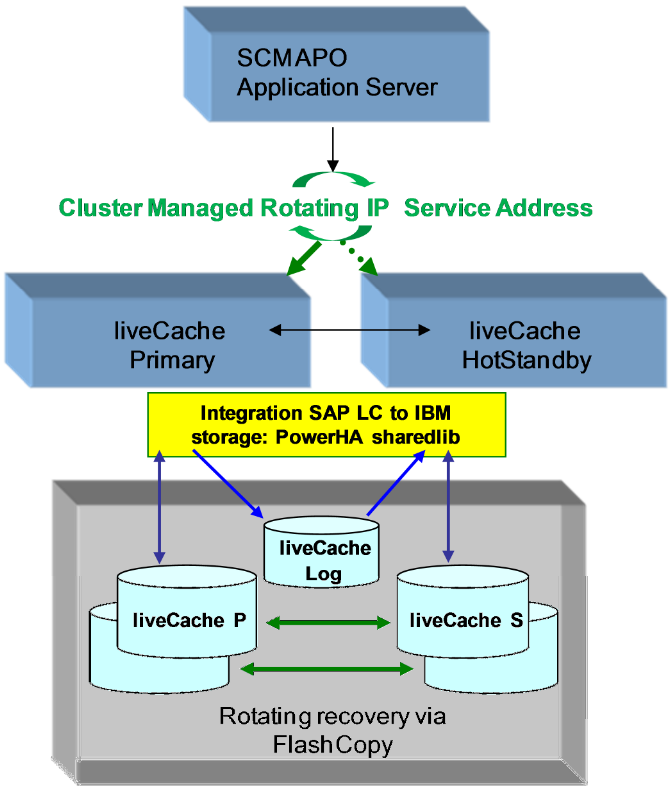
SAP recommends using the installation path */sapdb/<SID>/db*. Normally, if you install SAP liveCache with SAP tools, this will be the default path set during the installation process.

3 IBM HotStandby Implementation Overview

The implementation of HotStandby supported by IBM System Storage is based on IBM FlashCopy functionality. A complete, or clone FlashCopy within a consistency group is used to generate the consistent split mirror of multiple volumes. The initiation of a FlashCopy mirror creates a logical FlashCopy within seconds, which is an independent copy which can be used in read/write mode. The actual copy of the data takes place in the background. The freshly initiated copy can be seen as something similar to a paging space – pages that are being accessed are made available immediately and updates are done directly to the new copy. As a result, the HotStandby can be activated in seconds.

The actual full data copy will complete later, and the duration for this can be in minutes or hours depending on the server type, the speed set for the copy, the amount of data, and the layout of the data on the logical disks (whether serial or parallel FlashCopy paths are used). This activity is asynchronous and transparent to the new HotStandby instance.

If the standby has been offline for some time and the copy of its data is no longer compatible with the current online log, the SAP liveCache will reinitiate the FlashCopy and refresh the standby servers' data.



LC HotStandby Solution based on storage level libraries

Figure 3: Overview of HotStandby with FlashCopy (green) and read/write of shared log (blue)

In the case of a failure, where the standby has become primary, the HotStandby logic will reverse the FlashCopy (FC) direction to bring the failed ex-primary back online as standby as soon as it becomes available. The FC can flow in either direction depending on the roles of the liveCache HotStandby instances. A standby going online will check the state of its data and if the data is stale, will request a new FlashCopy. The date is stale if it does not match the current online log. The FlashCopy will take place between disk pairs that are currently accessible, and being accessed from the servers. The standby will have its volumes open to check for data consistency. The master will be accessing the log, and its data as it is operationally online. For this reason, the data disks are raw disks, or raw logical volumes (without file systems). The concurrently active log is also a raw device as this is being written by one server and read by the other simultaneously.

3.1 Overview of the solution and support

The integration of SAP liveCache with the IBM Storage Servers supports the IBM System Storage DS8000 models, IBM Storewize V7000, and the SAN Volume Controller. The SAN Volume Controller is a storage virtualization solution, such that below the SAN Volume Controller level, any SAN storage can be used.

Figure 4 shows the components of the solution. The libHSS<type>.so is the actual shared library which IBM developed for the API provided by SAP. This library is available as a feature of IBM AIX PowerHA 7.1.1, which also provides the necessary cluster management. The solution package includes the storage connectors which are scripts that interface the library to the specific storage server API.

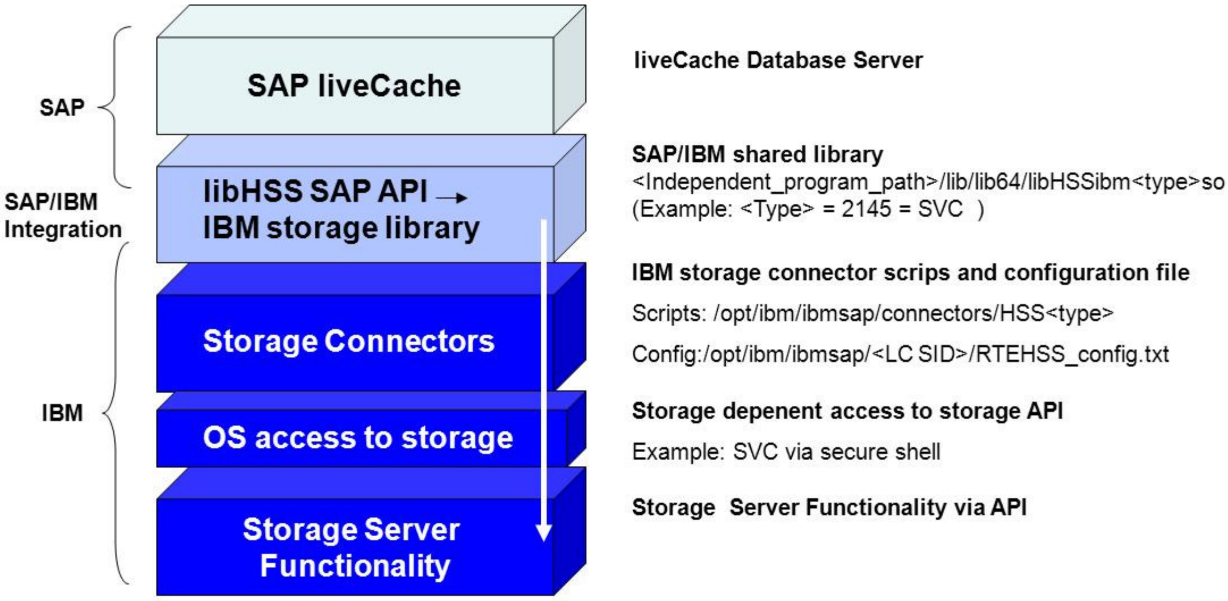


Figure 4: Overview of solution components

The shared library will be installed by the solution in the search path as indicated in Figure 4 to make it available to SAP liveCache. The other paths are default installation paths that are documented here for reference.

3.2 The solution on virtualization

HotStandby can be implemented using directly attached storage, as well as through virtualization in PowerVM. The VIOS provides a virtualization layer for the I/O hardware, allowing the LPARs to share the adapters and I/O paths. This is described in detail in the design document “The Invincible Supply Chain”. See section 13.1on related documents.

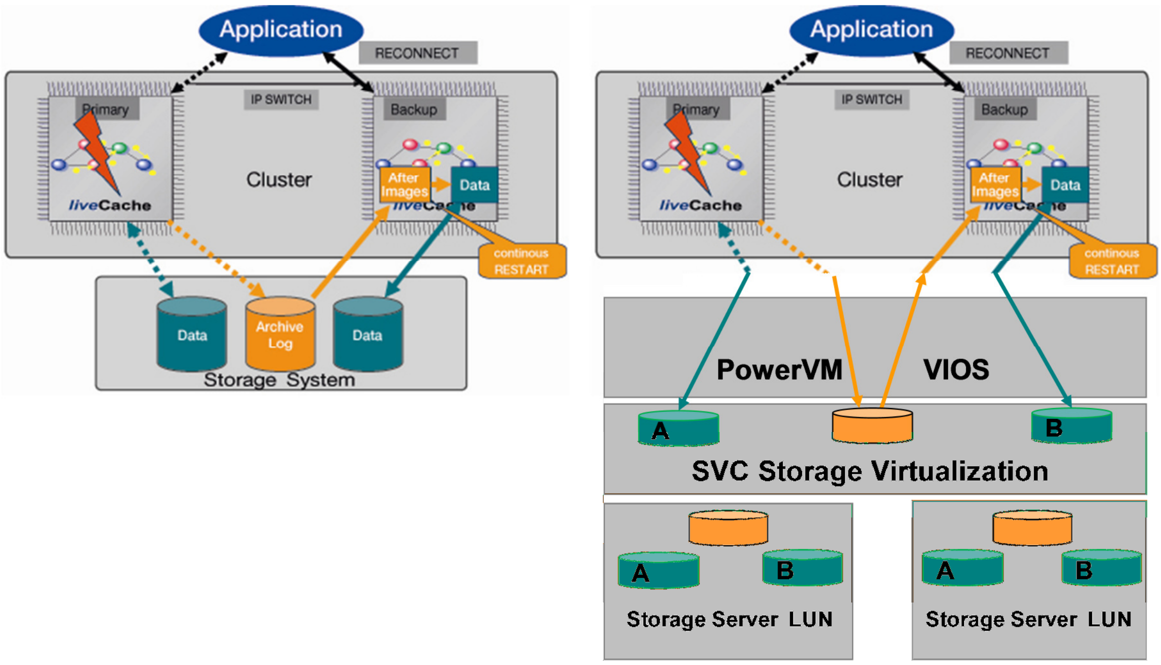


Figure 5: Implementation of virtualization layers

The major benefit for the HA implementation is that redundant I/O paths can be created inexpensively as the components are shared by all the LPARs.

The diagrams in Figure 5 show the original design with directly-attached storage (left) and map this to virtualization (right). In the proof of concept, two levels of virtualization were used in order to ensure that the most-flexible solution was also feasible. In this case, virtual I/O functionality of the server (VIOS) provides the virtualization of the server I/O paths, and the SAN Volume Controller provides the virtualization of the storage.

The integration solution depends on the library being able to issue storage function calls for FlashCopy services to the storage server through the storage API (dscli for direct attached DS8000 or ssh to SVC). The connectors of the libHSS do this, and in order to do this, there must be network connectivity between the LPARs on which the HotStandby solution is running, and the storage server.

In the example in Figure 5 (right), communication paths must exist between the cluster LPARs and SAN Volume Controller, which provides the FlashCopy services for downstream devices, and the SAP liveCache LPARs. The access is through secure shell.

The SAN Volume Controller also provides the wherewithal to mirror the mission-critical data across storage servers. With mirrored storage, the application can survive the loss of a complete storage server without interruption.

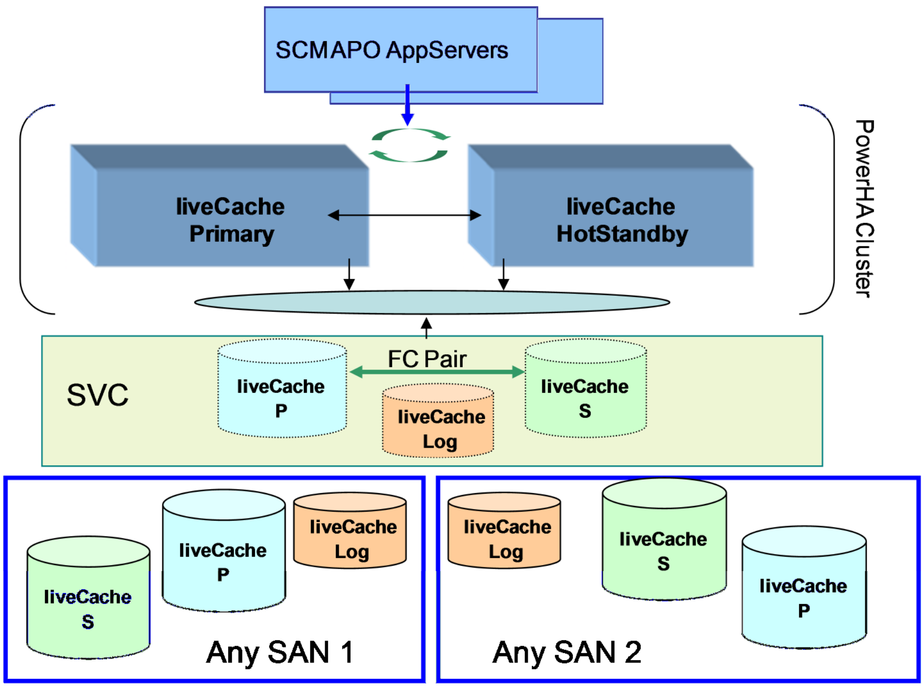


Figure 6: SAP liveCache with HotStandby with SVC mirrored storage

This is a significant benefit for mission critical systems. The flexibility of the virtualization layer in the SVC also makes it easy to provide a non-interruptive storage system migration. The volumes can be moved from one storage pool to another (from one physical server to another) without breaking the FlashCopy relationship or disturbing the cluster solution and hence is transparent to the application.

3.3 Design of the PowerHA Cluster for SAP liveCache HotStandby

PowerHA 7.1.1 differences to earlier versions such as the CAA repository disk and multicast heart-beat are discussed in section 5.4.1. In the following, the general design decisions are explained. The focus is on the difference between the infrastructure and the service and the start, the stop and the monitor logic. Figure 7 shows an example design for PowerHA version 7.1 and above.

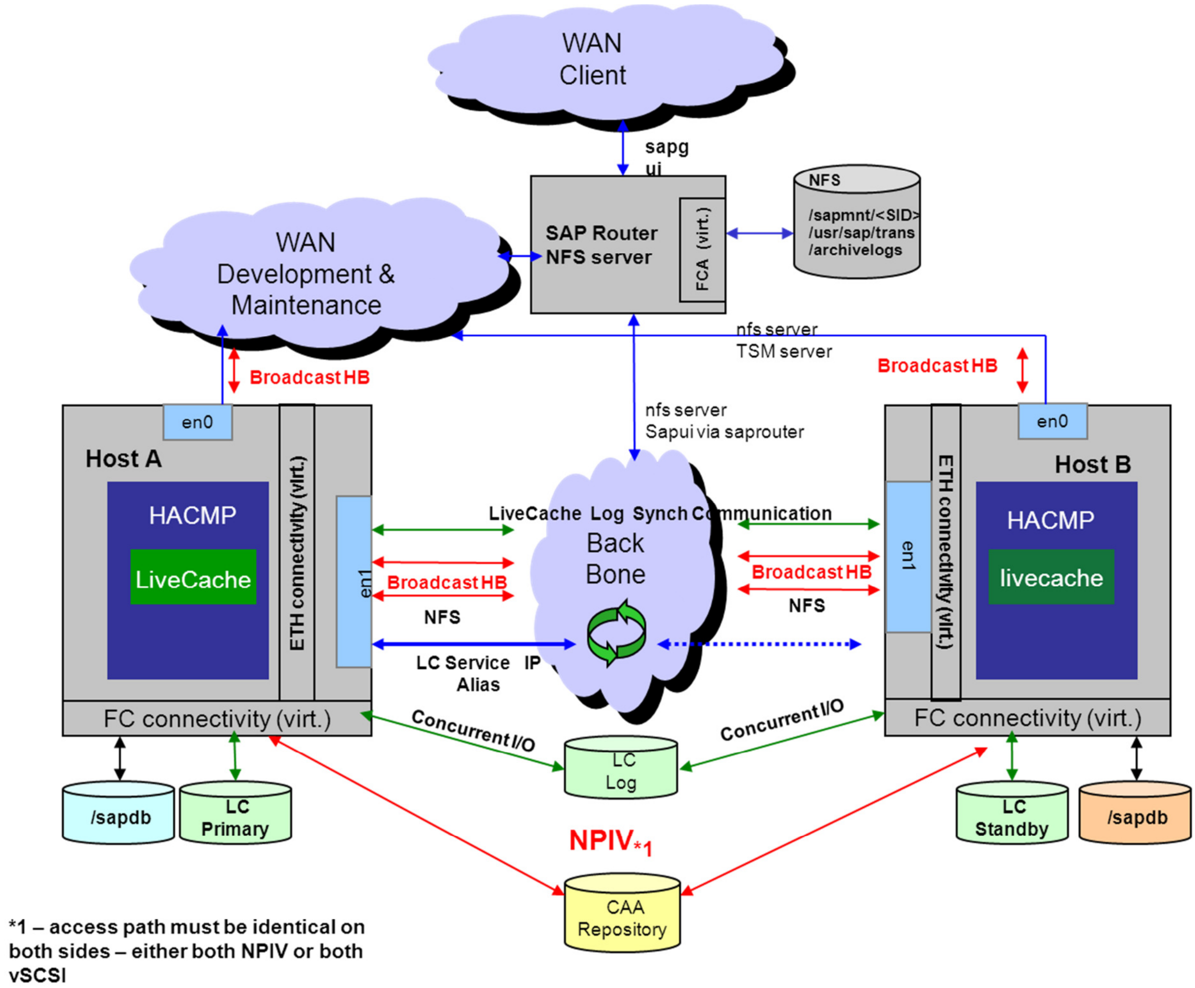


Figure 7: Overview of cluster concept used in reference architecture for proof of concept

3.4 Definition of infrastructure and service in SAP liveCache cluster design

The PowerHA design for SAP liveCache separates the application as a service under control of the corresponding SAP APO and the required infrastructure as seen from PowerHA. The infrastructure consists of the resources combined into the resource groups which need to be made available as pre-requisites for APO to be able to start the liveCache as a service. The liveCache as a service is started and stopped by SAP APO and not directly by the cluster functionality. The reason for this differentiation is the need for SAP APO to administer the SAP liveCache and keep synchronization between APO and liveCache.

Infrastructure:
Refers to the resources brought online by starting the cluster in preparation for SAP APO to start the service. It includes the volume groups for the shared log, service IP (OFFICIAL_NODE), PowerHA application servers/monitors and x_server process.

Service:

Refers to the online SAP liveCache service as an active database, whether the SAP liveCache is online or offline according to the actions taken by the SAP APO administrator.

This separation of power results in the following two categories of control:

- Administrative cluster tasks such as starting, stopping, and moving the service provided by SAP liveCache and its required infrastructure.
- Failures of SAP liveCache as a service and/or failure of the server nodes that result in the automatic recovery provided through the implemented cluster.

High-availability considerations for network and storage related components are covered by the infrastructure design of the master document and not here. The redundant hardware infrastructure is not specific to HotStandby but typical in an HA implementation.

3.5 Infrastructure design – PowerHA

Figure 8 shows the PowerHA resource groups used to support the SAP liveCache cluster design. The configuration consists of three resource groups. RG_Log_<SID> – this resource group is online on both nodes and contains the shared SAP liveCache LOG volume group. It must be brought online by PowerHA as it relies on the cluster functionality for concurrent access. It is also vital to have it to start the instance. It is vital to write the logs. In case this cannot be done anymore an instant failover needs to take place. To ensure the early trigger an application monitor is written.

NOTE: The DATA volumes are brought online on each node automatically at OS level as they require no special treatment.

RG_Master_<SID> – this resource group consists of the service IP alias used as the service address for SAP liveCache and the master application monitor. This resource group is only active on one of the nodes at any time – never on both. The liveCache service IP and the master monitor are rotating resources.

RG_Standby_<SID> – this resource group consists of the standby application monitor. It is only active on one node at any time different to the RG_Master_<SID>.

Both the standby (slave) and the master have application monitors. The monitors have different actions depending on the role of the application that they are monitoring. The master monitor attempts to keep the SAP liveCache in the ONLINE state. If the master is not ONLINE anymore its memory structure is lost. To maintain the data this resource group will be moved instantly to the second node in case it is no longer in ONLINE state. These results in making the standby become the primary (master). The standby monitor tries to maintain an instance in the standby mode on the local node by restart attempts in case of a failure. This is of course only done once the SAP APO has requested the SAP liveCache service to be started. In case a standby is in ADMIN mode the monitor accepts this state as well. For a MASTER instance ADMIN during runtime is no valid state. Only while the instance starts ADMIN is a valid state tolerated by the implemented logic.

In general, during a takeover, both resource groups switch nodes. In case of a one node situation the standby resource group will be put into “Offline due to lack of node” mode. Action is then triggered to change the SAP liveCache instance status from STANDBY to ONLINE (RG_Master_<SID>).

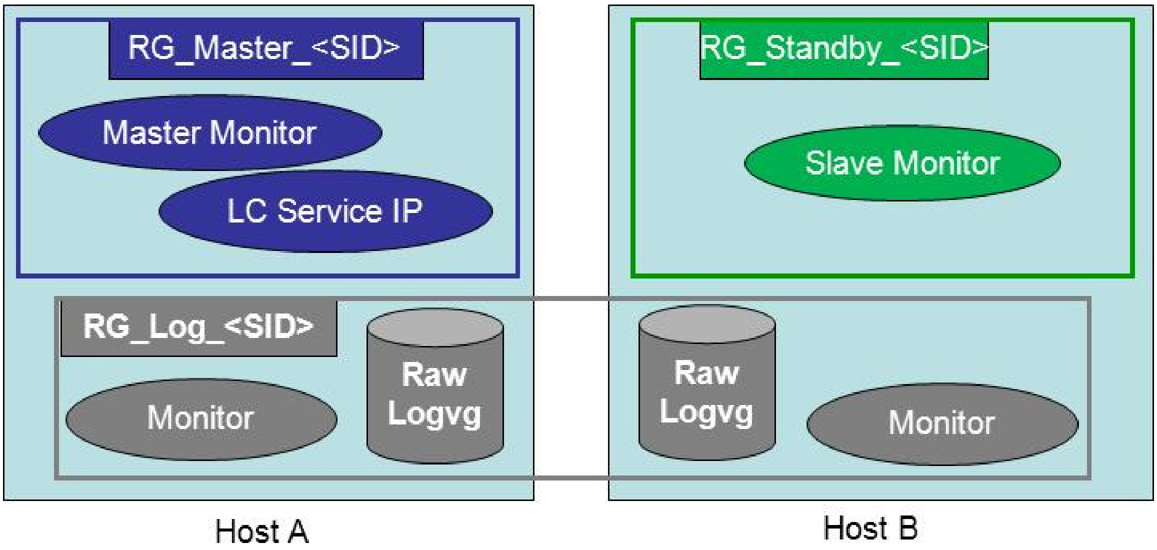


Figure 8: Overview of the PowerHA resource groups

View of resource groups in an active two node cluster
The *clRGinfo* command displays the status of the resource group infrastructure (the cluster related states). This does not mean the SAP liveCache service is online as this depends on the SAP APO application status. The cluster differentiates between the starting of the cluster, and the starting of the SAP liveCache service. The active cluster and the online resource groups are the prerequisite for starting the SAP liveCache service.

clRGinfo

Group name	*Group state*	*Node*
RG_Log_<SID>	*ONLINE*	*is03d6*
	ONLINE	*is04d6*
RG_Master_<SID>	*ONLINE*	*is03d6*
	OFFLINE	*is04d6*
RG_Standby_<SID>	*ONLINE*	*is04d6*
	OFFLINE	*is03d6*

The current implementation design also requires a reliable NFS location for control information between APO and PowerHA and within PowerHA which influences the behavior of the cluster. More information in section 5.4.6.

3.5.1 SAP liveCache service

The responsibility for the status of the SAP liveCache database service is with the application. SAP liveCache is a component of SAP APO, and integrated into the transaction LC10. From this transaction, the status of SAP liveCache is controlled. It is started, stopped, and initialized. These actions can lead to synchronization actions within the SAP APO application itself which are important for data consistency. For this reason, the cluster does not have the logical authority to start and stop SAP liveCache as a service according to the state of the cluster. The cluster must retain knowledge of the status of SAP liveCache, as set by the application, and manage the cluster activities accordingly.

SAP APO provides a hook in the path from LC10 to SAP liveCache that can be used to provide information on the ongoing action to the cluster.

The PowerHA cluster uses this hook to receive and maintain status information on the status expected by the application. When the cluster is started, this information is consulted (yellow line) and depending on the expected status, the monitors are set to active or passive mode. This prevents the cluster from starting a stopped liveCache instance. The SAP liveCache is brought online or not, according to the status of the last SAP APO request. The SAP APO request status is maintained in a highly available location shared by both the nodes. It is referred to as the lock directory later. In this prove of concept a highly available NFS file system was used. The access to this state data is from the start and monitor scripts configured in the PowerHA application servers is read only except for lock initialization indicated by the yellow path.

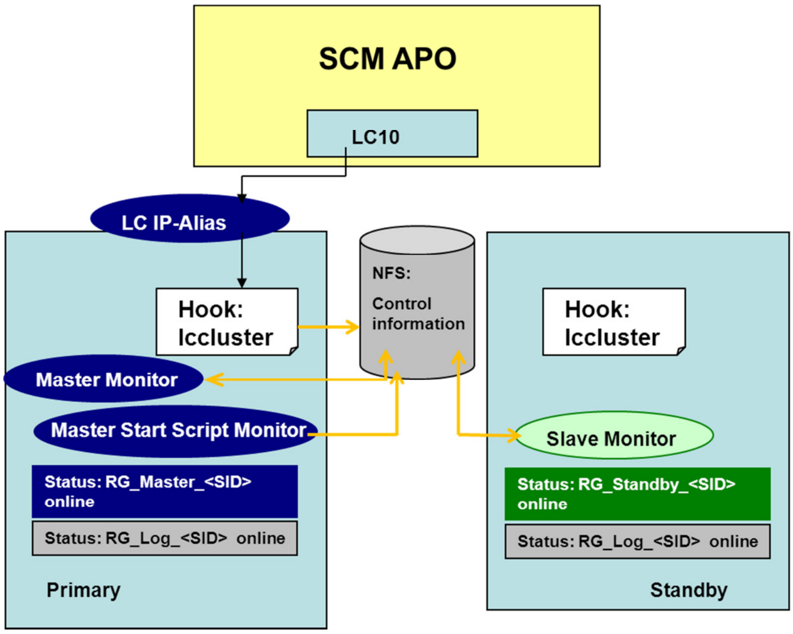
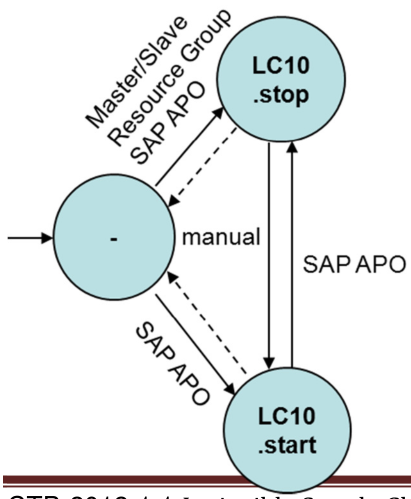


Figure 9: SAP APO control of SAP liveCache service

3.6 Start, Stop and cleanup logic for the service and the infrastructure

The design objectives for the startup of a master or standby instance and how control through APO is maintained is described in the following sections. Furthermore, the different flavors of the ADMIN state are described along with a summary to what extend the cluster reacts to it and the part which has to be covered by given procedures. Finally, the cleanup procedure for stopping/starting a cluster is described.

3.6.1 Starting a clustered SAP liveCache from SAP APO



The diagram in this section show the logic used by the cluster to start the SAP liveCache service for the master and the standby instance. To fulfill special needs of the SAP liveCache application, the logic needs to maintain state of the SAP liveCache as set by the SAP APO application to determine whether the service should be started or not. The application status is maintained in the shared lock directory. The location of the lock directory is any valid reliable NFS folder which is an input parameter for the PowerHA automation framework later.

If the cluster is being restarted, the administrator may need to reset the lock status to achieve the action that is wanted. The cluster will start the SAP liveCache service depending

on the status of the LC10 information located in the lock directory. The LC10 status can be either empty, LC10.start or LC10.stop.
The SAP APO LC10 status is the result of the last application request set by the cluster control hook. If there is no LC10 status information, the cluster assumes no start request from the application has been received and will not start the SAP liveCache database instance. But the script will initialize the state to LC10.stop. The SAP liveCache database instance is only started if the LC10 status is LC10.start.

NOTE: If a cluster is to be started having LC10.start, the cluster administrator must validate with the APO administrator whether he should manually reset this application state (indicated by the dotted lines). This application state formally belongs to the APO administrator.
Figure 10: Logic of the SAP APO PowerHA integration

3.6.2 Cluster control hook for APO:

The "hook" is provided through the SAP script LCINIT which invokes a cluster script "lccluster". This hook is used to give APO full control of starting/stopping liveCache. APO can access the liveCache only after the PowerHA cluster has been started as the cluster starts the liveCache x_server – the liveCache listener, and assigns the service IP alias used for communication between APO and the liveCache.

The hook is invoked twice for every action performed by APO – once as a request, and once to indicate as success or failure of the action. The logic differentiates between the action type (starting or stopping) and reacts accordingly.

The startup sequence:
Step 1 request:
APO indicates to PowerHA that it is about to start liveCache. This allows the cluster to prepare for the upcoming start request.
Step 2: APO will actually start the MASTER instance via transaction LC10 and the SAP script, LCINIT.
Step 3 liveCache start completed successfully:
Now the earlier mentioned lock LC10.start is set to activate the PowerHA application monitors, and the activity to start the standby is triggered. The standby startup requires the listener to be up on both nodes; otherwise the resource group of the standby itself will take the required actions.
The stop sequence:
Step 1 request:
APO indicates to PowerHA that it is about to stop liveCache.
This allows the cluster to prepare for the upcoming stop request.
Step 2:
APO will actually stop the MASTER instance via transaction LC10 and the SAP script LCINIT.
Step 3 liveCache stop completed successfully:
The standby will be stopped as a cleanup task

Figure 11 shows the tasks done by APO in yellow and the tasks performed by lccluster in blue.

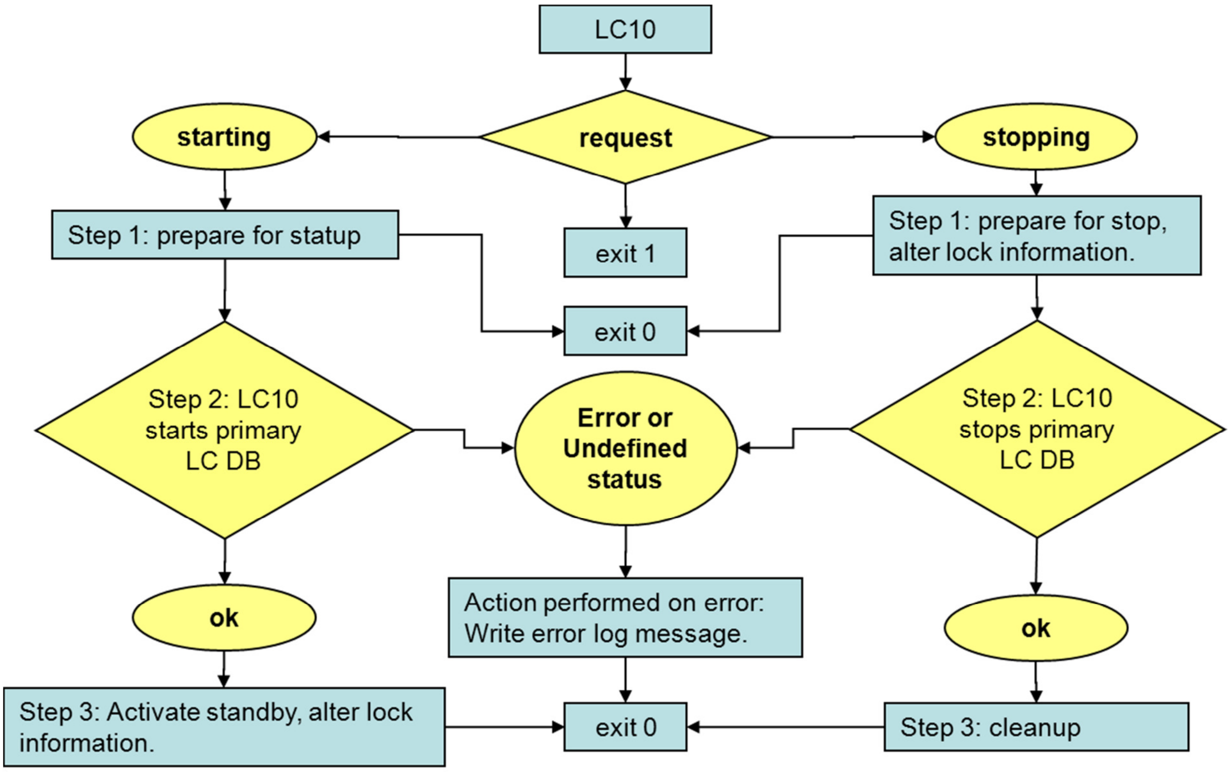


Figure 11: Logic of APO to PowerHA integration

3.6.3 Cluster Start Scripts

The two logic flows – master and slave – of the start script are executed when the cluster is started as part of the resource group initialization and later they are also used during a recovery or a move. In case of a recovery from "Unmanaged" after a maintenance window the start scripts will be called in case the highest priority monitor does not indicate running instance. They are not used if the start/stop request is triggered by APO.

The first action for both instances is to verify the application status, set by SAP APO to determine whether the cluster has or does not have the authority to start the service. This status is referenced to determine what action it should take as the result of a cluster (re)start, or a cluster failover. If the status does not exist, the start script assumes that no application direction has been given by SAP APO and the service will not be started. The script will prepare the cluster for a SAP liveCache startup request that will be expected to come later from the SAP APO administrator, via transaction LC10.

The cluster start script is executed once for either a startup or a failover to activate the SAP liveCache instances. If the initial startup of the master SAP liveCache fails on the current node, it attempts a failover if the second node is active. The default is to continuously try to bring the master online by means of a failover as long as there is a second node available. This default action is currently implemented, based on the internal variable RC_FOR_FAILED_MASTER_db_online set to 0.

NOTE: The master resource group's monitor itself is configured not to attempt a restart and never to fall back. In case only one node is available, and the db_online fails on this node, the cluster will log an error situation and stop processing. It will not continue to try to bring up the master repeatedly on the same node. For this situation a notification method should be enabled to inform the administrator instantly that the SAP liveCache database cannot be brought online.

For certain environments it could make sense to exit the start script with an error (return code of 1). PowerHA reacts on this return code stopping further actions and wait for manual interaction rather than attempting further recovery. To handle this, a notification method should be enabled in PowerHA to inform the administrator instantly.

The service start of the standby will fail until the Master is ONLINE. To reflect this design the configuration of the standby's application monitor has to be chosen carefully (section 9.6). The application runtime monitor of the standby is then activated every few seconds (configurable in PowerHA) so that after few seconds the application runtime monitor will detect the missing standby and initiate the successful recovery as soon the master is ONLINE.

In case of the node being the standby, the script intentionally always returns with success in regards to cluster processing. This is necessary to continue with cluster processing even if the activation of the standby service failed. Otherwise the complete cluster processing – including the master - will stop with error and will require manual recovery. This logic takes into consideration the fact that the slave may have failed, but the SAP liveCache can continue without a standby and therefore the cluster processing is not stopped. The standby monitor will trigger the recovery of the slave. If no master instance can be detected elsewhere or on this node the master is ONLINE the activation of the STANDBY is rejected.

NOTE: if the restart of a slave are used up and the standby goes into "Offline due to lack of node" a failover of the RG_Master_<SID> will be a cold restart. To prevent from that case the following tasks have to be done: Implement a notification method in case the slave has failed to inform the administrators in time and add sufficient monitor cycles to survive roundabout 30-60 minutes.

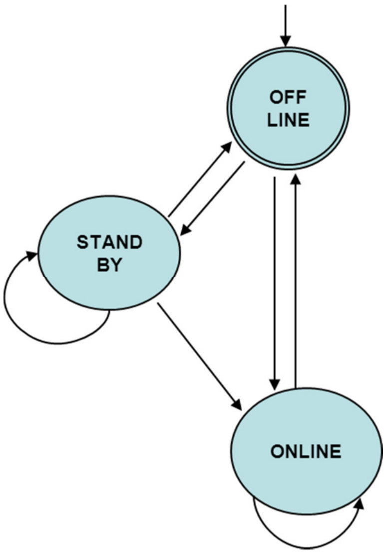


Figure 12 shows the state changes between master and slave database status. If the action is the result of a failover of an active database, the db_state at failover will be **standby** and the database will be brought **online** as master. If this is an initial start, the status of the database will be **offline** and the database will be brought **online** as master and the slave as **standby**. An **online** master or **registered standby** instance is untouched.

If the cluster or a resource group is stopped or the application requests SAP liveCache shutdown the database (slave and master) is also returned back to **offline** state. This figure does not cover the **ADMIN** state.

Figure 12: Database state transition without ADMIN

3.7 Stopping and clean-up of the SAP liveCache service cluster

Stop and cleanup is split into two pieces. Stop is the action performed when the master or standby resource group is stopped. Cleanup is performed when the resource group with the shared log volume is activated or deactivated.

The stop scripts of the two instances behave differently regarding the state of the instance – master or slave. It is used in failover recovery and cluster shutdown. A failing master is taken offline whereas the standby is left in standby status to complete any open failover transitions. If the standby were to be taken offline, the data cache memory structure would be deactivated and would need to be rebuilt on restart. The objective of the HotStandby implementation is to avoid this overhead when transitioning the standby to master status. The standby should simply change status and assume the role of the master SAP liveCache.

The final cleanup is driven by the RG_Log_<SID> resource group. When the log volume is being taken offline, all instances and database listeners will have to be stopped. This is true especially for the standby resource group. It does not take any clean up action (by design), but relies on the cleanup of the prerequisite resource group (the log) to ensure the consistent cleanup of the service. This avoids any type of erroneous recovery attempt being triggered.

At cluster startup, the start script of the RG_Log_<SID> ensures that the lock directory is cleaned, the listeners are activated and the concurrent access volumes of the log are online on both nodes.

3.8 Handling of ADMIN states

The ADMIN state can be encountered on an instance moving from between active and deactive states (ONLINE and OFFLINE) on a primary server, but never on a Standby directly. The SAP application does not allow a liveCache instance in STANDBY to be moved into ADMIN mode. The DB state changes are assessed and managed by the application monitors for the standby and by the start script of the master. These scripts must interpret the state changes and determine if and when these represent a failure situation.

Interim ADMIN state or long ADMIN state

The interim ADMIN state is part of a normal start up, but here usually only a short period. In the case of a master failover, the recovery time necessary for the standby to complete all recovery activity and catch up to the last commit can result in a longer ADMIN state.
The scripts tolerate a long admin state under these circumstances. Admin state is tolerated until the instance has achieved ONLINE.

Admin state for Maintenance

Once the master instance has been brought into ONLINE state, a return to ADMIN is interpreted as a failure. The objective of the cluster logic is to maintain an ONLINE master liveCache from the time APO has started the liveCache and until APO stops the liveCache. Therefore any attempt to put the master instance into ADMIN state, which is not coordinated with APO, will result in a failover to the standby and a new master in ONLINE state. The proper method of entering a maintenance procedure is to shutdown the HotStandby from APO.
Maintenance procedures are described in the administration guide – see section 13.1.

3.9 Monitor logic of the master's application monitor

To address special monitoring needs of the master the logic is split into a startup monitor and a runtime monitor executed as soon the cluster resource group is started/moved. The behavior of the scripts differs on base whether APO has started the application or not. PowerHA triggers the startup monitor before the start script for the instance is called. The logic inside ensures the start script is called no matter if the application is started or not. The way how to handle the current state of the instance is decided inside the start script. A valid instance will not be restarted. The startup is then monitored inside the start script on base of a new PowerHA feature. The runtime Monitor is called on completion of the start script. By this design the runtime monitor reacts to failures during startup and failures during runtime. This means the cluster values the application availability as the highest good and does everything to continue operation.

As mentioned earlier the startup sequence is monitored inside the start script by the new PowerHA feature to run a script in the so called "foreground" mode. As soon the master instance has reached the state ONLINE or the startup failed the logic hands over to runtime monitoring. From now onwards only the state ONLINE is valid for a running master instance. The logic also handles the different types of ADMIN state accordingly. The two types are the so called long ADMIN where the database has to redo logs before it can reach the ONLINE state. The other case is a database in ADMIN state what is a kind of error in handling. The long ADMIN will be handled in the start script. The start script will observe this state as long as necessary. The logic also reacts immediately in case the state changes to something different to ONLINE or breaks during ADMIN state to take corrective actions. In case the invalid ADMIN state is hit the logic will enforce to establish the master instance to provide the requested service.

3.10 Monitor logic and of the standby's application monitor

For the standby a pure runtime monitor is used. The check performed is a local database check for the states STANDBY or ADMIN.
PowerHA will also use this monitor as a startup monitor. To give the instance some startup time the logic allows the application 10 seconds to reach the state beyond the OFFLINE or UNKNOWN states. OFFLINE is the expected initial state of the instance, and UNKOWN is a brief transient state during startup.

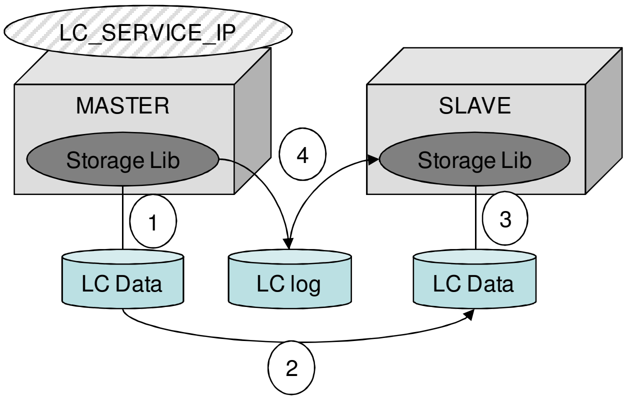
NOTE: If this is not sufficient for a certain landscape the PowerHA internal variable TIMEOUT [lc_wait_long_in_startup] can be increased.

3.11 The functionality of the storage library with a clustered liveCache

The interaction of liveCache and the storage library in sense of the FlashCopy volume pairs and the concurrent log volume can be described in three phases.

Phase 1: Start the HotStandby relationship

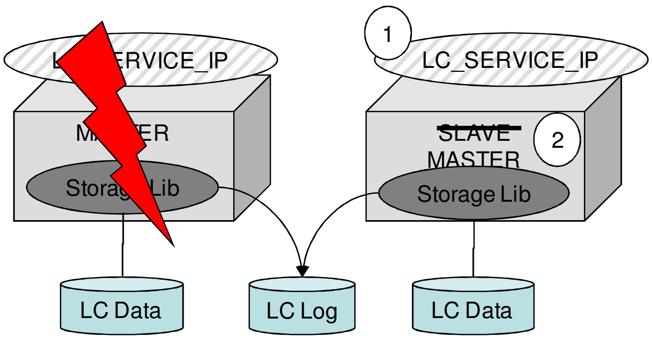
The primary instance – the master – is started by APO. APO selects the correct instance by accessing it always over the service IP alias (LC_SERVICE_IP).



1. The memory structure of the primary instance is created by reading the persistent data from disk.
2. The implemented logic of PowerHA triggers via the master instance the startup of the standby instance. This activates the FlashCopy which then initializes the persistent data of the standby. The FlashCopy will be withdrawn as soon the logical copy process is done. The physical copy then takes place in the background.
3. The Standby liveCache memory structure is created on base of its own persistent data.
4. The application ensures that only the master can write to the concurrent log and the slave reads it (log record shipping).

Phase 2: Failover of primary

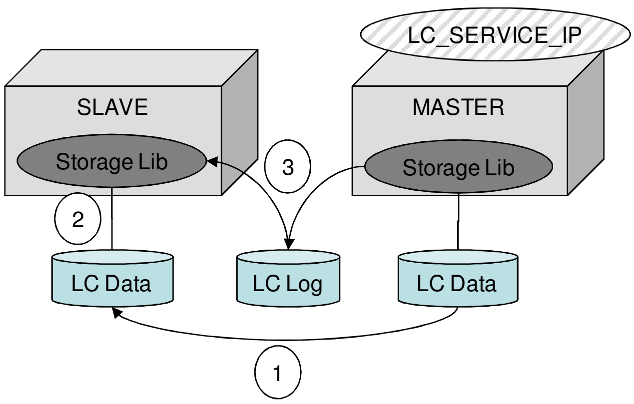
If the master instance is not operational a failover is triggered by the cluster logic.



1. The communication daemon of the liveCache (x_server) is stopped on the next hop node. Then the service IP is moved to the new node along with the master resource group. The library ensures that no second failover can trigger a FlashCopy unless all physical data has been written in the background to ensure data consistency.
2. The still active standby instance reads all outstanding entries from the shared log and is activated as master. Then the x_server is restarted.

Phase 3: Reactivation of a standby

In case a second cluster node is or becomes available the cluster logic restarts a standby instance.



1. The implemented PowerHA logic triggers the startup of the standby. This again starts a FlashCopy. In case the copy from Phase2 is not completed the library ensures data consistency by preventing a reversal of the FlashCopy until the currently running copy finishes.
2. The standby liveCache memory structure is created on base of its own persistent data.
3. The application ensures that only the master can write to the concurrent log and the slave reads it (log record shipping).

4 Storage Prerequisites and best practices for storage planning and design

In designing a HotStandby configuration, the following storage-level recommendations should be considered.

4.1 PowerHA SystemMirror Supported Disk configurations

PowerHA SystemMirror's SAP liveCache HotStandby solution has been tested for large number of disks, volume groups and volumes. The following configuration options are supported:

Logical Volume Size	Up to 128 GB
Number of Disks/Volume Group	Up to 6
Number of Volumes/Volume Group	Up to 12
Number of Volume Groups in Log or Data Volume Groups activated between Master and Standby	Up to 3

Refer to SAP documentation for any relevant limits on Disk and Volume sizes that SAP recommends.

SAP liveCache support individual raw device sizes up to 128GB. In the typical layout, a raw device is represented by a raw logical volume. Therefore the physical disks can be greater than 128GB and broken down into multiple logical volumes, not to exceed 12 logical volumes per volume group.

4.2 liveCache executable and work directories

liveCache has additional storage requirements in regards to the local storage for liveCache executable, the instance administration data and work directories. All of this data is normally in the file system */sapdb*. This data is not duplicated or shared between the two nodes but exist separately on each and is relevant to the individual node. This data includes the kernel messages, history information and liveCache error information pertinent to the node and its local history. This file system should be implemented on a volume group external to the root volume group. In the proof of concept, a separate volume group – on its own storage volume – is created on each node dedicated to */sapdb*.

4.3 Disk requirements for SCM APO HotStandby Design

Below is an example of a storage layout. This includes the data disks, the log disk(s), the repository disk, and local storage required for liveCache node related data (executable, runlogs, and error logs) and in this instance, space for archive logs which may be backed up periodically to a storage medium. The archive log directories are local for performance purposes in this design. The archive logs can also be archived to an NFS shared file system.

Explanation and use of volumes in the following two diagrams:
A & B: concurrent access raw devices – zone to both nodes. A is the PowerHA repository disk and B is the liveCache concurrent log.

C - G are the data disks for node1 and zoned only to node1. They have no file systems but maybe broken down in multiple raw logical volumes. The assuming data size is for liveCache on disk is < 250 GB for this example.
H - L are the data disks for node2 and are zoned only to node 2.
C-G/F-H are the flash copy pairs – they must match exactly in size with their counterpart.

O & P have been added as a possible solution for the intermediate log archive. These would be expected to be JFS2 and locally mounted.

M & N are also JFS2 each locally mounted. These are the runtime executable and local data for each liveCache instance. /sapdb must be available for the installation of the liveCache software.

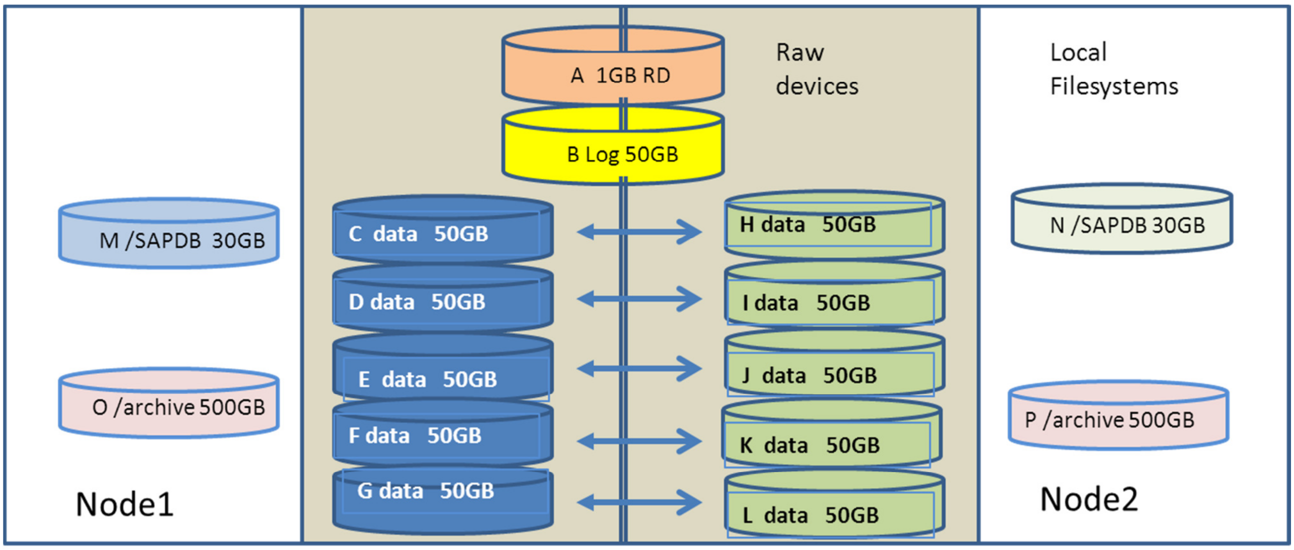


Figure 13: Example of a data layout for liveCache

NOTE: In this design, the log LUN is relatively large. This LUN can contain multiple logical volumes which will be seen by the liveCache as multiple raw log devices, or it can be a single LV. Within liveCache, the log can be further partitioned into log partitions. In this design, the single LV is partitioned within liveCache into two log partitions of 25GBs.

The storage layout in Figure 14 depicts the same design example implemented on a SAN Volume Controller stretched cluster. The intention of this design is to maintain copies of each LUN in both sites such that the loss of a storage server, or a portion of the SVC cluster, will remain transparent at application level.

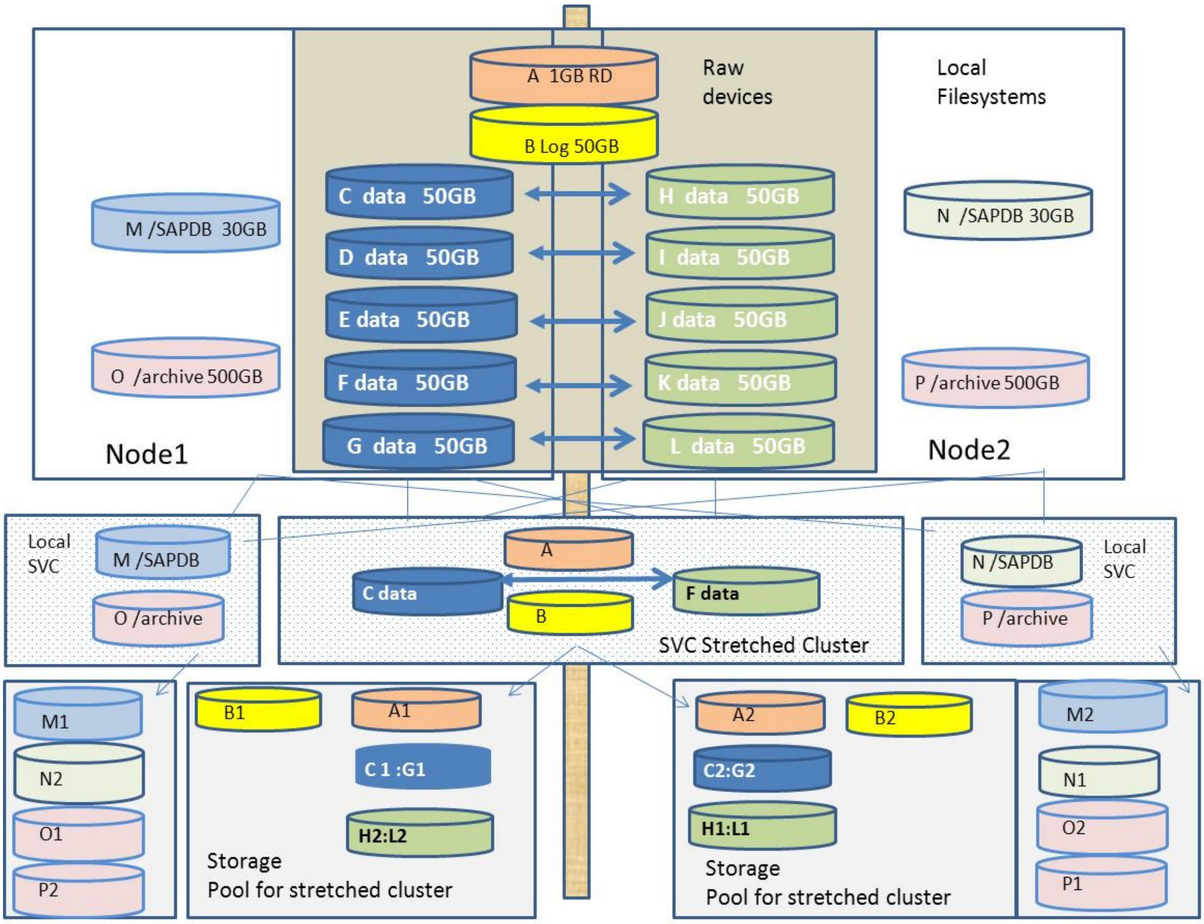


Figure 14: Example of storage design and volume group layout

4.4 Considerations for the layout design for data volumes

There is one situation in which the initiation of a standby liveCache can be delayed. Consider there is a large liveCache which has initiated a FlashCopy from the primary (P1) to the secondary (S1) and then a failure occurs which causes a failover to S1. S1 is now the primary and before the FlashCopy background copy has completed from P1 to S1, P1 is available to become a standby. The reverse FlashCopy cannot be done before the current copy is completed. All the data is not yet copied to S1, the current active liveCache. The block of starting the reverse FlashCopy which would destroy the data is implemented in the libHSS storage library. As soon as the FlashCopy has completed, the reverse FlashCopy will be initiated and P1 will be brought online as standby. However, this process must wait until the running copy is finished.

To address this very small window of exposure, it is recommended to split the data across more than one volume and thereby increase the number of parallel FlashCopy paths to speed up the data transfer. The amount of parallel running pairs is limited by the maximum MB/s transfer rate per running FlashCopy what is today around 50 MB/s enabling the copy rate to 100%. The overall limit of all active flash copy pairs is today around 500 MB/s. Please verify accordingly to the used model.

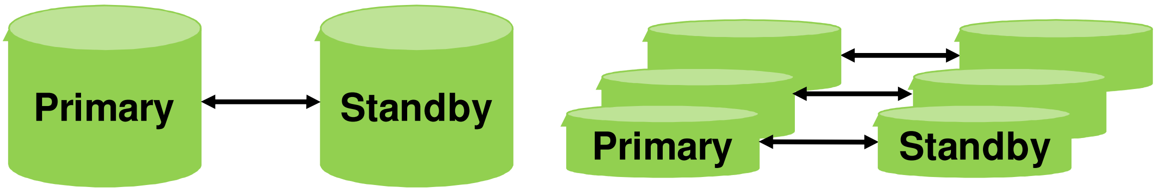


Figure 15: FlashCopy disk design – few vs. many small

4.5 Size consideration for log volumes

An additional consideration is the relative size of the log volume. A standby will consider its data stale if it cannot find the next outstanding sequence number in the current online log. This will result in the initiation of a new FlashCopy to refresh the data. This occurs only when a standby is being brought online and it will only be stale if it has been offline long enough for a log switch to have taken place (in liveCache, the log will have been archived and overwritten with the new data). The larger the log volume, the more likely it will be that the liveCache standby will find the log record sequence in the active log.

Again this could help in the situation described above where a failover takes place, and then the failed server comes quickly back online and is available to become a standby, but must wait for completion of a running FlashCopy. If the data is not stale, it need not wait and it need not refresh its data. In this case, the standby can come online, even though the FlashCopy is still running from the primary to the secondary.

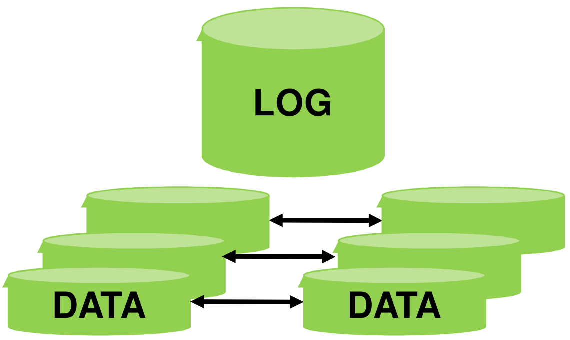


Figure 16: Log size

5 Installation Path: SAP liveCache HotStandby with PowerHA

The implementation of the integrated solution for SAP liveCache HotStandby, as shown in Figure 17, is covered in this section. This includes the storage setup to support the cluster, the installation of the SAP liveCache as necessary to achieve a proper HotStandby configuration using IBM System Storage, the PowerHA failover cluster solution and the integration into APO.

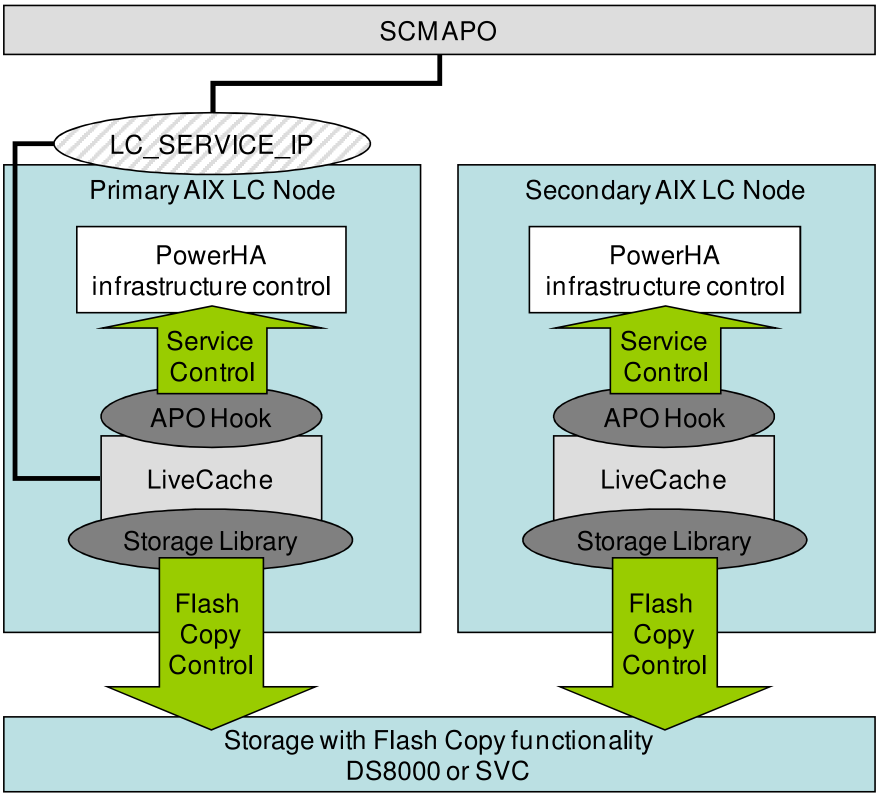


Figure 17: Stack overview for liveCache

There are two methods which can be used to implement the HotStandby with PowerHA 7.1.1 SP1:

- A manual approach to provide additional information

  This method is documented in the case that the more comfortable solution with the PowerHA Wizard and Smart Assist is not able to manage some aspect of the proposed design. As this is a new product, we cannot be sure that we have foreseen all possible permutations which might result from a given installation.

- A PowerHA 7.1.1 automated setup based on the Wizard and Smart Assist agent for liveCache.

5.1 Overview - Installation path using Smart Assist and Wizard

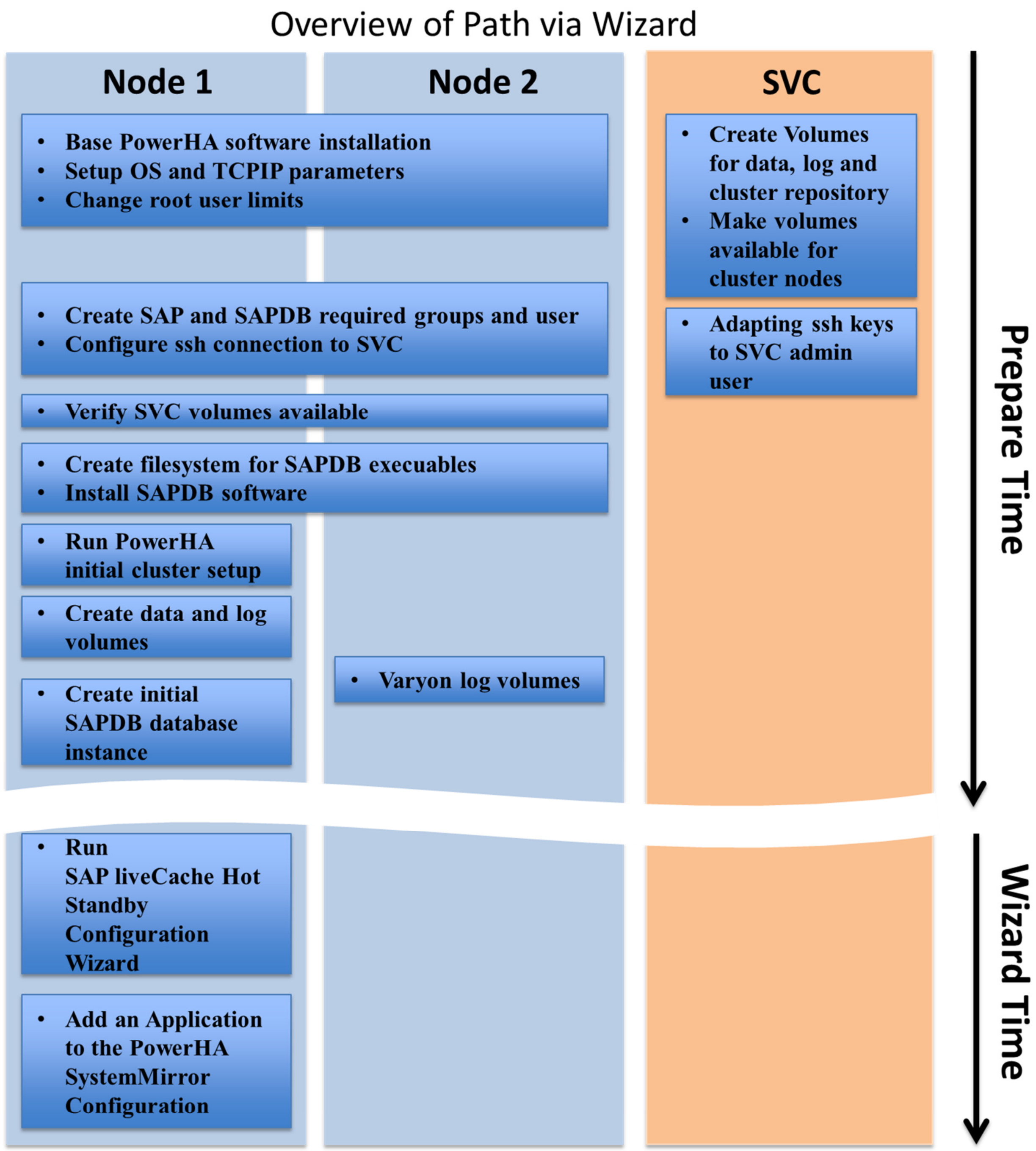


Figure 18: Timeline of implementation path using Smart Assist and Wizard

The diagram in Figure 18 shows an overview of the steps according to the recommended sequence. The high level flow when exploiting PowerHA 7.1.1 SP1 is described in the following sections. PowerHA provides two parts with the solution. The first part is the Wizard helping with the setup of the liveCache environment itself. The second is the Smart Assist agent configuring the cluster. Both can be used independently from each other.

1. Prepare (section 5.3.1 - section 7 keep to the flow of the document)
2. Use Wizard for configuring the HSS Storage library and FlashCopy relationship. Use the smit menu: SAP liveCache Hot Standby Configuration Wizard. (section 8)
3. Setup LC_XUSER environment variable for root and SAPDB xuser settings on both cluster nodes (section 6.3.3).
4. Use Smart Assist to configure the cluster setup for the HotStandby solution. Use the smit menu: Add an Application to the PowerHA SystemMirror Configuration (section 0)

Additional Consideration and Information:

- Naming conventions given by the framework must not be changed.
- At least during the time running of the wizard the SVC user for the ssh configuration to connect to the storage must be “admin”. If this user is not available a manual path is provided in the next section.
- The amount and size of hdisks, logical volumes and volume groups are limited (section 4)
- The wizard only allows one HMC to access the storage system to be used during the automation. For redundancy a second can be manually added after the wizard has been run (section 0)
- Avoid upgrading the liveCache before using the Wizard if a new installation is done.
- The wizard only functions in a two-node configuration.
- The automation works only for SVC based storage attachment through ssh. For DS8000 no automation is provided. Manual setup for DS8000 models is provided through PowerHA documentation.
- The wizard cannot run on vscsi virtualized disks. This will require manual implementation of the tasks the wizard would automate (section 0). After these steps the Smart Assist should be used to continue.
- Prepare a notification method to be included into the PowerHA application monitors.

5.2 Overview - Installation path overview for a manual approach

Figure 19 shows the sequence of steps which need to be taken for a manual installation. The sequence is not mandatory but represents a logical flow. The actual sequence will depend on the starting point, whether this is a new installation or a conversion.

1. Prepare (section 5.3.1 - section 7) incl. sections marked as [Manual] keep to the flow of the document)
2. Wizard or Smart Assist should be used if possible (section 8)
3. Continue with the outstanding tasks from section 9.1 onwards.

NOTE: Resource group naming conventions given by the framework must be used. See section 9.5.

Figure 19: Overview of the manual installation path

5.3 General preparation tasks overview

1) Install PowerHA software on both nodes – see section 5.3.1
2) Setting up specific OS and TCPIP parameters on both node – see section 0
3) Set root user limits to unlimited – see section 5.3.3

4) Create SAP and SAPDB required OS groups and OS users on both nodes – see section 5.3.4
5) Create ssh connection for SAPDB OS user (sdb) – see section 7.
6) Review disk prerequisites and prepare storage – see section 5.4.4.
   On both nodes: Create Volumes on the Storage Subsystem. Configure these disks for the liveCache application (data, log and instance) and prepare the PowerHA CAA in the OS.
7) Create File system for SAPDB software stack (/sapdb) on both nodes
8) Install SAPDB software stack on both nodes with SDBSETUP (see section 6.2)
9) Execute PowerHA initial cluster setup. – see section 5.4.3.
10) From one node: Setup a lock directory shared between both nodes (see Section 5.4.6)
11) From one node: Create the concurrent log as described in chapter 9.5.1 (PowerHA configuration) and 5.4.4 (disk structure) on both nodes. Ensure ownership of raw devices is set correctly on both nodes (section 5.4.5)
    a. Create data VG's on the primary node via smit vg "Create a Volume Group"
    b. Create data LV's on the primary node via smit lv "Create a Logical Volume"
    c. Create Log VG's via smit cl_vg (CSPOC) -> „Create a Volume Group"
    d. Create Log LV's via smit cl_lv (CSPOC) -> „Add a Logical Volume"
    e. Varyonvg Data VG's in active mode
    f. Varyonvg log VG's in concurrent mode
12) Install the Master instance on the primary node; do not configure it as master. If it is a new installation you can initialize the database as documented in sections 6.3 and 9.2 otherwise leave it as is.
13) Prepare the OS environment
    This section discusses some of the prerequisites that should be reviewed before the implementation is started. A review of these requirements can help simplify and expedite the implementation effort.

5.3.1 Install OS and PowerHA software

OS level: AIX 6.1 TL7 SP03
PowerHA: 7.1.1 SP01 + efix

NOTE: for intermediate term an efix on top of PowerHA 7.1.1 SP01 must be requested through a PMR.

On both nodes install following PowerHA software components:

PowerHA basic software components
PowerHA Smart Assist software component

5.3.2 Set global required OS and TCPIP parameter

These tasks have to be done on both nodes.

5.3.2.1 OS tuning

Increase the number of max processes to at least 2000. This can be done with the following OS command:

chdev -l sys0 -a maxuproc='2000'

5.3.2.2 TCPIP tuning

To avoid batch jobs becoming hung as a result of the failover, the socket handling needs to be adjusted. The time until the connection is closed needs to be reduced on the liveCache LPARs and on all application servers which connect to the liveCache. The following recommendations have to be validated against other prerequisites which may be set by applications not part of this PoC to ensure that there are no unexpected side effects.

It is essential to make the changes to be made permanent to span a reboot.
The following values are set:

no -o tcp_keepintvl=10
no -o tcp_keepidle=300
no -o tcp_keepinit=20

To make them permanent after reboot the -r flag is required.

For example: no -r -o <param>=<val>.

5.3.3 Change root user limits

Same installation and configuration steps will be run as user root. Set the following the soft and hard limits for CPU time, file size, data segment size, RSS size and stack size to unlimited for user root. This can be done by the following command:

```
chuser fsize='-1' data='-1' stack='-1' rss='-1' \
    cpu_hard='-1' fsize_hard='-1' data_hard='-1' stack_hard='-1' rss_hard='-1' \
    root
```

5.3.4 OS groups and users for SAP and SAPDB

Before installing the liveCache software, certain prerequisites for users, groups and database paths have to be met.
The user IDs and group IDs for liveCache must be the same on both servers, as well as the directory layout: dependent and independent paths for liveCache, as defined by SAP, must be identical. This can be achieved by determining the paths in advance and ensuring that the user ID and group ID to be used are available on both systems.

5.3.4.1 liveCache OS Administration User (if used)

The mandatory liveCache user and group is sdb:sdba. This is the user which executes OS level activities for liveCache. Additionally there is usually an administration user such as *<SID>adm*. The administration user is usually defined during the normal SAP installation. This user should also have the same UID/GID on both nodes. It is possible to setup this user later if necessary.

If no <*SID*>*adm* user is installed on the liveCache server, the sdb or other user can be used as the admin user. To do this, the default profiles must be modified. The liveCache environment profile should be copied into */home/*<*user*>.
The *.profile* script will need to be modified to call the liveCache profile.
These profiles can be copied from the application server where the liveCache client is installed under <*SID*>*adm*. They include:

.lcenv.sh
.profile

5.3.4.2 Create required SAP and SAPDB groups

SAP liveCache requires several users and user groups. For HotStandby, these must have the same characteristics on both nodes.

Create OS groups

On both nodes create the following SAP and SAPDB required OS groups by using smit or the command line on both nodes. The group names are "sapsys", "sapinst" and "sdba".

Example:

```
>> mkgroup -'A' id='301' sapsys
>> mkgroup -'A' id='302' sapinst
>> mkgroup -'A' id='303' sdba
```

Create OS users

On both nodes create the SAPDB administration user (sdb) and the SAP administration user (<sid>adm) by using the command line or using smit.

Both user must have the same uid on both nodes. For both users set the soft and hard limits for CPU time, file size, data segment size, RSS size and stack size to unlimited.

Example:

```
>> mkuser id='303' pgrp='sapsys' groups='sapsys,sdba' \
fsize='-1' data='-1' stack='-1' rss='-1' \
cpu_hard='-1' fsize_hard='-1' data_hard='-1' stack_hard='-1' rss_hard='-1' \
hlcadm
>> mkuser id='303' pgrp='sdba' \
fsize='-1' data='-1' stack='-1' rss='-1' \
cpu_hard='-1' fsize_hard='-1' data_hard='-1' stack_hard='-1' rss_hard='-1' \
sdb
```

For both user set the password via passwd.

The SAPDB administration user sdb must be locked. This can do with the following command: "chuser account_locked='true' sdb"

At OS level, the operating system settings for max processes must increase. A rule of thumb for this setting can be >=2000.

liveCache OS User and Software Owner - requirement details
The liveCache user on both systems is sdb with group sdba. This user must be set to use unlimited resources

# su - sdb	
$ ulimit -a	
time(seconds)	unlimited
file(blocks)	unlimited
data(kbytes)	unlimited
stack(kbytes)	unlimited
memory(kbytes)	unlimited
coredump(blocks)	unlimited
nofiles(descriptors)	2000
threads(per process)	unlimited
processes(per user)	unlimited

Group set is: sdba and staff.

The user account should also be locked. If not, liveCache installations often fail as this is checked.

Is this user ACCOUNT LOCKED? true

5.4 Base setup of PowerHA cluster software

PowerHA 7.1.1 requires AIX 6.1 TL7 with service pack 3, or AIX 7.1 TL1 as an OS level prerequisite. An IBM RedBook is available online which goes into detail on the functionality of PowerHA 7.1.1.
See chapter 13.1 for links to related documents and sources of further information.

5.4.1 New requirements for PowerHA 7.1 and above

There are two significant differences in the design of 7.1 and above, when compared to PowerHA 6.1 which should be considered in the planning stage.

1. **Multicast IP**
PowerHA 7.1 and above uses multicasting for heartbeat and cluster communication, and therefore, the switches in the environment should be enabled for multicast traffic. If necessary switch settings may need modification. The test tool "*mping*" provides functionality similar to the point to point IP test tool "*ping*", and can be used to test multicast connections. Use the *mping* tool first at AIX level to make sure that the multicast packets are flowing between the nodes. The tool *mping* requires that you start *mping* on the receive node first to look for a particular multicast address and then send a packet from the other node using *mping* for that particular multicast address. Any multicast communication issues must be resolved before taking up clustering. *mping* is also highlighted in the PowerHA RedBook mentioned above. This implies all networks hosting a service IP alias need to be enabled.

2. **Shared repository disk**
The heartbeat disk, which was supported as an optional communication path in PowerHA 6.1 is no longer necessary or actively supported from 7.1. A shared disk is, however, now mandatory for the purposes of a centralized cluster software repository.
This repository disk stores some of the configuration information centrally as well as providing the disk heart beat functionality. Currently only one disk is supported as a

repository disk and therefore this disk should be highly available. In the proof of concept, this disk is mirrored at hardware level by the SVC over multiple storage servers. For single storage solutions, such as the DS8000, this disk will be protected at storage level by RAID only. This single disk implementation is a temporary restriction of PowerHA which will be addressed in future.

5.4.2 Preparing IP for the cluster

1. **Setup IP addresses and hostnames in /etc/hosts:**

Setting up all cluster used IP addresses and hostnames into /etc/hosts on each cluster node.

2. **Setup cluster nodes IP addresses in /etc/cluster/rhosts**

Inserting the IP addresses of all cluster nodes in /etc/cluster/rhosts. Copy this file to all cluster nodes.

3. **Add service IP alias in /etc/hosts**

Prerequisites for this IP are described in section 5.4.7.

5.4.3 Run cluster initial step by using smit

The basic setup of the PowerHA cluster software depends on 3 main steps. First setup the cluster name and the involved cluster nodes. In the second step define the repository disk and the cluster IP address, which based on a multicast IP address. At the end verify and synchronize the cluster configuration. For details please refer to the PowerHA documentation coming along with your release.

1. Ensure clcomd daemon is running on both nodes
2. smit cm_setup_menu -> Run „Setup a Cluster, Nodes and Networks“

```
                 Setup Cluster, Nodes and Networks (Typical)

                                                        [Entry Fields]
* Cluster Name                                          [HLC_cluster]
  New Nodes (via selected communication paths)          [is04d6]
  Currently Configured Node(s)                          is03d6
```

3. smit cm_setup_menu -> Run „Define Repository Disk and Cluster IP Address“

```
              Define Repository and Cluster IP Address
                                                  [Entry Fields]
* Cluster Name                                    HLC_cluster
* Repository Disk                                 [hdisk2]
  Cluster IP Address                              [228.153.164.129]
```

Nothing should be defined on this disk, no volume group or logical volumes. PowerHA will find present suitable disks for selection and then create its own volume group. This disk is used for internal cluster information sharing and heart beat information. It is reserved for the cluster.

The disk selected now appears to the OS as shown below. It has its own disk type designation and has created a private volume group.

HDISK	PVID	VolumeGrp	Status
hdisk4	00f641cd824eb966	sapdbvg	active
hdisk5	00f641d4f32381da	lclogvg	concurrent
caa_private0	00f641d4f32707c6	caavg_private	active
hdisk1	00f641d409c08c10	lcdatavg01	active

4. smit cm_cluster_nodes_networks -> Run „Verify and Synchronize Cluster Configuration“
5. Start cluster services on both nodes (clstart). This is a prerequisite for the subsequent tasks.

5.4.4 Disks and Volumes

The enablement for HotStandby has specific requirements in regards to the storage layout, the volume groups and logical volumes. In general the distinction is made between a local disk, a shared disk, and a disk pair in a FlashCopy relationship. The Figure 20will give an overview.

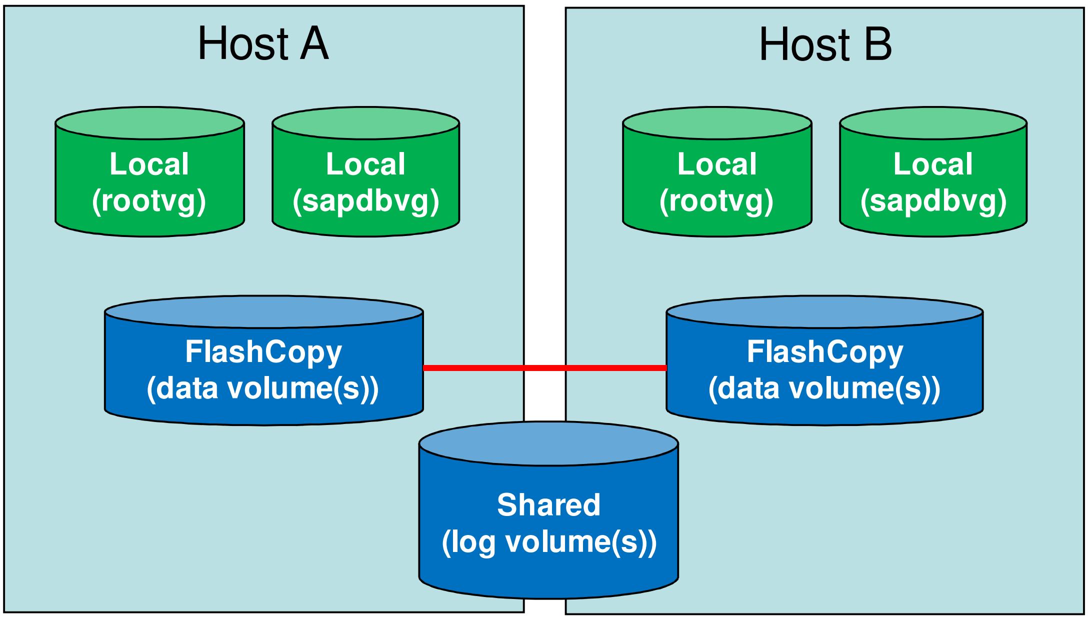


Figure 20: Disk types used in HotStandby setup

The HotStandby will have dedicated raw disks for data and log (blue). The content on these disks is shared across nodes either as part of a FlashCopy pair, or by means of concurrent access. liveCache supports up to 256 logical volumes, each up to (but not exceeding) 128GB in size. HOWEVER – the PowerHA HotStandby solution does not. The limits are documented in section 4.

In addition to these disks there is data which remains unique to the individual node (green). This includes the liveCache executable, the traces, working directories and kernel information belonging to the liveCache instance. This data is normally found in the directory */sapdb*.
In the proof of concept, the directory */sapdb* is resident on each node and is implemented on storage dedicated to this node.

NOTE: The setup and creation of the concurrent volume groups for the log is describe in section 9.5.1

5.4.4.1 Prepare local volumes for the liveCache software on both nodes

Beside the rootvg there are two more candidates for a local file system:

- /sapdb the liveCache executable and logs
- /archive the liveCache log archives

The archive vg can be either shared on base of NFS or local. Local has the advantage for more robustness and better performance

Example for /sapdb:

mkvg –y sapdbvg -V 103 -s128 -B hdisk10
mklv -y'lvsapdb' -t'jfs2' -e'x' -s's' -u4 sapdbvg 470 hdisk10
crfs -v jfs2 -d lvsapdb -m /sapdb -p 'rw' -a agblksize=4096 -a logname=INLINE

Ideally these and the rootvg are mirrored to different storage subsystems by means of LVM mirroring or SVC.

5.4.4.2 Prepare data volumes on the primary node

The FlashCopy is based on physical disk block. Hence, the source and target volumes used for data must be identical in size.

Having the redundant storage layout there is no value in adding the data disks into PowerHA or using enhanced concurrent volume groups in non-concurrent mode.

The volume groups for data are standard volume groups which then can be extended in size and maintained much more flexible. They will be brought online automatically by the restart of the Operating system. Ensure you have planned for the liveCache limit for lv sizes and the Smart Assist limitations (section 4.1).

mkvg -y datavg01 -B -V 102 -s 128 hdisk4 hdisk5 hdisk6 hdisk7 hdisk8

Here an example for a data volume group for a liveCache:

# lsvg datavg01			
VOLUME GROUP:	datavg01	VG IDENTIFIER:	00f641d400004c0000000133848c2b2a
VG STATE:	active	PP SIZE:	32 megabyte(s)
VG PERMISSION:	read/write	TOTAL PPs:	319 (10208 megabytes)
MAX LVs:	256	FREE PPs:	19 (608 megabytes)
LVs:	2	USED PPs:	300 (9600 megabytes)
OPEN LVs:	2	QUORUM:	2 (Enabled)
TOTAL PVs:	1	VG DESCRIPTORS:	2
STALE PVs:	0	STALE PPs:	0
ACTIVE PVs:	1	AUTO ON:	yes
MAX PPs per VG:	32512		
MAX PPs per PV:	1016	MAX PVs:	32
LTG size (Dynamic):	256 kilobyte	AUTO SYNC:	no
HOT SPARE:	no	BB POLICY:	relocatable
PV RESTRICTION:	none	INFINITE RETRY:	no

NOTE: Creating the lvs using smitty the auto select no longer provides the type “raw”. Selecting jfs will still create a raw device in /dev/r<lvname>.

In a second step the logical volumes within the volume group are to be created. Important is that these are to become a raw device. An example is shown below:

mklv –y lcdatalv0101 -t'raw' datavg01 hdisk4

Here an example for a data logical volume for a liveCache:

```
# lslv lcdatavg0101
LOGICAL VOLUME:     lcdatavg0101                           VOLUME GROUP:   datavg01
LV IDENTIFIER:      00f641d400004c0000000133848c2b2a.1     PERMISSION:     read/write
VG STATE:           active/complete                        LV STATE:       opened/syncd
TYPE:               jfs                                    WRITE VERIFY:   off
MAX LPs:            512                                    PP SIZE:        32 megabyte(s)
COPIES:             1                                      SCHED POLICY:   parallel
LPs:                150                                    PPs:            150
STALE PPs:          0                                      BB POLICY:      relocatable
INTER-POLICY:       minimum                                RELOCATABLE:    yes
INTRA-POLICY:       middle                                 UPPER BOUND:    32
MOUNT POINT:        N/A                                    LABEL:          None
MIRROR WRITE CONSISTENCY:        on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
INFINITE RETRY:     no
```

For all raw devices created on the primary node, the permissions should be given liveCache ownership as described later.

NOTE: Only if the volume group layout is modified (new disks or logical volumes are added), a resynchronization needs to be done at volume group level. In this case, the volume group will need to be offline, then exported and reimported on the 2nd node.

5.4.4.3 Prepare data volumes add-on [Manual only]

The PowerHA wizard for liveCache hot standby automates the following task for the data volumes after completing the prerequisites listed previously. If the wizard is not used, these steps will have to be performed manually. The manual steps are documented here.

After attaching the storage volumes to both nodes and configure the volume groups for data on the primary node the wizard initiates the first FlashCopy. The first FlashCopy will duplicate the previously assigned PVIDs – done on the primary node on OS level – and the logical volume layout from the primary to the secondary. The volume group will then be imported on the standby with all the necessary information. Subsequent liveCache activity will not change the layout. To perform these steps manually the following commands have to be used.

- Node1: data vgs created
- Node2: rmdev –dl <hdisk> for all data hdisks.
- Node1: issue a FlashCopy of all data disks to node2 what requires the ssh to the SVC to be set up already (section 7):
  su - sdb
  ssh admin@<svc ip>
  svctask mkfcconsistgrp -name <SID>_01
  svctask mkfcmap -source <volume1_node1> -target <volume1_node2> -consistgrp <SID>_01 -copyrate <choose rate>
  svctask mkfcmap -source <volume2_node1>-target <volume2_node2> -consistgrp <SID>_01 -copyrate <choose rate>
  [… all other data volumes ….]
  svctask startfcconsistgrp -prep <SID>_01

  The default copy rate is to run very slowly in background so as to not bother production I/O rates. In our case however we want a bit faster action and if building this by hand, the copyrate can be set. Recommended rate=100

- Node2: cfmgr
- Node2: importvg -y lcdatavg01 -V <same major as on node1> hdisk<no>
- Node2: Change ownership of raw devices to sdb.sdba
- Node1: Remove task on svc /opt/ibm/ibmsap/connectors/HSS2145/rsRemoveTask.sh -u admin -s <svc ip> <SID>_01

5.4.5 liveCache ownership of raw devices

The log and data volumes of a HotStandby liveCache are accessed as raw devices. The liveCache user (sdb) must have the authority to access the devices at raw level.

Set permissions for liveCache raw devices to user sdb and group sdba. This remains the same regardless of the liveCache SID and the administration user <SID>adm.
In this example we have two logical volumes, one for the log and one for the data. They are named lcloglv and lcdatalv respectively.

chown sdb.sdba /dev/rlcdatalv
chown sdb.sdba /dev/rlcloglv

Verify the status

*ls -la /dev/rlc**

```
crw-rw----  1 sdb   sdba   100,   1 Nov  4 12:04   /dev/rlcdatalv
crw-rw----  1 sdb   sdba    37,   1 Nov  2 19:28   /dev/rlcloglv
crw-rw----  1 root  sdba    36,   1 Nov  1 17:01   /dev/rlcsapdblv
```

The wizard will automate the ownership for data logical volumes on the standby node. The ownership for data logical volumes on the primary and for all log logical volumes has to be done manually. For a manual approach the permissions needs to be handled on both nodes.

NOTE: in case a manual import of the data volumes is performed at any time, these values reset to root, and therefore, this must be repeated after the import.

5.4.6 Setup of cluster state lock directory

This directory needs to be dedicated to this particular liveCache instance. The following users need to have read and write access:

- root
- sdb

It has to be:

- Attached to both nodes
- Highly Available

The mount point of this directory is one of the configuration parameters of the automation.

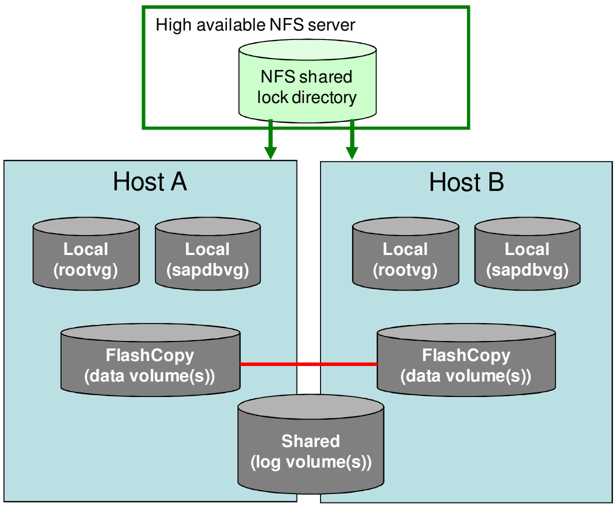
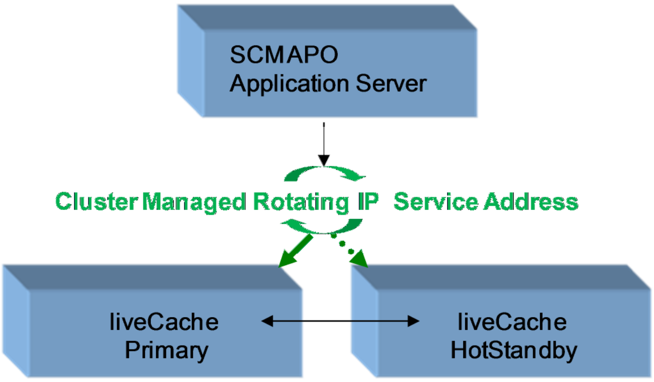


Figure 21: NFS filesystem for shared state information

Attention: The HotStandby cluster design depends on state information in the shared files system and sets an HA shared filesystem as prerequisite. Loss of the NFS service can impact the overall solution reliability.

5.4.7 Official hostname – IP alias considerations

The official host name configured in liveCache is the IP address used to access the liveCache from the APO application. It is implemented as a virtual rotating address.



The cluster services ensure that the current active liveCache is the owner of this address. In words of PowerHA it is the service IP moving along with the master resource group.
From PowerHA 7.1 onwards the networking interface hosting the service IP alias has to be enabled for multicast.

Figure 22: Reference overview HotStandby rotating service IP alias

NOTE: The APO accepts only letters and numbers for the official hostname; other characters will cause problems in connection later on. An example format for the name could be *lc<SID>ip* or in the case of the SID being HLC: *lchlcip*.

6 Installing liveCache

The installation is split up into 2 + 2 phases:

1. The installation of the general executable (both nodes).	6.2 or 6.2.2
2. The creation of the liveCache instance (node1).	6.3.1 or 0
3. The creation of the master instance (on node1) [Manual]	9.3
4. The activation of the standby instance (from node1). [Manual]	0

Prior to this step a storage location must be prepared for the liveCache software and instance administration as described in 5.4.4 .

NOTE: When creating a liveCache instance, you will be asked to select a liveCache user which is the SQL user that APO will use. The SQL user is the owner of the database scheme. The default proposed in the tools is SAP<SID> but any name can be selected. As this only has relevance in the APO integration and is only used by APO to access the liveCache data via SQL, a common user name is recommended, for example LCUSER. This has the benefit that data backed up from the production system can be restored in the test liveCache without having to change the SQL user, reassign the schema, or modify the APO integration. The SQL user, as owner of the database scheme, is transported with the backup/restore. Having a common user simplifies life.

The installation and instance creation step can be done using one of two approaches: using SDBINST and MaxDB DBMGUI, or by using SDBSETUP and then SAPinst. In general SDBINST is used when updating an existing liveCache and SDBSETUP in the case of a new installation. Both tools however can be used for installing the software.
At the end all that matters is the consistency of the installation. Special care should be taken to ensure that the independent and dependent paths of the liveCache are set identically on both nodes.

NOTE: The PowerHA 7.1.1 SP1 Wizard may have issues when trying to install on a cluster where there is more than one version of liveCache. Therefore if the liveCache has been installed and then upgraded, it may be necessary to perform the manual implementation path.

The following table summarizes the installation methods:

	SDBINST + DBMGUI	SDBSETUP + SAPinst
Preferred scenario	Update	New installation
Installation of the software (both nodes)	Installing SAP liveCache Executable using SDBINST option (6.2.1)	Installing liveCache software using SDBSETUP (6.2.2)
Installation of the database instance (primary node only)	Installing the liveCache database instance (6.3.1)	Using the installation CD to install a liveCache instance (0)

Figure 23: Installation methods

6.1 LiveCache dependent and independent program paths

LiveCache can support multiple liveCache instances at different installation levels on a single server. To do this, there is a concept of dependent and independent paths for executable and related configuration data. This section shows how to verify the different paths that liveCache uses and the target paths for the current installation.

6.1.1 Locating the independent path

The independent path is the directory structure for components that are common to all local implementations. The command line interface, *dbmcli*, the *x_server* and other such general programs are found under this directory path.

The independent path can be found in the configuration file: */usr/spool/sql/ini/SAP_DBTech.ini.* The following example shows an independent program path of the */sapdb/db/programs.*

```
[Globals]
IndepData=/sapdb/sdb/data
IndepPrograms=/sapdb/db/programs

[Runtime]
/sapdb/db/programs/runtime/7300=7.3.0.0,
/sapdb/db/programs/runtime/7301=7.3.1.0,
/sapdb/db/programs/runtime/7403=7.4.3.0,
```

6.1.2 Locating the dependent path

The dependent path contains components that are particular to a given installation.
It can be found by command line: *<independent path>/bin/dbmcli db_enum*

Output example shows that the dependent path */sapdb/HL2/db*

```
is04d1:sdb> dbmcli db_enum
OK
HL2    /sapdb/HL2/db          7.7.07.39    fast    offline
HL2    /sapdb/HL2/db          7.7.07.39    quick   offline
HL2    /sapdb/HL2/db          7.7.07.39    slow    offline
HL2    /sapdb/HL2/db          7.7.07.39    test    offline
```

Alternatively, there is an installation file in */sapdb* which has related information.

```
# cd /sapdb
```

 # find . | grep Installations.ini

This file shows the dependent path for HL2 is */sapdb/HL2/db*

[Installations]
/sapdb/HL2/db=7.7.7.39

[Params-/sapdb/HL2/db]
Type=0x0

6.2 Installing SAP liveCache Executable

This section covers the two methods of installing SAP liveCache executable. In one case the installation tool is used, SDBSETUP – this would be the likely scenario for a new SCM installation. In this case the full install DVDs are available.

In the alternative method, SDBINST is used, which would be a likely scenario when an existing SCM system is being moved to HotStandby. This method is based on a download of the liveCache install image from the SAP marketplace.

NOTE: see http://www.sapdb.org/htmhelp/ef/00299bf1fb11d4aa9a006094b92fad/frameset.htm for more information on the necessary user types for implementing liveCache. There are several users required: one is the DBM manager which will be used by APO for stopping, starting, monitoring and reinitializing the liveCache. This user is typically CONTROL. This is also used by the cluster to manage the HotStandby.
The second user is the DB user which will be used by APO to access the liveCache business data. This SQL user is the owner of the database scheme. This is typically sap<sid>. However it is recommended to use a common name such as LCUSER for backup and maintenance convenience.

The OS level group and owner of the database software is always sdb:sdba.

6.2.1 SDBINST option

In the following example, liveCache was installed from an image downloaded from the SAP marketplace using command line installation SDBINST. The installation images in the example below are running from a shared file system.

Using SAP SDBINST to install the liveCache software, select the option for “Server + Client”.
In this example we have selected the dependent data path in accordance to the SID selected for the liveCache instance being implemented: */sapdb/HLC*. The result is that all the dependent data generated for the implementation will be placed under the relevant SID in the */sapdb* file-system.

```
#./SDBINST
        Installation of SAP liveCache Software
        ****************************************

starting installation Fr, Jan 13, 2012 at 13:47:33
operating system: AIX PowerPC 6.1.0.0
callers working directory: /sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64
installer directory: /sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64
archive directory: /sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64

existing component groups:
0:  Server + Client
1:  Client
2:  Custom
3:  none
please enter component group id: 0

starting preparing phase of package Base 7.7.07.39 64 bit
--------------------------------------------------------

no updatable installation of package "Base" found
please enter group name for database programs [sdba]:
please enter owner name for database programs [sdb]:
please enter independent data path [/var/opt/sdb/data]: /sapdb/data
directory "/sapdb/data" does not exist, create? (y/n) y
please enter independent program path [/opt/sdb/programs]: /sapdb/programs
directory "/sapdb/programs" does not exist, create? (y/n) y
checking interferences to other packages... ok

collecting data finished:

independent data path: /sapdb/data
independent program path: /sapdb/programs
owner: sdb
group: sdba
start extraction test run of "/sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64/SDBBAS.TGZ"
checking mmap ...
mmap check OK

package Base successfully checked

starting preparing phase of package SAP Utilities 7.7.07.39 64 bit
-----------------------------------------------------------------
checking interferences to other packages... ok
collecting data finished:
: /sapdb/data
independent program path: /sapdb/programs
owner: sdb
group: sdba
start extraction test run of "/sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64/SAPUTL.TGZ"

package SAP Utilities successfully checked

starting preparing phase of package Server Utilities 7.7.07.39 64 bit
--------------------------------------------------------------------
checking interferences to other packages... ok
collecting data finished:
independent program path: /sapdb/programs
owner: sdb
group: sdba
```

```
start extraction test run of "/sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64/SDBUTL.TGZ"

package Server Utilities successfully checked

starting preparing phase of package Database Kernel 7.7.07.39 64 bit
------------------------------------------------------------------

no updatable installation of package "Database Kernel" found
please enter dependent path [/opt/sdb/7707]: /sapdb/HLC/db
directory "/sapdb/HLC/db" does not exist, create? (y/n) y
checking interferences to other packages... ok

collecting data finished:

dependent path: /sapdb/HLC/db
owner: sdb
group: sdba
start extraction test run of "/sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64/SDBKRN.TGZ"

package Database Kernel successfully checked

starting preparing phase of package APO LC APPS 7.00.012 64 bit
-------------------------------------------------------------
checking interferences to other packages... ok
collecting data finished:
apo com path: /sapdb/HLC/db/sap
owner: sdb
group: sdba
start extraction test run of "/sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64/APOLCA.TGZ"
package APO LC APPS successfully checked

[…]

starting preparing phase of package SQLDBC 76 7.6.06.07 64 bit
------------------------------------------------------------
checking interferences to other packages... ok
collecting data finished:

dbc path: /sapdb/programs
owner: sdb
group: sdba
start extraction test run of "/sysmgmt/LC/DATA_UNITS/LC_AIX_PPC64/SQLDBC76.TGZ"
package SQLDBC 76 successfully checked

checking filesystem "/sapdb"... free disk space ok

starting installation phase of package Base 7.7.07.39 64 bit
----------------------------------------------------------

[…]

checking unpacked archive… ok
installation of SAP liveCache Software finished successfully Fr, Jan 13, 2012 at 13:50:21
```

Figure 24: SDBINST - Screen shot of command line installation

6.2.2 SDBSETUP option

SDBSETUP provides an alternative graphical installation interface to SDBINST when installing from a software CD image. The difference in the two installations is that using this

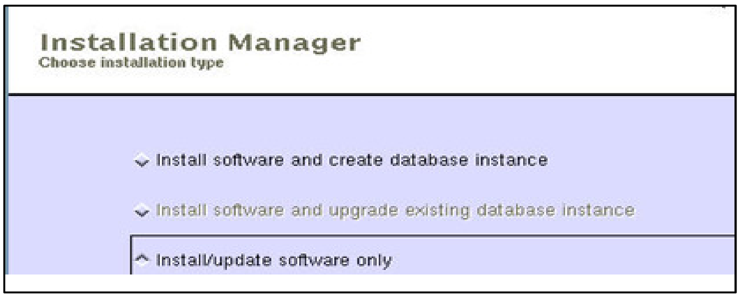
tool, the dependent path is first defined when the liveCache instance is created. Not a major difference. In both cases the liveCache instance is created in a further step.
The actual installation of the liveCache database instance must wait until the storage layout is complete in either case.

The software has to be installed on both nodes whereas the database instance itself is done differently in a later step.
This tool is found on the installation CD under: *DATA_UNITS/LC_AIX_PPC64/SDBSETUP*

NOTE: Before starting to create the instance the log volume group must be brought online as described in 9.5.1 and if a manual installation is required also the required configuration of this resource is required as described in 0.

1. Starting SDBSETUP select "Server+Client" Component as shown in Figure 25.

Figure 25: SDBSETUP - Installation Manager



2. Then select "Install/update software only" to install only the software as shown in

Figure 26.

Figure 26: SDBSETUP - Install software only

3. In the next step define the independent program path and the database software owner.

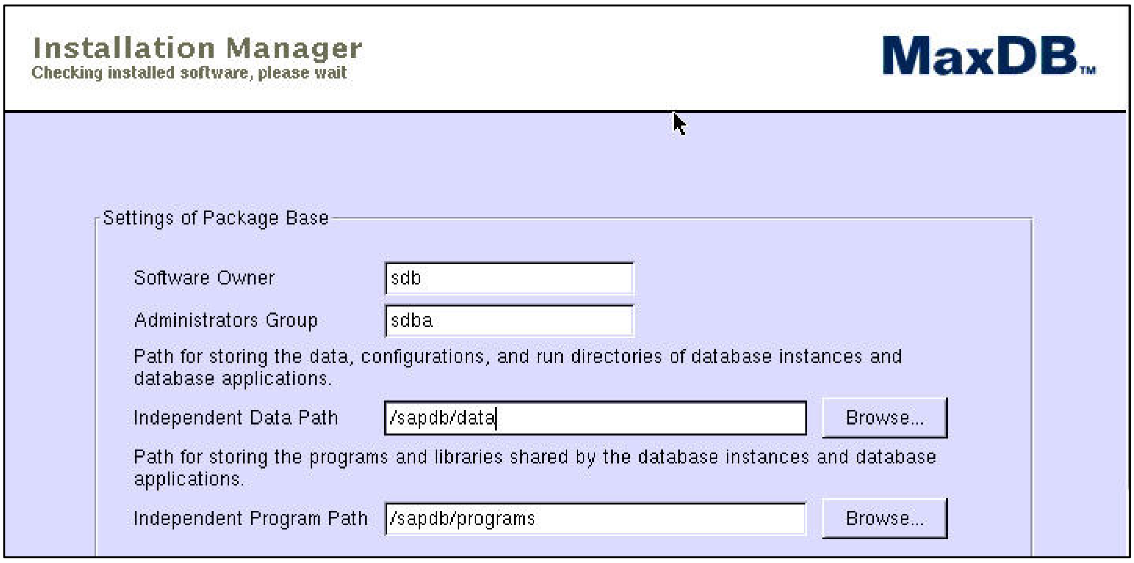


Figure 27: SDBSETUP - define independent paths

4. Repeat step 1-3 on the second node with same user and paths

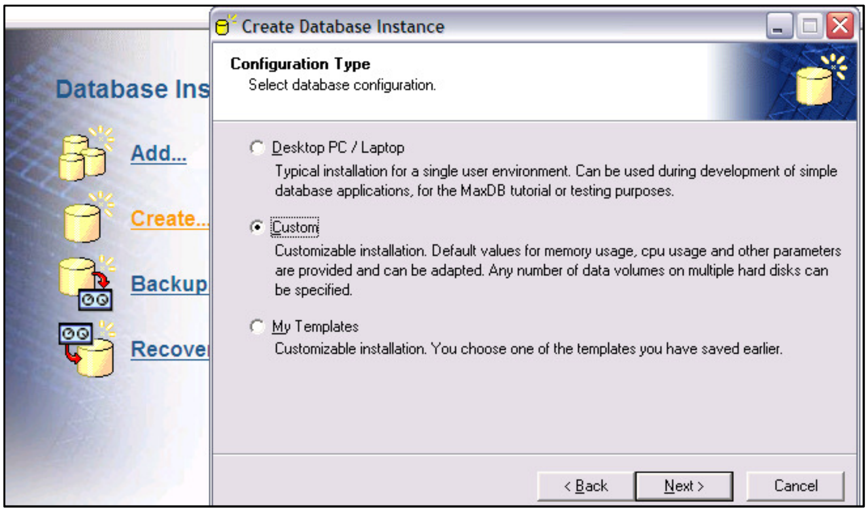
6.3 Installing the SAP liveCache database instance

This step can only be started once the storage layout is completed and the raw disk volumes are available for liveCache. In this step the database instance is configured to use the log and data devices.

The following two approaches are described:

- DBMGUI in section 6.3.1
- SAPinst in 0

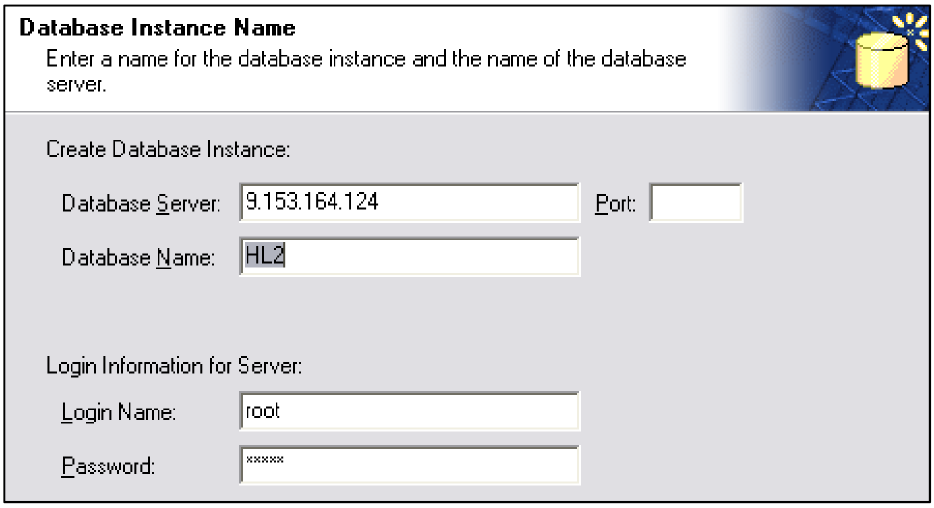
6.3.1 Creating liveCache instance using the DBMGUI



The example of DBMGUI installation follows:
In DBMGUI select *instance* and then select *create and custom.*

Figure 28: DBMGUI – installing the primary instance

Ignore the first error which is looking for a liveCache instance on your local machine and continue to the next option where you enter the target host and the SID for the liveCache "to be". In this example it is HL2. The database name is the SID you have selected for the new instance, and the login name is a user and password which will have the authority to create a liveCache DB image on the target machine.



The user is a user with authorization to install the instance on the target server. We are doing this via root. The selected user should be included in group sdba. This user is only needed for the installation step.

Figure 29: DBMGUI – creation of the liveCache instance using OS level user

Select the installation path used for the dependent path earlier.

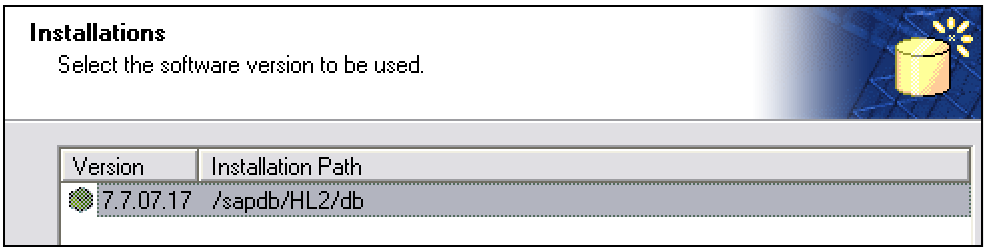
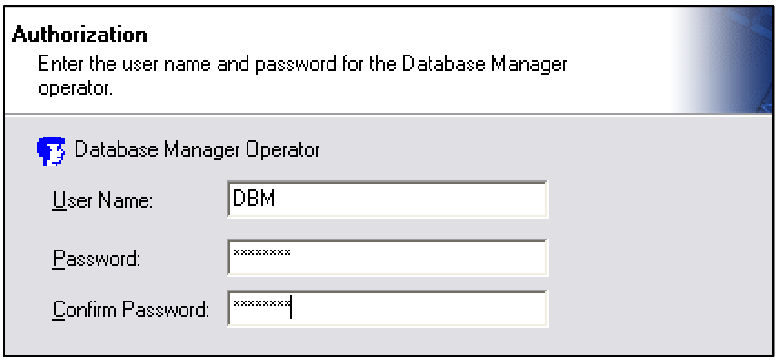


Figure 30: DBMGUI – dependent installation path



Enter the DBM operator name and password. This user will be used as well in the liveCache to APO integration. The DBM operator will be added into the liveCache as a DBA. This user for liveCache implementations is typically either “CONTROL” or as in the default “DBM”.

Figure 31: DBMGUI – set DB Manager name and password

NOTE: The user and password are needed later for integration of liveCache with APO. This user will need to be the database operator defined in the XUSER information for the cluster and used by the APO integration (section 6.3.3).

Select liveCache and initialize with default parameters as shown in Figure 32.

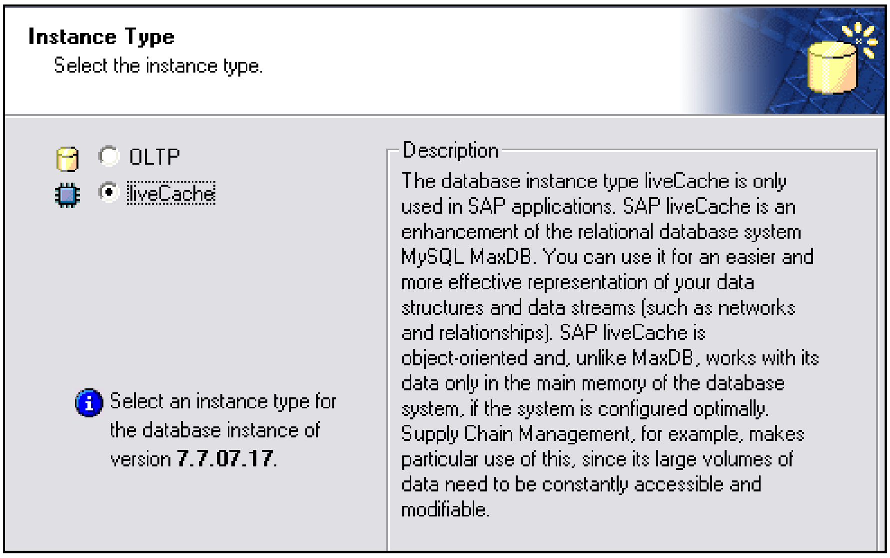


Figure 32: DBMGUI – select liveCache as installation type

Enter the data paths as you have defined them when asked for storage. For a liveCache HotStandby, these are the raw device names such as /dev/rlcdatalv1.

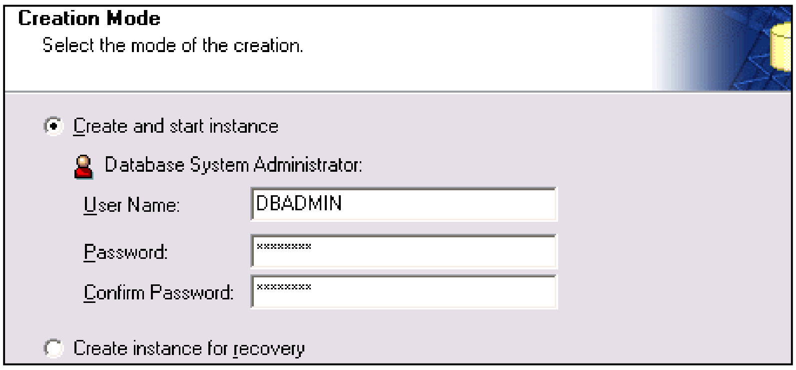
```
# lslv lcdatalv1
LOGICAL VOLUME:     lcdatalv1               VOLUME GROUP:     lcdatavg
LV IDENTIFIER:      00f641d400004c000000012ca1b1530f.2     PERMISSION:     read/write
VG STATE:           active/complete         LV STATE:         opened/syncd
TYPE:               raw                     WRITE VERIFY:     off
MAX LPs:            512                     PP SIZE:          512 megabyte(s)
COPIES:             1                       SCHED POLICY:     parallel
LPs:                133                     PPs:              133
STALE PPs:          0                       BB POLICY:        relocatable
INTER-POLICY:       minimum                 RELOCATABLE:      yes
INTRA-POLICY:       middle                  UPPER BOUND:      128
MOUNT POINT:        N/A                     LABEL:            None
MIRROR WRITE CONSISTENCY: on/ACTIVE
EACH LP COPY ON A SEPARATE PV ?: yes
Serialize IO ?:     NO
```

Figure 33: OS overview of a raw data device used for the datavg

Define both the log and the data volumes as raw devices using the /dev/r<lvname> path. Calculate the number of "*used partitions*" * "*partition size*" to determine the raw data area. Accordingly to the lslv output in Figure 33, the data volume has 133 partitions, each 512 MBs in size gives a volume size of 68 GB. The log volume in the reference installation has 59 partitions of 512 MBs giving a volume size of approximately 28 GB.

Select properties to update the default setting for the data and log volumes. Select the path to the raw device */dev/r<lvname>* and select type "raw". Figure 34 shows an example for the data volume. Same has to be done for the log volume.

Figure 34: DBMGUI – data volume configuration in liveCache



Set the name and password for the DB administrator. This is typically DBADMIN or SUPERDBA as one of the default user names.

Figure 35: DBMGUI – Set password for DB administrator

Once all of the parameters have been entered, the tool will create and initialize the instance, and format the storage volumes. The liveCache database instance will then be activated. The new liveCache will then need to be initialized from APO.

6.3.2 Using SAPinst to create a liveCache database instance

SAPinst on the installation CD provides additional options when creating a liveCache instance. Important entries will be highlighted along with the screenshots.
Start the liveCache installation as shown in Figure 36. Select "*liveCache Server installation*"

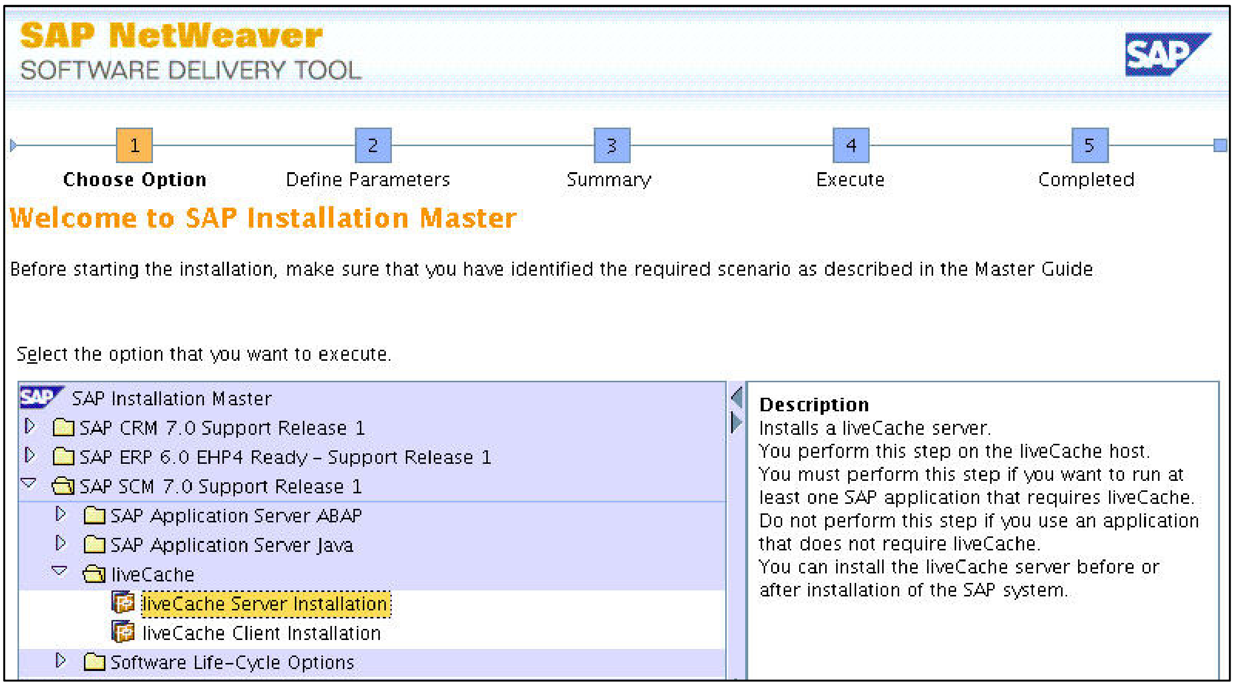


Figure 36: SAPinst – Start the database installation

Ensure in the next screen that liveCache should be aware if the APO system is Unicode. If this is left to default, it must be manually corrected later. The liveCache ID is the SID you have selected

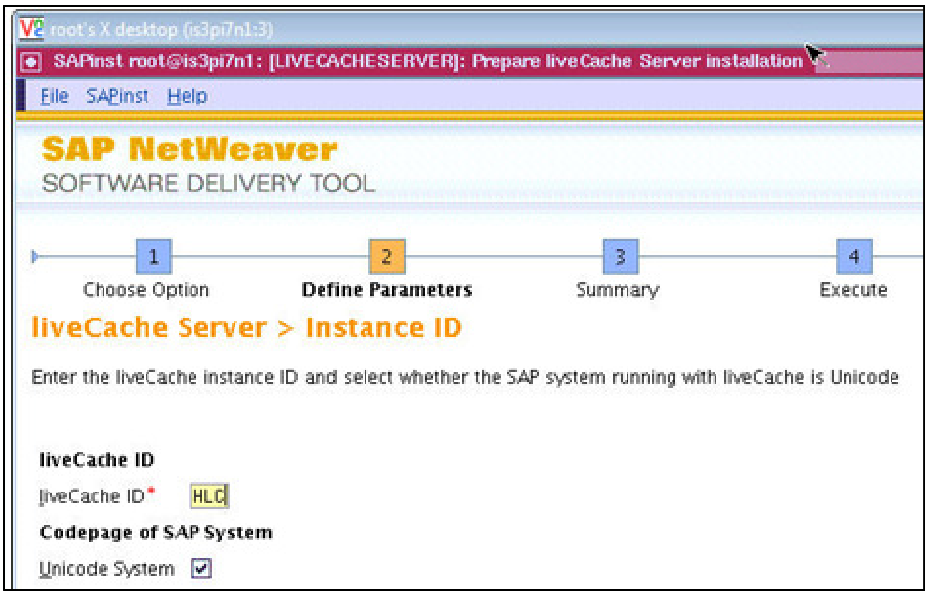


Figure 37: SAPinst – define SID and Unicode settings

Select "*custom*" installation method in the following screen and click next to define the database users. Ensure to choose a user and group ID which is also available on the second node. The user being defined here is the liveCache OS administration user. The tool will prepare the environment of this user with the necessary environment profiles and paths. See section 5.3.4.1.

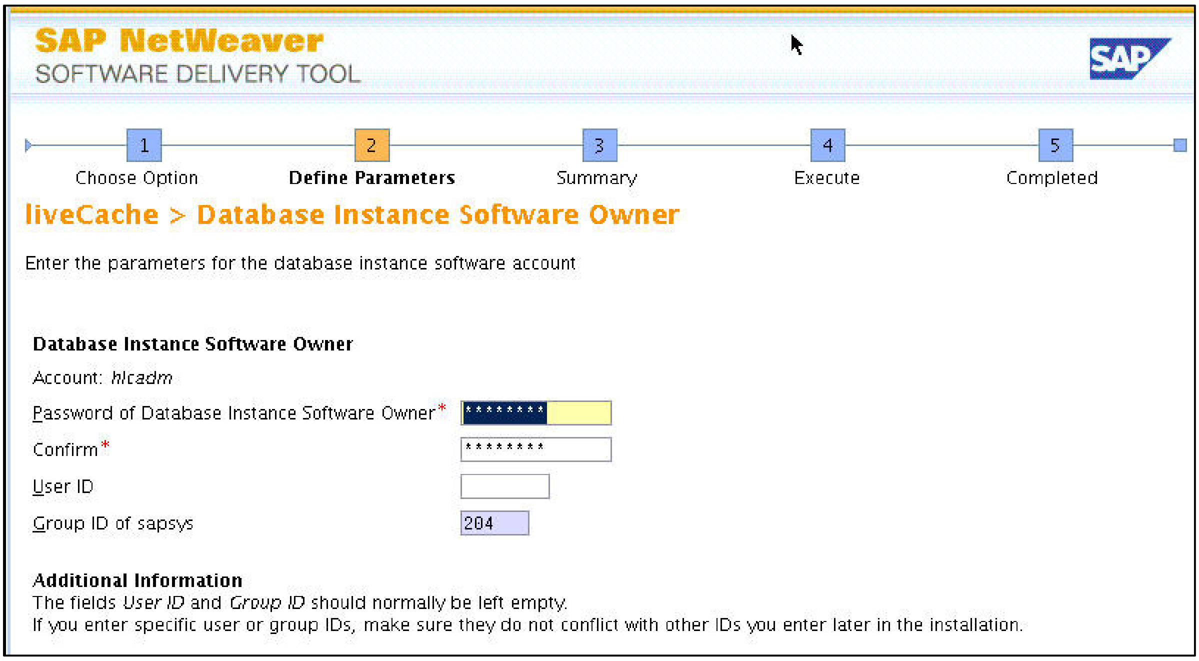


Figure 38: SAPinst – Define user and group ID which is available on both nodes.

NOTE: user and group IDs must be same on both nodes. If the <sid>adm is not created prior to run sapinst this panel is not shown.

This creates a liveCache administration user at OS level if it does not already exist. The profiles of this user are then extended to include the independent program path such that it can access such fundamental commands as dbmcli and x_server. If this user is not created, it can be manually created later. See the section on the liveCache administration section on the liveCache administration user in section 5.3.4.1.

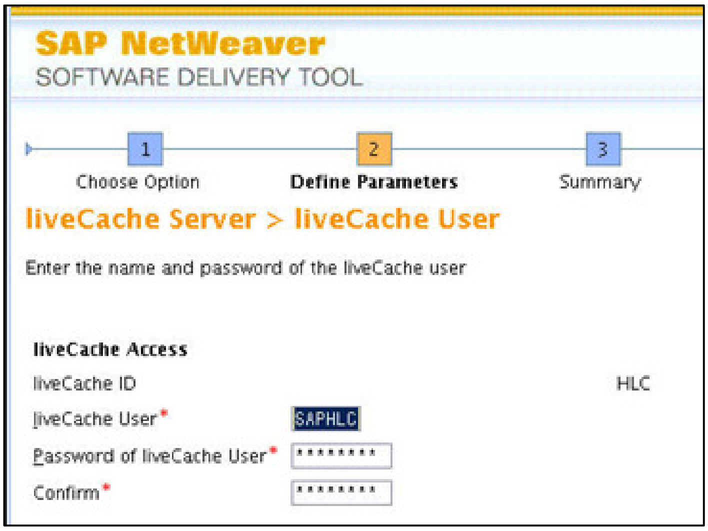


Figure 39: SAPinst – create liveCache user

The liveCache user defined here is the owner of the database schema. This is the DB user referenced earlier and is the SQL user for APO to liveCache. This user and password must be supplied to the APO administrator for integration with APO. APO connects to liveCache via two users, the DBM operator user for liveCache administration such as starting, stopping, reinitializing and tracing, and the sql user (this liveCache user) for access to the actual application data.

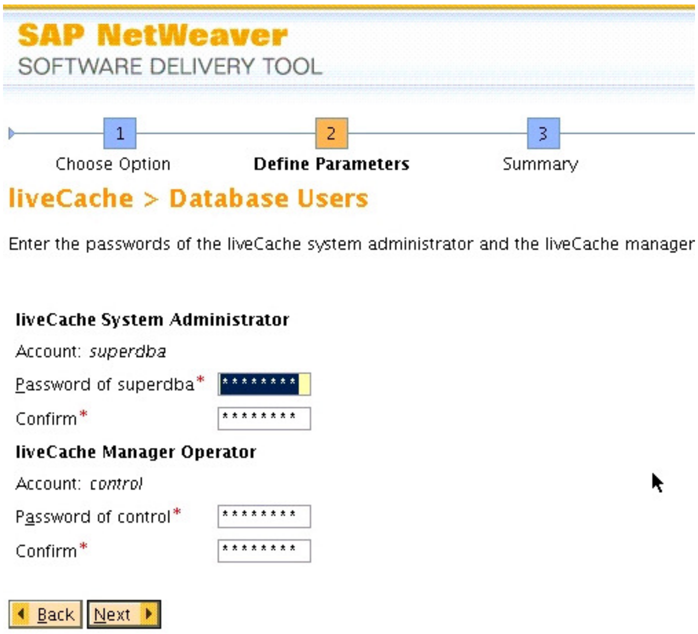


Figure 40: Defining the DB Operator and the Administrator

Two further users are now defined which have relevance only in liveCache. The superdba is the major administrator user. This use is used to create other operators and has full authorization. The second user is of particular importance to the cluster and APO as this is the operator user that APO can use to control the liveCache, and this user is also necessary for the cluster control of the HotStandby. See section 6.3.3. Once this user is defined for the cluster setup, it should not be modified.

SAPinst then continues with the liveCache configuration, similar to the DBMGUI installation above. Now as it comes to the definition of the data and log volumes ensure to choose the raw devices. The screen shot in Figure 41 shows an example for defining a log volume.

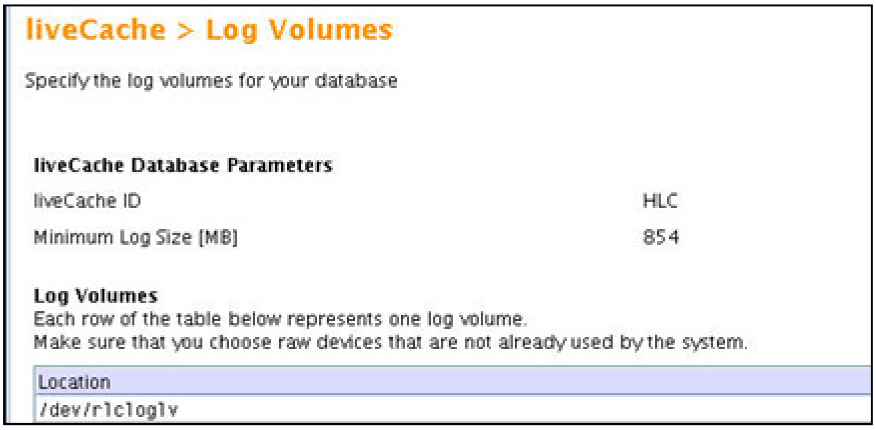


Figure 41: SAPinst – Defining a raw logical volume as log volume

Apply the same for the data volumes and run the installation. After this step node1 is successfully installed. The setup of the standby is discussed later.

6.3.3 Setting up the Required XUSERs for the Cluster

The LiveCache HotStandby solution needs access to the SAPDB database on database level even in the installation, configuration path and at runtime. The database user is independent of the OS user and requires liveCache login information.
To make access to the database easier and more secure, the cluster uses sapdb tools and relies on XUSER encryption of the passwords. For this purpose it's necessary to create an XUSER entry for the database manager user that the cluster will rely on – this is typically either DBM or CONTROL but must have DBM operator authorization. The XUSER entry must be created for **both OS users root and sdb**. The XUSER information is stored in the file .XUSER.62. This file placed in the home directory of the OS user. The entry is created by using the following command from each of the OS users with exactly the same content:

- XUSER name is usually <SID_XUSER>
- Operator is usually CONTROL or DBM. This is the operator specified when installing the liveCache instance see Figure 31 for example.

/sapdb/programs/bin/xuser –U <XUSER> –d <SID> –u <OPERATOR>,<Password>

List the contents of the XUSER file can be do with the following command:

/sapdb/programs/bin/xuser list

Create the environment variable LC_XUSER:

Set the "LC_XUSER" environment variable in the root shell on both cluster nodes to ensure that the automation finds the XUSER information.

export LC_XUSER=<selected XUSER>

If you later switch the shell before calling the Smart assist perform this task again to have it set for the automation.

7 SSH authorization from HotStandby to SAN Volume Controller

The series of steps described here are not extremely intuitive so they must be followed as documented. The end result is that the AIX user, sdb will be able to access the SAN Volume Controller server with the user admin and carry out the necessary commands to drive the liveCache to storage integration.

NOTE: If the installation environment does not allow the use of the SVC "admin" user for such purposes, use the manual instructions for changing the user after creating the ssh keys. If due to security policies, the same ssh key cannot be used from both nodes, a manual workaround is also provided in 0 which should be implemented after the installation is complete.

Setting up the keys on AIX

The user on AIX HotStandby side is sdb. The keys are generated from this user. From root on the primary node, change to the sdb user using the following command.

```
su - sdb
```

Now generate the SSH keys using the following command.

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sdb/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/sdb/.ssh/id_rsa.
Your public key has been saved in /home/sdb/.ssh/id_rsa.pub.
The key fingerprint is:
[…]
```

There should now be a directory under */home/sdb* names .ssh

```
# ls /home/sdb
.profile      .sh_history     key.bk       smit.script
.sdb          .ssh
```

Ensure that the owner is sdb.sdba for both this directory and all files in it

```
cd .ssh
# ls -l
total 24
-rw-------    1 sdb     sdba        1675 Nov  3 19:02 id_rsa
-rw-------    1 sdb     sdba         392 Nov  3 19:02 id_rsa.pub
-rw-r--r--    1 sdb     sdba         442 Nov 15 19:53 known_hosts
```

This directory will need to be copied to all the liveCache servers participating in SAN Volume Controller communication. After copying, ensure that the authorizations are granted for sdb.sdba on all HotStandby servers.

From the laptop, get the public key in binary format through FTP or by other possible means. The key will have the following name:

```
id_rsa.pub
```

This key will now need to be loaded in the SAN Volume Controller.

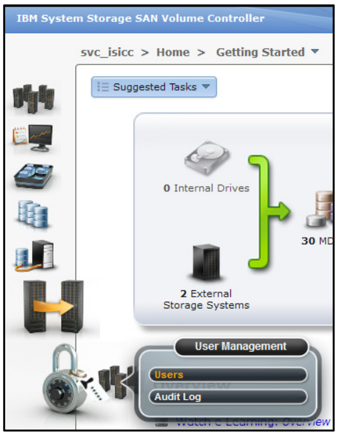


Figure 42: mange users on SVC

Setting up the SAN Volume Controller with the public key access

Login to the SAN Volume Controller through the browser and create a new user.

Use the following characteristics: No password, local authentication, and SSH.

NOTE: the automation will only work for the user named “admin”. As shown also other users can be used along with a manual implementation.

Attributes	Values
ID	10
Name	ha_scm
User Group ID	1
User Group Name	Administrator
Password	No
Authentication	Local
SSH Key	Yes

Figure 43: SVC SSH key

Figure 44: SVC user administration

Browse for the SSH public key file and add the public key loaded up from the AIX server.

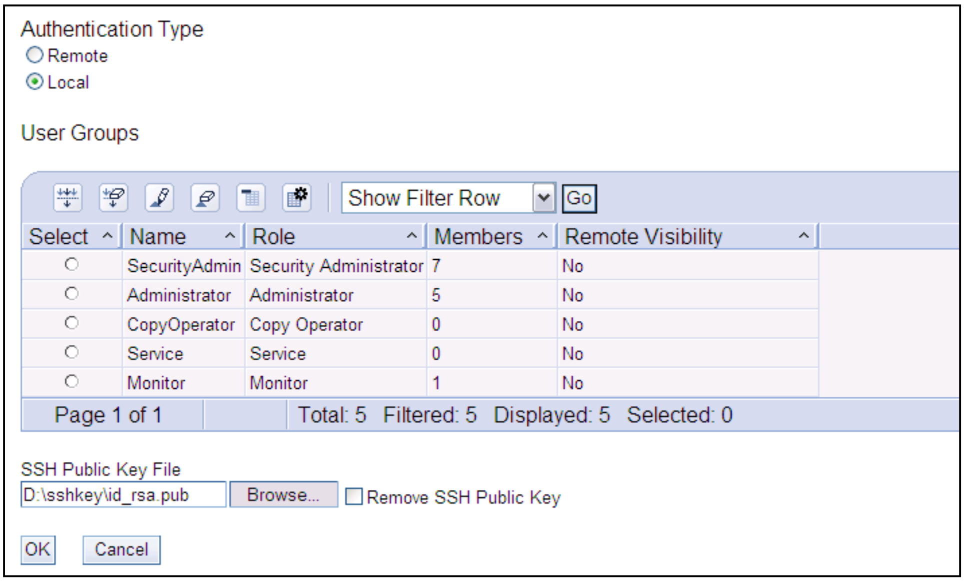


Figure 45: SVC user group administration

NOTE: if you have multiple LC clusters the key can be shared by all if desired.

Testing the access from AIX to SAN Volume Controller

Regardless of the user you specified, you must login using the ID as admin from the liveCache server.

From user sdb, enter ssh admin@<IP address of SAN Volume Controller server>

$ ssh admin@svc_isicc
IBM_2145:svc_isicc:admin>

8 Installation path using Smart Assist and Wizard

The high level flow when exploiting PowerHA 7.1.1 SP1 is described in the following sections. PowerHA provides two parts with the solution. The first part is the wizard helping with the setup of the liveCache environment itself. The second is the Smart Assist agent configuring the cluster. Both can be used independently from each other.

8.1 Run the Wizard

Limitations of the automation given by the wizard:

- For SVC admin user only. If a different user on SVC is used for the ssh connection the manual path has to be taken.
- For SVC only on key for both nodes can be used with the automation. Workaround is provided later.
- Supports only one SVC HMC → See section 9.1.4 how to manually add a second HMC
- The wizard will not work if the standby node has the datavg activated. Ensure that the data volume group is offline and exported from the standby node. The initial flash copy will change PVIDs and structural data which will be reimported.
- The wizard is available when cluster services are running and working correctly on all nodes in the cluster.

NOTE: Ensure that the XUSERs are setup before starting the wizard. See section 6.3.3 for information on the setup. Naming conventions given by the framework must not be changed at all.

Call the wizard to configure the standby. Select “SAP liveCache Hot Standby Configuration Wizard”. To run the wizard use the following smit fastpath: smit clsa.
On this screen select the menu point "SAP liveCache Hot Standby Configuration Wizard"

- ⇒ smitty hacmp
- ⇒ Applications and Resources
- ⇒ Make Applications Highly Available (Use Smart Assists)
- ⇒ SAP liveCache Hot Standby Configuration Wizard

8.1.1 Use the wizard to setup the standby instance and the storage library

Start the wizard by selecting the involved cluster nodes. After this action select the storage system. At this time only SVC is available for this wizard.

Figure 46: Wizard node select panel – choose both

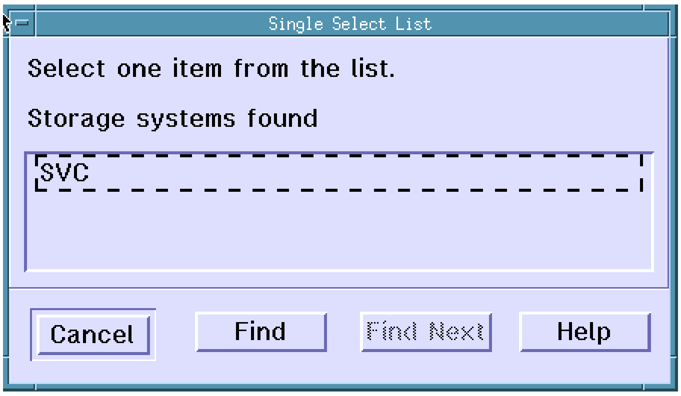


Figure 47: Wizard storage server selection

liveCache HotStandby Parameters for Wizard

In the panel of the wizard depicted in Figure 48 the details of the liveCache implementation are inserted. The relevance of these parameters is described here.

MaxDB DBM User – DBM Operator (Database Manager Operator) DBM operators manage database instances with the database tool Database Manager. This User has created as XUSER with the xuser sapdb utility. Insert only this XUSER-Name. The password for this user is stored in the xuser file and is not visible anymore. Password changes must be done only with sapdb utilities.

The DBM user has relevance to liveCache and to APO only; The DBM user is not an OS level user but a liveCache control user which is referenced via the XUSER information setup in section 6.3.3 and references in Figure 31and Figure 39.

SAP liveCache Hot Standby Configuration

* liveCache Instance Name (SID)	[HLC]
* MaxDB DBM User	[HLC_XUSER]
* Storage server(HMC)	[9.153.165.120]
* liveCache Global Filesystem Mount point	[/archive/sapdb/HLC_LOCK]
* Primary Node	is03d6
* Service IP Label	lchlcip
* Node Names	is03d6,is04d6
* Log Volume Group(s)	lclogvg02 lclogvg01
* Data Volume Group(s)	lcdatavg02 lcdatavg01
* HDISK(s) pairs of Data Volume Group(s)	[hdisk8-->hdisk8,hdisk5-->hdisk5]

Figure 48: Wizard – specify liveCache

NOTE: The syntax for data volumes is important. Please verify the final version on base of the PowerHA Smart Assist manual in case this has changed since SP1.

The entries in Figure 48 will create the following:

- APO DBM user for liveCache will be CONTROL in this case. The xuser entry is HLC_XUSER (it can be any XUSER name created accordingly).
- The liveCache HotStandby cluster will be have the instance name “HLC” on both nodes. This is determined by the SID parameter.
- The storage server is SAN Volume Controller or SVC. Storage server HMC is the address which will be used for the SSH connection to the SVC.
- The “Global Filesystem Mount Point” actually refers to a NFS mounted shared location where cluster state information is maintained. The sdb user must have RW

access to this directory from both nodes. The “Service IP Label” is the IP address used by APO to find the active liveCache primary node. This is a rotating resource which is maintained by the cluster services.

- The HDISK(s) pairs of Data Volume Group(s) link the hdisk of the primary Node to the corresponding hdisk on the secondary Node.

Depending on the release the wizard might not work through smitty. The command line invocation is as follows (using the parameters from the example above in Figure 48.

/usr/es/sbin/cluster/sa/hswizard/sbin/cl_hotstandby_createInstance -I'HLC' -D'HLC_XUSER' -t'SVC' -H'9.153.165.120' -F'/archive/sapdb/HLC_LOCK' -p'is03d6' -s'lchlcip' -n'is03d6,is04d6' -l'lclogvg02 lclogvg01' -d'lcdatavg02 lcdatavg01' -o'hdisk8-->hdisk8,hdisk5-->hdisk5'

NOTE: To specify a second HMC for SVC open the libHSS configuration file, add second HMC by specify CSbIP parameter later– see section 0

8.1.2 Post processing for manually workaround the wizard

If the Wizard could not be used perform the following manual steps regardless whether a manual or automated creation of the Resources is performed

Create in the following files owned by sdb.sdba on both nodes:

vi /usr/es/sbin/cluster/sa/hswizard/etc/LCHSenv.cfg
cp usr/es/sbin/cluster/sa/hswizard/etc/lccluster /sapdb/<SID>/db/sap/lccluster

Insert into LCHSenv.cfg the following lines on both nodes (do not copy the description provided in < >):

vi /usr/es/sbin/cluster/sa/hswizard/etc/LCHSenv.cfg

```
SID=HLC                                   <The SAP System ID      >
LCADM=sdb                                 <the os user of the DB, usually sdb>
LCSERVER=is03d6                           <primary server>
TAKEOVER=is04d6                           <standby server>
LCSERVER_IP=lchlcip                       <service IP for APO connectivity>
LCSERVER_DB_IP=is03d6                     <primary server>
STANDBY_DB_IP=is04d6                      <standby server>
CONTROL_LOGIN=”-U HLC_XUSER”              < Xuser created eg. <SID>_XUSER>
LOCKFILE_DIR=/archive/sapdb/HLC_LOCK      <the NFS share>
MAXDB_PROGRAM_PATH=/sapdb/programs        <indep. Program path>
DBMCLI_COMMAND=/sapdb/programs/bin/dbmcli #<path to dbmcli>
LCBIN=/sapdb/programs/bin
RC_FOR_FAILED_MASTER_db_online=0          <default 0.
                                            This release does not support any
                                            other values.>
```

Copy lccluster from /usr/es/sbin/cluster/sa/hswizard/etc/lccluster into the instance directory /sapdb/<SID>/db/sap:

cp /usr/es/sbin/cluster/sa/hswizard/etc/lccluster /sapdb/<SID>/db/sap
chown sdb.sdba /sapdb/<SID>/db/sap/lccluster
chown sdb.sdba /usr/es/sbin/cluster/sa/hswizard/etc/LCHSenv.cfg

8.2 *Use Smart Assist to configure the cluster*

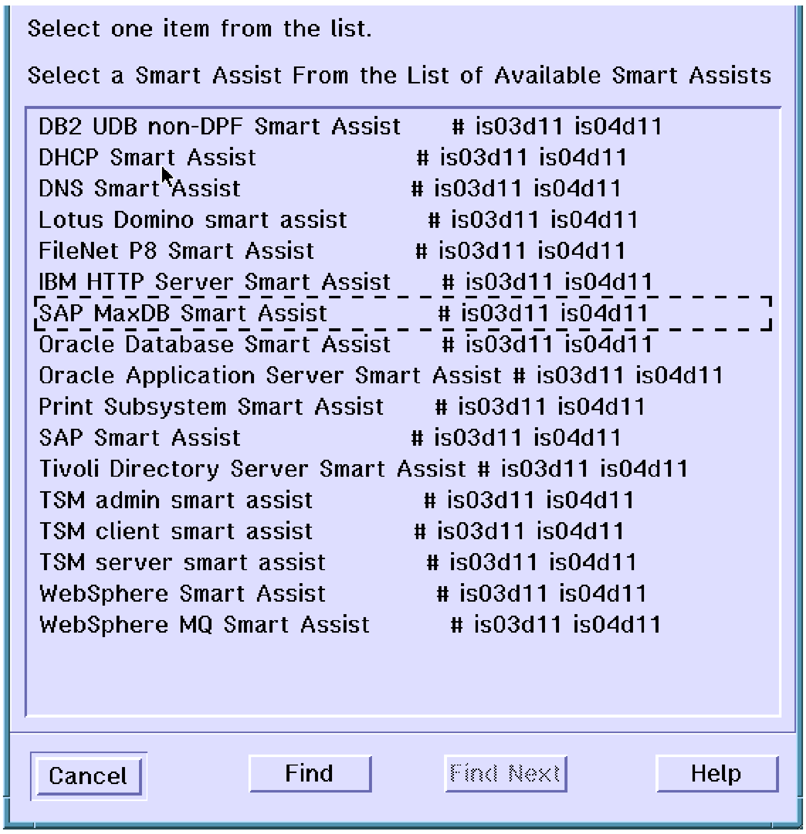
Preparation on both nodes as root:

- Start Hot standby pair – Master and Standby
- Start x_server on both nodes
- Start cluster nodes
- Log volume groups concurrent activated on both nodes
- Data volume groups online on both nodes
- If echo $NODE shows only one node run following command: unset NODE

NOTE: Only for PowerHA 7.1.1 SP1 set the following environment variables before starting smit:

export TRACE_FLAG="false"
export INFO_FLAG="false"
export KLIB_OUTPUT_CONSOLE="false"

These 3 environment variables will suppress unexpected trace output which has a bad influence on the execution of the Smart Assist of MaxDB.



Run Smart Assist:
In the Smart Assist panel choose “Add an Application to the PowerHA SystemMirror Configuration” and open it. Then select “SAP MaxDB Smart Assist” as shown in Figure 49.

Figure 49: Smart Assist – Start cluster configuration for liveCache

As shown below there are two ways of passing the parameters. Automatic will ask for some details in the smitty panel. Manual configuration picks same information from an xml sheet. For Details see the Smart Assist Manual.

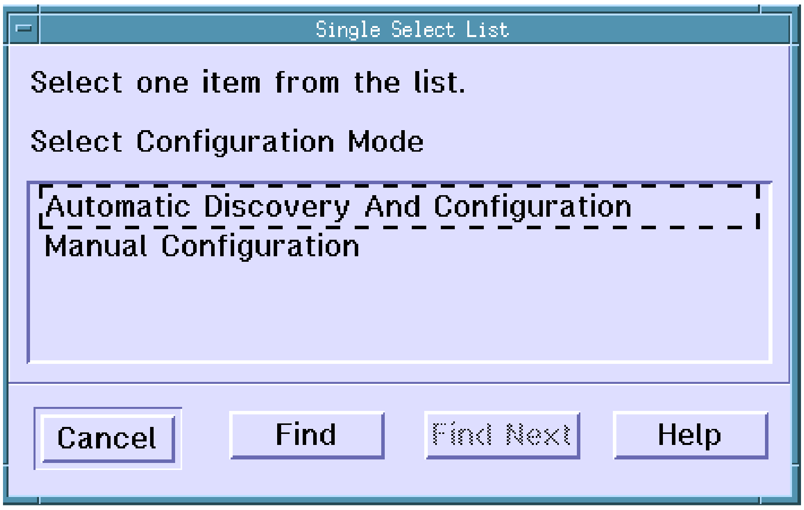


Figure 50: Smart Assist – Start automatic discovery

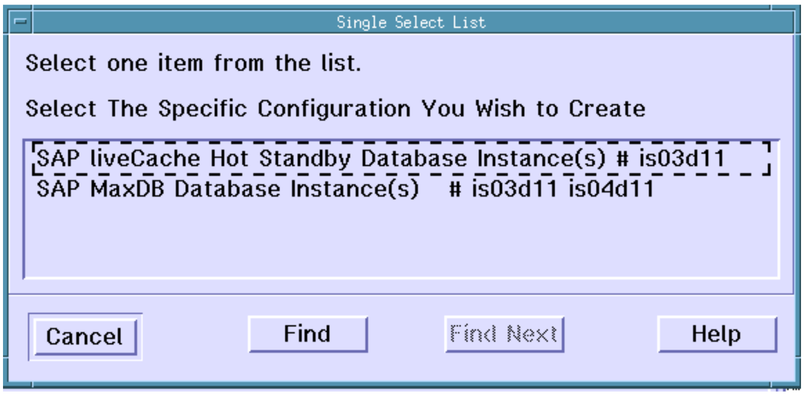


Figure 51: Smart Assist – Confirm to continue for HotStandby

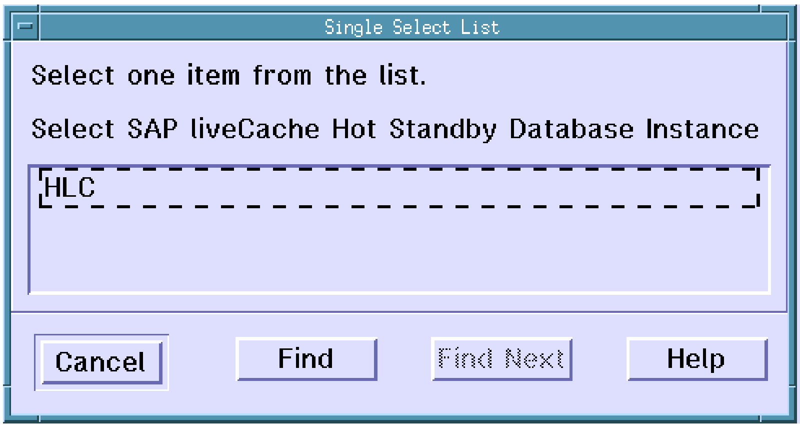


Figure 52: Smart Assist – Select the SID related to the installation

In this menu panel the parameters have the following meaning:

- **SAP liveCache HotStandby Instance DBM User** - this is MaxDB DBM or CONTROL user which was entered in the wizard panel (Figure 48) and in APO context, this is the APO DBM User. Insert here the XUSER entry created earlier for the DBM user.
- **Global Filesystem Mount Point** – this is the location of the NFS shared directory used to maintain the state information (see section 5.4.6).

Add a SAP MaxDB Hot Standby Database Instance	
* SAP liveCache Hot Standby Instance Name	HLC
* SAP liveCache Hot Standby Instance DBM User	[HLC_XUSER]
* liveCache Global Filesystem Mount Point	[/archive/sapdb/HLC_LOCK]
* Primary Node	is03d6
* Takeover Node(s)	is04d6
* Service IP Label	lchlcip
Netmask(IPv4)/Prefix Length(IPv6)	[]
* Data Volume Group(s)	[lcdatavg02 lcdatavg01]
* Log Volume Group(s)	[lclogvg02 lclogvg01]

Figure 53: Smart Assist – Provide information for discovery

Run a "Sync and Verify"

NOTE: depending on the PowerHA release you have to ensure that a log file for the APO interaction exists with the right ownership. Verify if the user sdb or the user you created instead has read and write access to a file called /var/hacmp/log/lccluster.log. If not create it (touch /var/hacmp/log/lccluster.log) and change permissions (chmod sdb.system /var/hacmp/log/lccluster.log)

Below is the command line equivalent of this panel:
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_maxdb_addInstance -H -i'HLC' -U'HLC_XUSER' -F'/archive/sapdb/HLC_LOCK' -o'is03d6' -t'is04d6' -s'lchlcip' -d'lcdatavg02 lcdatavg01' -l'lclogvg02 lclogvg01'

8.2.1 Manual correction

Service IP Label on wrong network interface

The automation assigns the service IP alias to the first network stored internally. Verify the assignment and adjust.
Run smitty cm_change_service_ip.select. In this case the IP Alias is lchlcip.

Change/Show a Service IP Label/Address	
.	[Entry Fields]
IP Label/Address	lchlcip *
New IP Label/Address	[lchlcip]
Netmask(IPv4)/Prefix Length(IPv6)	[24] *
Network Name	[net_ether_02]
Resource Group Name	RG_Master_HLC

Figure 54: Smart Assist – Provide information for discovery

Monitor Interval Considerations:

The monitor interval for master and standby may need to be increased accordingly to the requirements and load on the environment. They are currently set to the best known values resulting from the proof of concept testing and first implementation. If these are too sensitive, the cluster may failover as result of a monitor timeout. A timeout will be documented in the error logs. If this occurs, the monitor interval can be extended. This may also extend the failover time as the monitor will run less frequently. See section 9.6.

NOTE: Ensure no IPv6 enabled interfaces are assigned to the two nodes and the autoconf daemon for IPv6 is disabled.

Manual clean-up if necessary:

The wizard as well Smart Assist does a clean-up in case of a failure. But some items cannot be cleaned up in all cases.

1. Storage: delete Consistency group and FC Mappings
2. To remove the cluster configuration only use “Remove an Application from the PowerHA SystemMirror Configuration” from the Smart Assist submenu
3. On primary node: Disable first standby and then master
    a. Standby: hss_removestandby <host> (0)
    b. Master: hss_disable (preferred) or manual. The manual naming of parameters may vary between releases.
       dbmcli on HLC>param_directdel HotStandbyNodeName001
       dbmcli on HLC>param_directdel OfficialNodeName
       dbmcli on HLC>param_directdel HotStandbyStorageDLLPath
4. On both nodes remove the storage library from the dependent path:
   rm /sapdb/HLC/lib/lib64/libHSSibm2145.so
5. On both nodes remove the storage library configuration file:
   rm -rf /opt/ibm/ibmsap/HLC
6. On standby node vary all vgs off:
   varyoffvg datavg01;varyoffvg datavg02;varyoffvg datavg03
7. On standby node export all data vgs:
   exportvg datavg01;exportvg datavg02; exportvg datavg03
8. On both nodes clean up the APO to cluster integration Hook:
   rm /sapdb/HLC/sap/lccluster

8.3 Post processing for manually workaround the Smart Assist

The smart assist will fill the odm with all metadata the scripts need to store. Having used Smart Assist odmget HACMPsa_metadata is showing the values listed below. The ones required for the actual runtime of the scripts are marked yellow. They are referenced to later to get the required values to use when configuring the script parameters.

```
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "SMARTASSIST_ID"
        value = "MaxDB_7.6"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "COMPONENT_ID"
        value = "MaxDB_Hot_Standby"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "LC_XUSER"
        value = "HLC_XUSER"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "INSTANCE_DBM_USERID"
        value = "HLC_XUSER"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "MAXDB_PROGRAM_PATH"
        value = "/sapdb/programs"
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "RESOURCE_GROUP"
        value = "RG_Master_HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "APPLICATION_NAME"
        value = "liveCache_Master_HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "LC_LOCK_DIR"
        value = "/archive/sapdb/HLC_LOCK"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "INSTANCE"
        value = "HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "FILE_COLLECTION"
        value = "HLC_instFiles"
```

```
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "RESOURCE_GROUP"
        value = "RG_Log_HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "APPLICATION_NAME"
        value = "liveCache_log_HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "LC_LOCK_DIR"
        value = "/archive/sapdb/HLC_LOCK"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_log_HLC"
        name = "INSTANCE"
        value = "HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "SMARTASSIST_ID"
        value = "MaxDB_7.6"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "COMPONENT_ID"
        value = "MaxDB_Hot_Standby"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "LC_XUSER"
        value = "HLC_XUSER"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "INSTANCE_DBM_USERID"
        value = "HLC_XUSER"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Master_HLC"
        name = "MAXDB_PROGRAM_PATH"
        value = "/sapdb/programs"
        reserved = 0
```

```
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "SMARTASSIST_ID"
        value = "MaxDB_7.6"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "COMPONENT_ID"
        value = "MaxDB_Hot_Standby"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "LC_XUSER"
        value = "HLC_XUSER"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "INSTANCE_DBM_USERID"
        value = "HLC_XUSER"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "MAXDB_PROGRAM_PATH"
        value = "/sapdb/programs"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "RESOURCE_GROUP"
        value = "RG_Standby_HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "APPLICATION_NAME"
        value = "liveCache_Standby_HLC"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "LC_LOCK_DIR"
        value = "/archive/sapdb/HLC_LOCK"
        reserved = 0
HACMPsa_metadata:
        application_id = "liveCache_Standby_HLC"
        name = "INSTANCE"
        value = "HLC"
        reserved = 0
```

The entries above have to be added to the ODM database by using the odmadd command. Build a file with all options and overwrite all highlighted options by using your own definition. All other options can leave as default. For all options you have to change the <SID> to yours.

9 Manual tasks for customization

9.1 Storage library installation and setup [Manual]

This section covers the installation of the libHSS storage library itself, and the steps necessary to ensure it is found by SAP liveCache, executable by the liveCache user and configured properly for the liveCache instance it is to support.

9.1.1 Installation of the HSS library on both nodes

The library is automatically installed with the Smart Assist package. The installation will be performed into /opt/ibm/ibmsap.
Create a directory mkdir */opt/ibm/ibmsap/HLC*

```
# ls -l /opt/ibm/ibmsap
total 600
drwxr-xr-x  2 root   system   256 Nov 30 10:27      HLC
-rw-r--r--   1 root   system   4405 Jul 11 15:29     RTEHSS_config_sample.txt
drwxr-xr-x  4 root   system   256 Oct 14 01:43      connectors
-rwxr-xr--  1 sdb    sdba     145531 Jul 12 23:16   libHSSibm2107.so  (DS8000 library - 2107)
-rwxr-xr--  1 sdb    sdba     147811 Jul 12 23:16   libHSSibm2145.so  (SVC library - 2145)
```

The shared library must be copied by the wizard into the liveCache dependent path where it can be found by liveCache. This path can be found in *<dependent program path>/lib/lib64.* The owner of the library is the liveCache user and group, sdb.sdba.

9.1.2 Configuration of the library

The wizard will create the configuration file. This section provides details to understand its capabilities. The following example explains the profile and the configuration necessary for SAN Volume Controller attached storage.
In the case that a manual configuration is being followed, this action must be done in preparation for the initializing the HotStandby cluster. The library is used to drive the storage integration to the cluster.

The default directory for the library components is */opt/ibm/ibmsap*. Under this directory the library, the connector directory, and a sample configuration file *RTEHSS_config_sample.txt* can be found. The sample configuration should be copied to the /opt/ibm/ibmsap/<SID> directory for configuration for this specific HotStandby. The configuration file RTEHSS_config.txt is placed under */opt/ibm/ibmsap/SID* and owned by sdb.sdba. Be sure to verify the authorizations and ownership. The example that follows shows the configuration for an SVC implementation. The parameters which are expected to be modified for the specific installation are highlighted in yellow. Below each is a note which describes the purpose.

cp /opt/ibm/ibmsap/RTEHSS_config_sample.txt /opt/ibm/ibmsap/<SID>/RTEHSS_config.txt
chown sdb.sdba opt/ibm/ibmsap/<SID>/RTEHSS_config.txt

A valid SVC configuration can contain following parameters:

```
########## RTEHSS_config.txt ######################################################
# SAP liveCache / MySQL MaxDB
# RunTimeEnvironment HotStandbyStorage configuration file
#################################################################################
# CSmode: Copy Server Services
# Choose FC if only one storage system is used, flash copy only the data volumes
#
CSmode FC
#
# Ibmclidir: directory of the storage command line interface
# DS8xxx: path to the DScli
# SVC: path to the storage connector interface
#
#Ibmclidir /opt/ibm/DScli
Ibmclidir /opt/ibm/ibmsap/connectors/HSS2145
<Note: HSS2145 is the library for SVC>
# Ibmsapapodir: the install directory of the storage dependent runtime library
Ibmsapapodir /opt/ibm/ibmsap
<Note:Is already set to the default. Nothing to change.>
#
# MlCLogVdiskID: ID of Log volume on Master
# DS8xxx: ID of volume (four digit hex, e.g. MlCLogVdiskID 1400)
# SVC: vdisk_id or vdisk_name
MlCLogVdiskID shared_log
<Note: The volume can be specified either by name or by ID as shown in the SVC>
#
# MlCDataVdiskID: ID of Data volume on Master liveCache Server
# DS8xxx: ID of volume, for multiple volumes use a comma separated string
#    (four digit hex, e.g. MlCDataVdiskID 1401, 1403)
# SVC: vdisk_id or vdisk_name, for multiple volumes use a comma separated string
MlCDataVdiskID node1_data01
<Note: First of the flash copy LUNs used as data disks. We are starting with node1 as primary>
#
# SlCLogVdiskID: on 1st Standby liveCache Server, ID of Log volume
# DS8xxx: ID of volume (four digit hex, e.g. SlCLogVdiskID 1400)
# SVC: vdisk_id or vdisk_name
#
SlCLogVdiskID shared_log
<Note: Both servers are using the same log  LUN so there is only one ID>
# SlCDataVdiskID: on 1st Standby liveCache Server, ID of Data volume
# DS8xxx: ID of volume, for multiple volumes use a comma separated string
#    (four digit hex, e.g. SlCDataVdiskID 1401, 1403)
# SVC: vdisk_id or vdisk_name, for multiple volumes use a comma separated string
#
SlCDataVdiskID node2_data01
<Note: 2nd of the flash copy LUNs used as data disks. We are starting with node2 as 2ndary>
# CSaIP: Copy Server IP address
# DS8xxx: IP address of SSPC
# SVC < 6.x: IP address of SSPC
# SVC >= 6.x: IP adress of SVC
#
CSaIP 9.xxx.xxx.xxx
<Note: This is the ip address of the SVC which is used by the connectors through SSH communication.
The address should be complete, it is purposely not shown here. >
# CSaUID: Copy Server User ID (admin)
# DS8xxx: user ID (perform copy service task)
# SVC: ID name for SSH connection to SVC
#
```

CSaUID admin
*<**Note:** This is the name which is used by the SSH connection to the SCV. It must be admin.>*
 # CSapwd: Copy server password
 # DS8xxx: password of user ID
 # SVC: leave this field blank
#
CSapwd
*<**Note:** Not relevant for SVC. It is configured in the SSH setup.>*
 # CSbIP: IP address of backup master console (SVC2) / hmc2.
 # User ID and password must be the same for both servers.
#
CSbIP
*<**Note:** In our example we have no backup master console.>*
 # DSdevID: ID of DS8xxx storage dev
 # SVC: leave this field blank
#
DSdevID
*<**Note:** Not relevant for SVC.>*
 # HSS_NODE_00x: lists all Hot Standby nodes available in this section
 # (with x from 1 to 3, max 3 nodes).
 # Retrieve the node name by calling 'uname -n'
#
HSS_NODE_001 is03d6
HSS_NODE_002 is04d6
*<**Note:** Host names of the HotStandby LC servers . primary and 2ndary.>*
 # EstDataCST_00x_00y: Defines the copy server tasks, in case where one storage
 # system uses flash copy to copy the Data volumes from HSS_NODE_00x to HSS_NODE_00y
 # DS8xxx: specify the sequence number which will be used to copy data volume from
 # current MASTER (HS_NODE_00x) to requesting STANDBY (HS_NODE_00y).
 # The sequence number is a four digit hex number (0000 - FFFF). The tasks will
 # be built up dynamically.
 # e.g. EstDataCST_001_002 1020
#
 # SVC: specify unique name which will be used to name the dynamically created FC
 # relation to copy data volume from current MASTER (HS_NODE_00x) to
 # requesting STANDBY (HS_NODE_00y) (the name shouldn't begin with a digit).
 # The FlashCopy relation will be built up dynamically, do not create it!
 # e.g. EstDataCST_001_002 FC_1_2
#
 # establish FC between is03d1 (d3) and is04d1 (d4)
EstDataCST_001_002 d3_d4
EstDataCST_002_001 d4_d3
*<**Note:** Names given for the flash copy consistency groups used to establish flash copy . One in each direction>*
 # TermDataCST_00x_00y: Define the task names to terminate the FlashCopy relation
 # between volumes of HS_NODE_00x to HS_NODE_00y
 # DS8xxx: set the same sequence numbers than the one specified to establish the FlashCopy
 # relation (see EstDataCST_00x_00y).
 # SVC: set the same names than the one specified to establish the FlashCopy relation
 # (see EstDataCST_00x_00y).
#
TermDataCST_001_002 d3_d4
TermDataCST_002_001 d4_d3
*<**Note:** Names given used to remove . withdraw . a FlashCopy mapping in order to restart . again depending of direction must be the same as the actually establish names.>*

9.1.3 Addressing multiple HMCs

A second redundant HMC can be specified configuring the variable CSbIP in the RTEHSS_config.txt file on both nodes.

9.1.4 Addressing multiple ssh keys

In case different ssh keys are used on the nodes the RTEHSS_config.txt CSaUID and if used the CSbUID parameter is not the same on both nodes anymore. These values become host specific and have to be maintained manually over the whole life cycle of the cluster.

- Place the RTEHSS_config.txt on both nodes
- Adapt the value of CSaIP and CSbIP accordingly to the ssh key on that node
- Maintain it over the whole life cycle
- Never put this file into a PowerHA file collection from now onwards

9.2 Configuring SAP liveCache HotStandby manually

This section covers methods used to configure SAP liveCache into a HotStandby pair in case the wizard based setup cannot be used.

Configuring the HotStandby SAP liveCache requires setting database parameters and invoking actions within the primary liveCache. This activity requires access to the primary liveCache instance. These activities can be done using the command line (dbmcli) or using the SAP tool, Data Base Manager GUI (DBMGUI). Please see chapter 13.1, “Related documents and sources of further information”, for more information on the command line and DBMGUI.

1. Start SAPDB x_server on both nodes
2. Setup primary sapdb node for HotStandby
    a. Set primary node offline
    b. Enable HSS storage library
    c. Set primary node online
    d. Get HSS node information
    e. Set liveCache parameters use the following statement
    f. Verify parameters
    g. Set initialize the master use the following SAPDB commands:
3. Setup standby sapdb node for HotStandby
    a. Add standby node
    b. Check HSS node configuration with
    c. Set standby node offline
4. Set standby node in standby mode

9.3 Checklist before activating HotStandby

- Ensure all disks are properly imported and active (the concurrent access volumes for the liveCache log are activated by the PowerHA resource group see chapter 9.5.1).
- Verify raw devices used for log and data have authorization for sdb.sdba on both nodes.
- Ensure that the installed version of the liveCache software is the same on both nodes and at least kernel version 7.7.07.
- Verify HSS library is either copied to or linked to dependent path (usually: /sapdb/<SID>/db/lib/lib64) on both nodes. Also validate permissions.

- Check HSS profile (/opt/ibm/ibmsap/<SID>/RTEHSS_config.txt) and library has been configured on the primary and copied to the secondary node.
- Ensure HSS profile is executable and has owner sdb.sdba.
- Be sure you have stopped any FlashCopy tests between source and targets for HotStandby data volumes on storage level.
- Activate the service IP alias and the x_server by starting the MASTER resource group with lock set to LC10.stop. If done manually be sure to deactivate it later.

Additional initialization considerations
The table HOTSTANDBYGROUP is only created in the database when initialized from APO (not dbmcli or other DBM tools). In order for the scripts to work, the liveCache needs to be attached to the APO and initialized from application layer.

9.3.1 Configuring HotStandby from the GUI (DBMGUI)

The next steps are done from an active primary that has been initialized from APO. The primary is configured to accept a standby and knowing its service IP alias. Then the standby server is setup from the primary. For this step all previous configuration steps must have been completed, including the initial FlashCopy and volume group import to the standby server which is now to be initiated as standby.

Step 1) Start x_server on both nodes and activate service IP alias on the primary.

Section 9.6 describes the tasks to configure the PowerHA resource groups accordingly to bring the required infrastructure (IP and listener) up. All three resource group are required. For manual start up ensure to reset the settings. Only the log volume resource group RG_Log_<SID> must be online on both nodes.
x_server: <independent path>/bin/x_server start
IP alias: ifconfig en<x> alias <service IP alias>

Step 2) Log into the primary liveCache in DBMGUI

Using the DBMGUI, open a connection to the primary liveCache. The button to add an instance can be found in the right bottom frame when starting DBMGUI.

Step 3) Go to the HotStandby submenu

Now select the “Hot Standby” which can be found on the left hand side (Figure 55) and enable the instance for HotStandby (Figure 56).

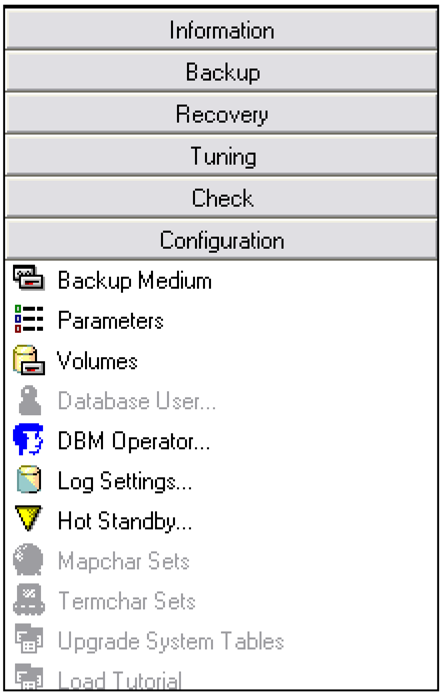


Figure 55: DBMGUI – Entry menu

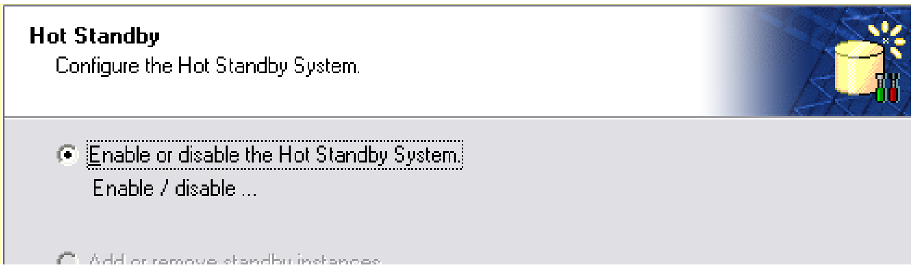


Figure 56: DBMGUI – configure HotStandby system entry screen

Step 4) Configure the library access

If the instance is active, the opportunity to place liveCache in offline status and continue the activation will be given. The virtual server name should be the name of the rotating IP alias.

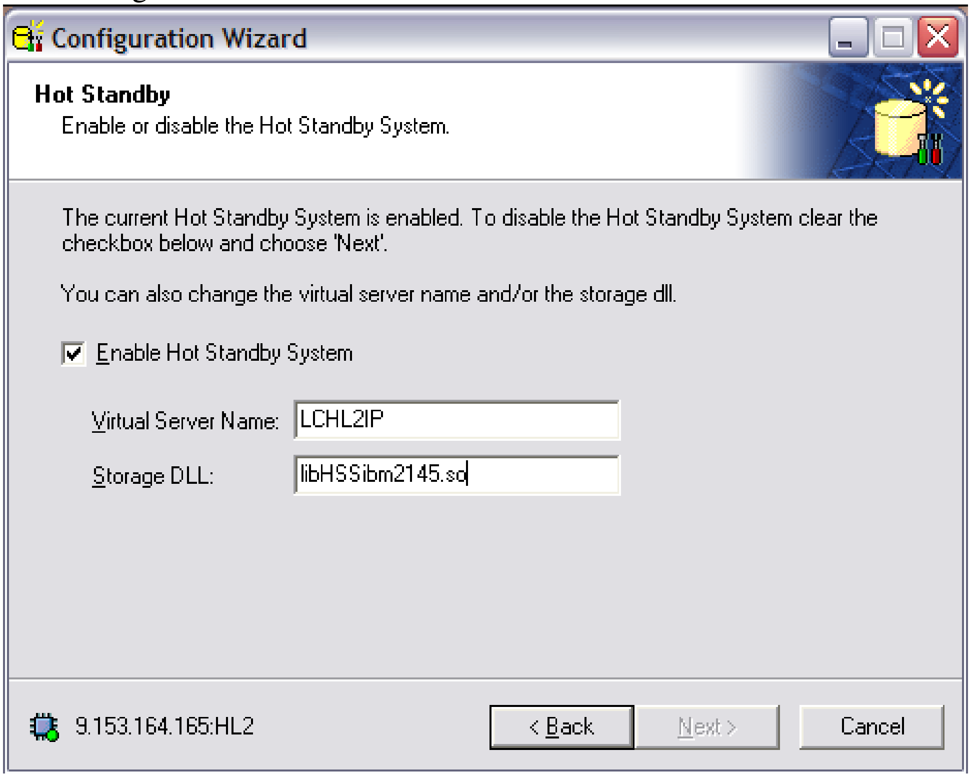


Figure 57: DBMGUI, identifying the storage library

Step 5) Enable the Master instance.

In the next screen press “Start” to enable the instance for HotStandby capability.

Add the name of the storage shared library which supports the type of storage server being used for this configuration. Figure 58 shows the name of the SAN Volume Controller library as used in the reference installation.

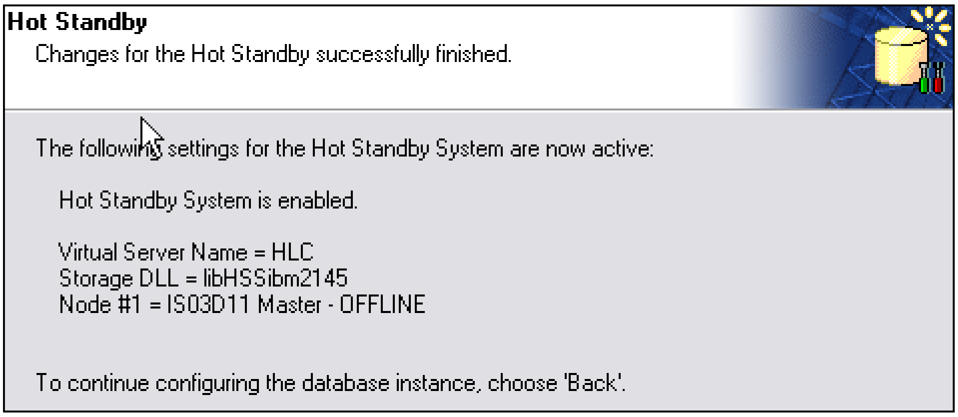
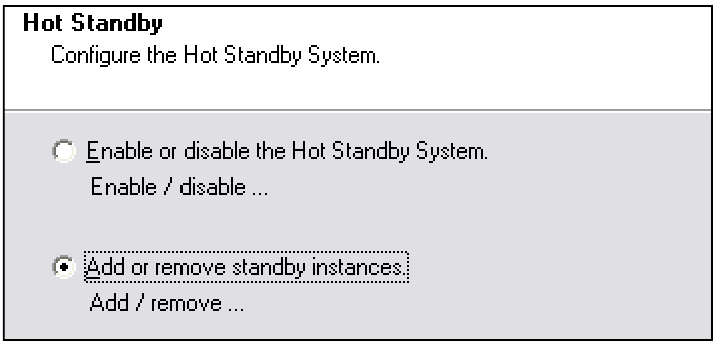


Figure 58: DBMGUI – setup is enabled

Step 6) Creating HotStandby secondary database



Add the standby instance. Press the "Hot Standby" button to open the configuration wizard as done for the master. Then select "Add or remove standby instances".

Figure 59: DBMGUI – adding the standby instance from the master liveCache

Select the new standby instance. The available installation path selected as shown in Figure 60 is the dependant path.

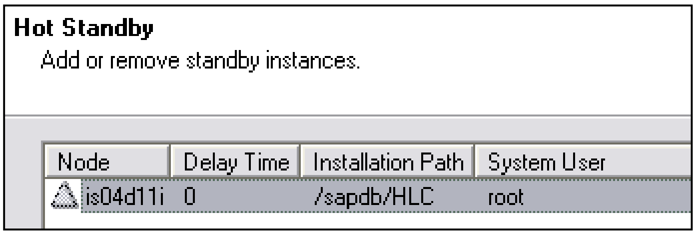


Figure 60: DBMGUI – identifying the installation path which is to be used

Step 7) Initiate the new standby.

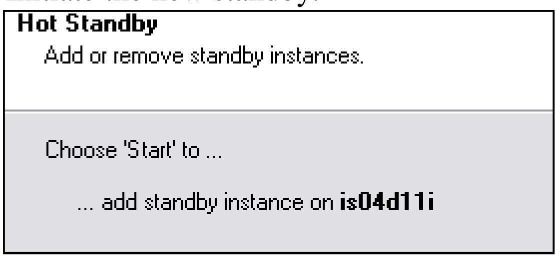


Figure 61: DBMGUI – starting the standby instance creation

9.3.2 Configure HotStandby primary from the command line (dbmcli)

The command line activation of a HotStandby pair is equivalent to the process performed through DBMGUI. The virtual server name is again the virtual service IP alias for the liveCache to allow APO connecting to the primary instance. Also the storage library name which will be used to integrate the storage functionality of the DS8000 or SVC is specified. Ensure that prerequisites listed in section 9.3 are fulfilled.

Start the x_server on both nodes:
/sapdb/programs/bin/x_server

Login on the primary server using the command line tool dbmcli:
dbmcli -d <SID> -u <user>,<password>

Step 1) Set primary database offline to allow new configuration.
From dbmcli on primary server enter following DB command:
dbmcli > db_offline

Step 2) Enable HotStandby and configure the target library which will be used and the IP alias.
From the dbmcli on the primary node enter the following database command:
dbmcli >hss_enable lib=<hs_storage_dll> node=<virtual_server_name>
example: dbmcli > hss_enable lib=libHSSibm2145.so node=lchl2ip

Step 3) Set liveCache primary online (db_online)
dbmcli > db_online

Step 4) Verify the results from dbmcli on the master node:
From the dbmcli on the primary node enter the following database command:
dbmcli >hss_getnodes
OK

HotStandbyStorageDLLPath	*libHSSibm2145.so*	
OfficialNodeName	*LCHL2IP*	
CURRENT_NODE	*IS03D1*	
HotStandbyNodeName001	*IS03D1*	*0 ← The primary node is now displayed along with the storage library and IP to use.*

For a fresh installation you can initialize the master instance as follows using the dbmcli:
dbmcli > db_admin
dbmcli > db_activate superdba,<passwd>   ← allow superdba to execute sql statements required
by the PowerHA scripts and APO administration
dbmcli > load_systab -u superdba,<passwd> ← load tables that they exist and can be accessed
dbmcli > user_sap <sapr3 user>,<passwd>   ← schema user – this is the sql user definition.
This is setting the user which must be configured in the APO integration as the standard liveCache user. Be sure
these two matches.
dbmcli > load_lcapps <sapr3 user>,<passwd>

NOTE: The initialization done here does not replace the initialization from APO! This activity is still necessary as soon as the APO integration is completed.

9.4 Configure Standby from the command line (dbmcli)

For this step, you have already installed the liveCache software on the target standby node using the same independent path as used for the primary and setup the users and group id's to match the primary. The steps to validate the paths are described in section 6. As a prerequisite the primary has to be already activated and online as described in section 0.

On the **primary node** open the dbmcli command prompt and follow the following steps to activate the standby instance. In the following the variable <standby host> refers to the name of the secondary node as shown by uname -n.

Step 1) Ensure the standby is offline on node 2.

Step 2) Introduce the standby to the primary server from the primary node

dbmcli > hss_addstandby <standby host> login=<os_user>,<pwd> [path=<dependent_path>]
example: hss_addstandby is04d1 login=root,<password> path=/sapdb/HL2/db

Step 3) Check configuration from primary server

dbmcli > hss_getnodes
OK

HotStandbyStorageDLLPath	*libHSSibm2145.so*
OfficialNodeName	*LCHL2IP*
CURRENT_NODE	*IS03D1*
HotStandbyNodeName001	*IS03D1*
HotStandbyNodeName002	*IS04D1 ← The second node is now displayed as well*

Step 4) Initialize the standby instance from the standby node (node2)

When activating the HotStandby for the first time, it may be a good idea to force the FlashCopy initialization through dbmcli. This ensures that the library and the entire underlying infrastructure are functional. This must be done on the new standby node.

Login to the standby instance using dbmcli on the standby server and execute the following DB commands:

dbmcli > db_admin
dbmcli > db_execute init standby

If this is successful, continue with the next steps. Otherwise look at the protconv output on the standby and start tracking the errors. The tool, protconv, can be invoked by the dbm administrator or sdb user to look at kernel messages. It should be invoked from the liveCache instance work directory (wrk).

Step 5) Starting the standby relationship from the master server.

The standby is always logically started from the master as this activates the synchronization communication flow. From the dbmcli interface on the master server, enter the following DB commands:

dbmcli > hss_execute <standby node> db_offline
dbmcli > db_standby <standby node>

example: dbmcli > hss_execute is04d1 db_offline
dbmcli >db_standby is04d1

Removing a standby instance

To remove the standby from command line, shutdown the standby and use the dbmcli command:

dbmcli > hss_removestandby <standby host>

Removing a master instance

Details are listed in the manual cleanup instructions in section 0.

9.5 Customizing PowerHA for HotStandby liveCache

The following chapters focus on HotStandby specifics. Information on the basic setup of the cluster is covered in section 5.4.3. For more detailed information please see the PowerHA RedBook listed in chapter 13.1.

First this chapter will show how the cluster is configured to support a highly available concurrent log volume by its own resource group. The following chapters discuss IP considerations and configuration as well as the other two resource groups required to control the master and the standby instance along with the required infrastructure.

9.5.1 Preparation of the concurrent log volume [Automated]

Depending on using the Smart Assist Wizard of following a manual approach the flow is slightly different. First the flow using the wizard is described followed by the tasks to be performed going a manual path. The main difference in using automation is the time creating the resource group and the method to bring it online as a start point for the liveCache installation and the wizard to work.

For both ways: It is recommended to use PowerHA functionality to create the log volume group. This requires having the base cluster configured and started. The advantage is that this ensures having the correct set up on both nodes with a minimum of effort. Everything defined will automatically be applied on the standby node.

Enter *smitty cl_admin* → *Storage* → *Volume Groups* → *Create a Volume Group*. In the following screen select both nodes and press enter. In the next two panels select first all nodes on which the volume group should become online later. The next screen shows shared disks – meaning having the same pvid on both nodes. In case the initial FC has transported a pvid of a data volume to the other node they will show up as well. Be careful not to mix up the disks in a FlashCopy relationship and the actually physically shared disk. Here the physically shared disk(s) is/are to be selected. The next screen asks for the volume type. In our example we selected “Scalable”.

NOTE: If you do not find the disks you expect (independent to the PowerHA release), run cluster discovery and restart creating the volume group. (*Cluster Nodes and Networks →Discover Network Interfaces and Disks*).

Prepare for Wizard - the Smart Assist will create the resource group in a later step. Never the less the log volumes must be online on both nodes and configured through C-Spoc.

```
+-----------------------------------------------------------------------------+
| Node Names                                                                  |
| Move cursor to desired item and press F7.                                   |
| Press Enter AFTER making all selections.                                    |
| > is03d6                                                                    |
| > is04d6                                                                    |
+-----------------------------------------------------------------------------+
```

Figure 62: Cluster Nodes

NOTE: in the following screen all disks with the same PVID are listed. Be careful to pick only valid log disks.

```
+--------------------------------------------------------------------+
| Physical Volume Names                                              |
| Move cursor to desired item and press F7.                          |
| Press Enter AFTER making all selections.                           |
| > 00f70ae9be5dd823 ( hdisk3 on all cluster nodes )                 |
+--------------------------------------------------------------------+
```

Figure 63: Concurrent log disk found on both nodes

In the next panel choose the volume group type you desire. In the PoC we used “Scalable” or “Big” vgs.

Figure 64 illustrates how the volume group in the reference installation was defined in case of planning for Smart Assist and wizard. In case the wizard is not used the Resource group name can be inserted into the menu panel.

Node Names	is03d6,is04d6
Resource Group Name	[<empty using Smart Assist later>]
PVID	00f70ae9be5dd823
VOLUME GROUP name	[lclogvg]
Physical partition SIZE in megabytes	128
Volume group MAJOR NUMBER	[105]
Enable Fast Disk Takeover or Concurrent Access	Concurrent Access
Volume Group Type	Big <not mandatory a Big>
CRITICAL volume group?	no

Figure 64: PowerHA – configuration of the concurrent log volume group if using the wizard

The key setting is to select in the row “*Enable Fast Disk Takeover or Concurrent Access*” to “*Concurrent Access*”.

The next step is to create the correct sized logical volumes (section 5.4.4). They will be named in this document as following:

lcloglv01, … lcloglv<nm>

C-SPOC is the tool of choice to create the logical volumes. For the manual approach C-SPOC can be used, but other ways creating it as a raw device are fine as well. The entry point is C-

SPOC (smitty cl_admin). Following the path *Storage* → *Logical Volumes* → *Add a Logical Volume* allows the previously defined volume group to be selected. In the next screen we selected the primary node as reference node.

Figure 65: Log Disk

Resource Group Name	<Not in a Resource Group>
VOLUME GROUP name	lclogvg
Node List	is03d6,is04d6
Reference node	is03d6
* Number of LOGICAL PARTITIONS	[399]
PHYSICAL VOLUME names	hdisk3
Logical volume NAME	[lcloglv1]
Logical volume TYPE	[raw]
POSITION on physical volume	outer_middle
RANGE of physical volumes	minimum
MAXIMUM NUMBER of PHYSICAL VOLUMES to use for allocation	[] #
Number of COPIES of each logical partition	1
Mirror Write Consistency?	active
Allocate each logical partition copy on a SEPARATE physical volume?	yes
RELOCATE the logical volume during reorganization?	yes
Logical volume LABEL	[]
MAXIMUM NUMBER of LOGICAL PARTITIONS	[512]
Enable BAD BLOCK relocation?	yes
SCHEDULING POLICY for reading/writing logical partition copies	parallel
Enable WRITE VERIFY?	no
File containing ALLOCATION MAP	[]
Stripe Size?	[Not Striped]
Serialize I/O?	no
Make first block available for applications?	no
User ID	sdb
Group ID	sdba
Permissions	[]

Figure 66: configuration of the concurrent log logical volume (wizard)

Now run a sync and verify in the cluster to make the change available to all nodes. In case of the wizard the VG is to be brought online on both nodes using varyonvg –c.

Be sure to set correct raw device permission as described in section 5.4.5. This has to be verified on both nodes.

9.5.2 Preparation of the concurrent log volume [Manual]

The preparation of the log volume group – RG_Log_<SID> - requires several steps. The log will be defined as an enhanced concurrent volume in concurrent mode, which will allow it to be accessible on both nodes simultaneously. This concurrent access is provided by PowerHA functionality and available once the cluster and the corresponding resource group is active.

Create a Scalable Volume Group

Node Names	*is04d6,is03d6*
Resource Group Name	*[RG_Log_<SID>]*
PVID	*00f641cd84914648*
VOLUME GROUP name	*[lclogvg]*
Physical partition SIZE in megabytes	*4*
Volume group MAJOR NUMBER	*[43]*
Enable Fast Disk Takeover or Concurrent Access	***Concurrent Access***
Volume Group Type	*Scalable*
CRITICAL volume group?	*no*

[…]

Figure 67: Scalable Volume Group for log volume

The key setting is to select in the row "*Enable Fast Disk Takeover or Concurrent Access*" to "*Concurrent Access*".

Now run a sync and verify in the cluster to make the resource group change available to all nodes. Then bring the resource group RG_Log_<SID> online and validate using *lspv* if the vg is activated in concurrent mode on both nodes.

Example output to check on both nodes:

#lspv
[…]
hdisk12 00f641d4d95b8fe0 lctestvg ***concurrent***

The corresponding resource group RG_Log_<SID> will show following characteristics:

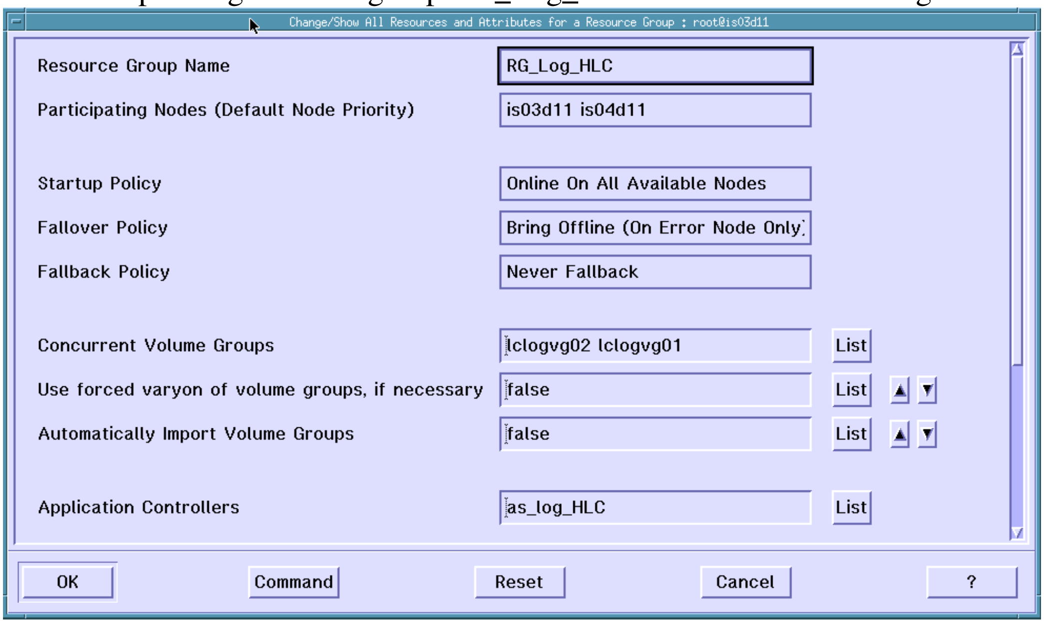


Figure 68: PowerHA – Resource group definition for RG_Log_<SID>

When the volume group is defined the next step is to create a logical volume. C-SPOC delivers this functionality as described before, along with the preparation for the wizard. But here all other approaches creating a raw logical volume is fine.

NOTE: The smitty menu does not allow raw type logical volume to be explicitly created. The type jfs is used instead. This will create a raw logical volume accessible in */dev/rlcloglv<no>*.

Now the permission should be changed as described in section 5.4.5 and verified on both nodes.

9.5.3 Configuration and preparation of the data volume group and logical volume

Use C-SPOC, smitty or command line to create the data volumes as a local resource.

9.6 Modeling the resource groups – Overview

This section shows examples of how the resource groups are defined and modelled against each other. There are three resource groups which must be defined: the prerequisite resource group that contains the log (RG_Log_<SID>), the master (RG_Master_<SID>) and the standby (RG_Standby_<SID>) resource group.

1. Add Resource Groups
    a. RG_Log_<SID>
    b. RG_Master_<SID>
    c. RG_Standby_<SID>
2. Configure Online on different nodes Dependency
3. Configure Resource Group Acquisition and Release Ordering
4. Configure Service IP Labels/Addresses
5. Add Application Controller Scripts
    a. as_log_<SID>
    b. as_master_<SID>
    c. as_standby_<SID>
6. Add a Custom Application Monitor
    a. mon_master_<SID>
    b. mon_master_startup_<SID>
    c. mon_standby_<SID>
    d. mon_log_<SID>
7. Change/Show Resources and Attributes for a Resource Group
    a. RG_Log_<SID>
    b. RG_;aster_<SID>
    c. RG_Standby_<SID>

9.7 Create Resource Groups

Create the three resource groups as shown in the following three pictures. Choosing for the Master and Standby “Online on first Available Node” brings flexibility into the solution you do not want to miss. By that the handling of a startup of a single node is getting the expected results.

cmd> /usr/es/sbin/cluster/utilities/claddgrp -g 'RG_Log_HLC' -n ‘is04d6 is03d6 ' -S 'OAAN' -O 'BO' -B 'NFB'

Change/Show Nodes and Policies for a Resource Group	
	[Entry Fields]
Resource Group Name	RG_Log_HLC
New Resource Group Name	[]
Participating Nodes (Default Node Priority)	[is04d6 is03d6]
Startup Policy	Online On All Available
Fallover Policy	Bring Offline (On Error Node Only)
Fallback Policy	Never Fallback

Figure 69: Resource group policy for log

cmd> /usr/es/sbin/cluster/utilities/claddgrp -g 'RG_Master_HLC' -n is04d6 is03d6' -S 'OFAN' -O 'FNPN' -B 'NFB'

Change/Show Nodes and Policies for a Resource Group	
	[Entry Fields]
Resource Group Name	RG_Master_HLC
New Resource Group Name	[]
Participating Nodes (Default Node Priority)	[is03d6 is04d6]
Startup Policy	Online On First Available Node
Fallover Policy	Fallover To Next Priority Node In The List
Fallback Policy	Never Fallback

Figure 70: Resource group policy for Master

cmd> /usr/es/sbin/cluster/utilities/claddgrp -g 'RG_Standby_HLC' -n ' is03d6 is04d6' -S 'OFAN' -O 'FNPN' -B 'NFB'

Change/Show Nodes and Policies for a Resource Group	
	[Entry Fields]
Resource Group Name	RG_Standby_HLC
New Resource Group Name	[]
Participating Nodes (Default Node Priority)	[is04d6 is03d6]
Startup Policy	Online On First Available Node
Fallover Policy	Fallover To Next Priority Node In The List
Fallback Policy	Never Fallback

Figure 71: Resource group policy for standby

9.8 Mutually Exclusive Resource Groups: Online on different nodes

In the section below, the resource group distribution over the two nodes is implemented. This forces the standby and master resource groups to different nodes.

The master and standby resource groups are rotating and mutually exclusive. This dependency ensures that master and standby run on different nodes and that as soon a second node comes up the slave can be automatically restarted on the second server. PowerHA allows the resource groups to be prioritized. By giving the master resource group a higher priority the cluster always keeps this online as long there is at least one cluster node. Only if there is a second node the slave is brought online.

Resource group dependency submenu in "*smitty hacmp*":
Cluster Applications and Resources → *Resource Group Run-Time Policies* → *Configure Dependencies between Resource Groups* → *Configure Online on Different Nodes Dependency*

In this node dependency, the master resource group is always handled with higher priority than the standby.
cmd:>/usr/es/sbin/cluster/utilities/clrgdependency -t'ANTICOLLOCATION' -u -hp'RG_Master_HLC' -ip'RG_Standby_HLC' -lp''

```
Configure Online on different nodes Dependency

                                              [Entry Fields]
 High Priority Resource Group(s)              [RG_Master_HLC]
 Intermediate Priority Resource Group(s)      [RG_Standby_HLC]
 Low Priority Resource Group(s)               []
```

Figure 72: PowerHA – Configure online on different node dependencies

9.9 Service IP alias for the liveCache Master instance

In this section the service IP alias is defined as a rotating resource. This resource is part of the master resource group and follows the active online database.

Naming convention: lc<SID>ip

Preparation (this task has been already performed following the providing flow):
For the clusters, the cluster addresses including alias addresses are maintained in the local */etc/hosts* file. The search which may include any additional name servers must always search the local */etc/hosts* first. This avoids external influence on the addresses used by the clusters. Add the service IP alias into both nodes' */etc/hosts* file.

Add the service IP alias into the cluster:
The following configuration steps will offer the available networks and IP for selection. As a prerequisite the cluster topology has to be configured and the IP addresses must be listed in */etc/hosts* on both nodes. The proof of concept has used the internal network for all inter-node communication. As a result the net_ether_02 is selected for the proof of concept as this represents the internal network.

The smitty submenu is: *Cluster Applications and Resources* → *Resources* → *Configure HACMP Service IP Labels/Addresses* → *Add a Service* IP Label/Address

```
              Change/Show a Service IP Label/Address

  IP Label/Address                            lchlcip
* New IP Label/Address                        []
  Netmask(IPv4)/Prefix Length(IPv6)           [24]
* Network Name                                [net_ether_02]
  Resource Group Name                         RG_Master_HLC
```

Figure 73: PowerHA – configure the service IP label (Change screen)

NOTE: only use letters and numbers for the IP alias. Other names will not be accepted by SAP APO for integration later on. See chapter 5.4.7 regarding “Official hostname – IP alias considerations”.

9.10 Create Resource Group Processing Order

Change or create the following Resource Group Order. This will handle the aspect of releasing or acquiring the Resource Groups in the correct manor.

```
          Change/Show Resource Group Processing Order

 Resource Groups Acquired in Parallel
 Serial Acquisition Order        RG_Log_HLC RG_Master_HLC RG_Standby_HLC
 New Serial Acquisition Order    [RG_Log_HLC RG_Master_HLC RG_Standby_HLC

 Resource Groups Released in Parallel
 Serial Release Order            RG_Standby_HLC RG_Master_HLC RG_Log_HLC
 New Serial Release Order        [ RG_Standby_HLC RG_Master_HLC RG_Log_HLC ]
```

Figure 74: Resource Group Processing Order

9.11 PowerHA Application Servers

This section begins to link in the scripts that are used to keep the application available on base of its resource groups. The liveCache HotStandby solution uses the same scripts for both, master and slave and triggers the right action within the scripts by means of variables. On base of these parameters and a central configuration file the scripts adapt themselves while the user has to only change one text file.

NOTE: The master’s application monitor runs in the foreground compared to the other two.

The smitty submenu is:
Cluster Applications and Resources → Resources → Configure User Applications (Scripts and Monitors) → Application Controller Scripts → Add Application Controller Scripts

Naming convention for the application servers for the two database instances are:
as_master_<SID>, as_standby_<SID>
Valid Parameters for the scripts are: [MASTER/SLAVE] <SID>

Naming convention for the application servers for the log is:
as_log_<SID>
Valid Parameters for the script are: START/STOP] <SID>

9.11.1 Example of the Master Application Server

/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby3_startDB MASTER <SID>
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB MASTER <SID>

```
Change/Show Application Controller Scripts

 Application Controller Name       as_master_HLC
 New Name             [as_master_HLC]
 Start Script         [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB MASTER HLC]
 Stop Script          [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB MASTER HLC]
 Application Monitor Name(s)
 Application startup mode     [foreground]
```

Figure 75: PowerHA – configure master application server

9.11.2 Example of the Standby Application Server

/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB SLAVE <SID>
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB SLAVE <SID>

```
Change/Show Application Controller Scripts

 Application Controller Name       as_standby_HLC
 New Name               [as_standby_HLC]
 Start Script           [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB SLAVE HLC]
 Stop Script            [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB SLAVE HLC]
 Application Monitor Name(s)
 Application startup mode           [background]
```

Figure 76: PowerHA – configure standby application server

9.11.3 Example of the Log Application Server

/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_cleanDB START <SID>
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_cleanDB STOP <SID>

```
Change/Show Application Controller Scripts

 Application Controller Name       as_log_HLC
 New Name              [as_log_HLC]
 Start Script          [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_cleanDB START HLC]
 Stop Script           [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_cleanDB STOP HLC]
 Application Monitor Name(s)
 Application startup mode                   [background]
```

Figure 77: PowerHA – example of the log application server

9.12 PowerHA Application Monitors

After configuring the application servers specifying the start and stop mechanisms for the application a health monitor – a PowerHA Application Monitor – is added to the application monitors for each resource group.

If not already done: prepare a notification method to be added to the application monitors.

The smitty submenu is:
Cluster Applications and Resources → Resources → Application Monitors → Configure Custom Application Monitors

The application monitors are used to detect the status of the application itself and trigger a failover or restart if necessary. The monitor is started right before and while a resource group is activated. The application monitor of the master is designed to trigger an immediate failover in case of a problem with the service. It does not try to recover the application locally as for liveCache this would mean a cold start from a failed liveCache when we have a HotStandby waiting on the other node. The slave will never be failed over. It has some restart attempts. As an alternative a failure action of "*notify*" along with a "notification method" can be specified. Never the less, the design strategy is to always keep on trying to keep everything activated as transparent as possible.

As the application start scripts for the log and standby are started in background, and the monitor delay time begins as soon as the application startup scripts are called, it can happen that the monitor begins to monitor the application before it is completely finished with its activation. For the standby monitor a little sleep is implemented on base of the internal configuration variable TIMEOUT[lc_wait_long_in_startup] what is set to 10 seconds.
The application monitor of the master is set to foreground along with a startup monitor. The purpose of the startupmonitor is to ensure the start script is called for every start activity of the RG. The actual monitoring is done inside the start script. This is possible by the "foreground" setting. Meaning the runtime monitor will not be called before the start script has finished.

NOTE: The "*Monitor Interval*" configured is not only the frequency the monitors are called. It is also the timeout after which a not yet returned monitor will be considered hung. A cluster will react on this with an application failure. When making a decision also the TIMEOUT variable (set to 10 seconds as a fixed value in the product) needs to be taken into account. This parameter effects :

- A failed application will be reported immediately, only in a hung situation we lose some time.
- The failure detection will be delayed in these cases where the failure happened in the first ten seconds of a new "*Monitor Interval*".

Naming conventions for the instance runtime monitors
Monitor Name:
mon_[master|standby]_<SID>
Monitor Method:
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_monitorDB [MASTER|SLAVE] <SID>
Cleanup Method:
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB [MASTER|SLAVE] <SID>
Restart Method:
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB [MASTER|SLAVE] <SID>

Additional naming conventions for the startup monitor of a master instance
Monitor:
mon_master_startup_<SID>
Monitor Method:
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startupmonitor MASTER <SID>

The monitor for the log device is used to early detect a failure of the device and proactive detect issues.
Monitor Name:
mon_log_<SID>

Monitor Method:
/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_logmonitor

9.12.1 Commandline for creating an application monitor

In case the smit panels do not allow parameters to be passed along with the scripts (example startup monitor of Master instance) it may be necessary to bypass this by using command line.

```
/usr/es/sbin/cluster/utilities/claddappmon name=mon_master_startup_HLC \
    MONITOR_TYPE='user' \
    RESOURCE_TO_MONITOR=as_master_HLC \
    INVOCATION='startup' \
    FAILURE_ACTION='notify' \
    RESTART_METHOD="/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB MASTER HLC" \
    CLEANUP_METHOD="/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB MASTER HLC" \
    MONITOR_METHOD="/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startupmonitor MASTER HLC"
\   MONITOR_INTERVAL='10' \
    HUNG_MONITOR_SIGNAL='9' \
    STABILIZATION_INTERVAL='60' \
    RESTART_COUNT=1 \
    RESTART_INTERVAL=77
```

9.12.2 Example of the Master Start-up Monitor

```
Change/Show Custom Application Monitor

                                     [Entry Fields]
* Monitor Name                                 mon_master_startup_HLC
  Application Controller(s) to Monitor         as_master_HLC
* Monitor Mode                                 [Startup Monitoring]
* Monitor Method         [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startupmonitor MASTER HLC]
  Monitor Interval                             [10]
  Hung Monitor Signal                          [9]
* Stabilization Interval                       [60]
  Restart Count                                [1]
  Restart Interval                             [77]
* Action on Application Failure                [notify]
  Notify Method                                []<put one in place! Details are in PowerHA docu>
  Cleanup Method         [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB MASTER HLC]
  Restart Method         [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB MASTER HLC]
```

Figure 78: PowerHA – configuring the start-up monitor of the master

The intention of the start-up monitor and the design points are described in section 3.8. The only purpose is to not use the runtime monitor to determine if the startscript should be called or not. With this additional monitor it is ensured that the startscript is invoced always. The startscript itself can handle all states of a master instance including the startup surveillance.

9.12.3 Example of the Master Runtime Monitor

Change/Show Custom Application Monitor

	[Entry Fields]
* Monitor Name	mon_master_HLC
Application Controller(s) to Monitor	as_master_HLC
* Monitor Mode	[Long-running monitoring]
* Monitor Method	[/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_monitorDB MASTER HLC]
Monitor Interval	[60] <do not reduce! Consider to increase in case you encounter false failovers>
Hung Monitor Signal	[9]
* Stabilization Interval	[180]
Restart Count	[0]
Restart Interval	[0]
* Action on Application Failure	[fallover]
Notify Method	[]
Cleanup Method	[/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB MASTER HLC]
Restart Method	[/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB MASTER HLC]

Figure 79: PowerHA – configuring the runtime monitor of the master

The run-time monitor takes over from the time onwards the startscript completes. This is due to the “foreground” parameter set in the ResourceGroup definition. This monitor will trigger a service takeover by the standby in cases of failure as soon any state except from ONLINE is detected. The monitor interval is choosen on base of a pilot customer implementation in production.

NOTE: Ensure the “*Monitor Interval*” is also under load sufficient to complete the monitor cycle. Also the timeout set in the central configuration file will play a role in defining the interval in case the Process monitor method is chosen. Otherwise the cluster will get a signal of “*hung monitor*” and react to it as application failure. The value should be increased if not sufficient.

9.12.4 Example of the Standby Runtime Monitor

Change/Show Custom Application Monitor

	[Entry Fields]
* Monitor Name	mon_standby_HLC
Application Controller(s) to Monitor	as_standby_HLC
* Monitor Mode	[Long-running monitoring]
* Monitor Method	[/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_monitorDB SLAVE HLC]
Monitor Interval	[45]
Hung Monitor Signal	[0]
* Stabilization Interval	[120]
Restart Count	[30]
Restart Interval	[5445]
* Action on Application Failure	[fallover]
Notify Method	[]
Cleanup Method	[/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_stopDB SLAVE HLC]
Restart Method	[/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_startDB SLAVE HLC]

Figure 80: PowerHA – configuring the runtime monitor on the standby

The following parameters of the runtime monitor of a standby needs to be well chosen to first allow to bring up the master instance and then successfully start the standby instance:
Monitor Interval should be set to 45 as a start point. This value is based on our first experience with our pilot customer.
The time a standby should retry to establish should cover the following:

- Startup of the master (4 sec + 3*startupmonitor + 2*start script of master)*2 +X
- Long ADMIN state of a master (we have seen up to 30 minutes, can take longer depending on the log-volume size .
- TIMEOUT[lc_wait_long_in_startup] = 10 seconds + 30

NOTE: specify a notification method. In case the retry attempts have not been well chosen the RG can go into "OFFLINE DUE TO LACK OF NODE" and will not establish a standby automatically from then onwards.

9.12.5 Example of the log Runtime Monitor

```
                         Change/Show Custom Application Monitor

* Monitor Name                              mon_log_HLC
  Application Controller(s) to Monitor      as_log_HLC
* Monitor Mode                              [Long-running monitoring]
* Monitor Method              [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_logmonitor]
  Monitor Interval                          [60]
  Hung Monitor Signal                       [30]
* Stabilization Interval                    [180]
  Restart Count                             [0]
  Restart Interval                          [0]
* Action on Application Failure             [fallover]
  Notify Method                             []
  Cleanup Method  [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_cleanDB STOP HLC]
  Restart Method  [/usr/es/sbin/cluster/sa/maxdb/sbin/cl_hotstandby_cleanDB START HLC]
```

Figure 813: PowerHA – configuring the runtime monitor for the log

An outage of the log volume is detected only on write. To have an early detection mechanism this script is put into place for this type of application.

9.12.6 Adding the Application Servers and IP Labels into the resource groups

In order to put the service IP alias and the scripts into a PowerHA resource group the following smit submenu is used:

Cluster Applications and Resources → *Resource Groups* → Change/Show All Resources and Attributes for a Resource Group

9.12.7 Example of the Log Volume Resource Group

```
                  Change/Show All Resources and Attributes for a Resource Group

  Resource Group Name                            RG_Log_HLC
  Participating Nodes (Default Node Priority)    is04d6 is03d6
  Startup Policy                                 Online On All Available Nodes
  Fallover Policy                                Bring Offline (On Error Node Only)
  Fallback Policy                                Never Fallback
  Concurrent Volume Groups                       [lclogvg01 lclogvg02]
  Use forced varyon of volume groups, if necessary  false
  Automatically Import Volume Groups             false
  Application Controllers                        [as_log_HLC]  <F4 and select the app.serv.>
  […]
```

Figure 82: PowerHA – example of the concurrent log resource group RG_Log_<SID> – the prerequisite resource group

NOTE: This resource group is created during the setup of the shared log volume in section 9.5.1, "Configuration and preparation of the log". Please refer to this section for more detailed information.

9.12.8 Example of the Master Resource Group

Once a failover has occurred and there is an active online database, there should be no automatic fallback driven by the recovery of the failed node. This would interrupt the running application. A fallback should be done when the application is least affected and therefore this decision is left to the operations management of the application (if and when a failover to the original server is desired).

```
                  Change/Show All Resources and Attributes for a Resource Group
                                                    [Entry Fields]
 Resource Group Name                                RG_Master_HLC
 Participating Nodes (Default Node Priority)        is03d6 is04d6
 Startup Policy                                     Online On First Available Node
 Fallover Policy                                    Fallover To Next Priority Node In The List
 Fallback Policy                                    Never Fallback
 Service IP Labels/Addresses                        [lchlcip]
 Application Controllers                            [as_master_HLC]
[… change nothing below … ]
```

Figure 83: PowerHA – example of the master resource group RG_Master_<SID>

9.12.9 Example of the Standby Resource Group

```
                  Change/Show All Resources and Attributes for a Resource Group
                                                    [Entry Fields]
 Resource Group Name                                RG_Standby_HLC
 Participating Nodes (Default Node Priority)        is04d6 is03d6

 Startup Policy                                     Online On First Available Node
 Fallover Policy                                    Fallover To Next Priority Node In The List
 Fallback Policy                                    Never Fallback
 Service IP Labels/Addresses                        []
 Application Controllers                            [as_standby_HLC]
[… change nothing below …]
```

Figure 84: PowerHA – example of the standby resource group RG_Standby_<SID>

9.13 PowerHA HotStandby Scripts

The scripts are shipped along with the Smart Assist agent and are located in /usr/es/sbin/cluster/sa/maxdb/sbin. Detailed information can be found in the PowerHA documentation. For the manual path verify you have performed the tasks of section 8.3.

10 liveCache Tuning and Customizing

This chapter captures some tuning recommendations and customization options. The tuning recommendations are applicable to most environments. The values used in the examples are choosen on base of the given hardware in the PoC and will need to be reviewed in light of the target environment.

10.1 Tuning liveCache

MaxCPUs – for a liveCache in its own LPAR, the number of the MaxCPUs can be set to the number of logical CPUs (SMT threads) in this LPAR. This will allow full utilization of the CPU resources. Any less than the number of SMT threads can limit the parallelization of the workload. Defining more than the available threads makes no sense, as each liveCache CPU is capable of multi-tasking and adding more liveCache CPUs than there are SMT threads would result in superfluous levels of multi-tasking. The number of SMT threads, referred to as logical processors, can be determined via the following command options:

smtctl - this command shows the linkage of the threads to the physical or virtual processors
vmstat – this shows the lcpus (logical cpu) number in the system configuration.

For the workloads used to achieve the results in this document, the following parameters were configured in the liveCache:

- **LogQueueSize** – varied from 400 to 2000. When using 64K pages, the higher settings for liveCache improved the throughput.

- **LogQueues** – this parameter was set equal to the MaxCPU setting. MaxCPU and LogQueues were both set to 16. For PP/DS, the parallel workload used for the majority of tests was always <=16 so the MaxCPU was not set beyond this number.

- **MaxLogWriterTasks** - this was set to 16 as well, one per LogQueue.

The following parameters also need review in relation to the size of the APO system.

- **Max User Tasks**
  Max user tasks depend on the number of SAP work processes which will connect to the liveCache. Each work process has the possibility of opening multiple concurrent connections. To be on the safe side, this parameter is often set to 3 times the number of work processes.

- **Max SQL Locks**
  This parameter should be set quite high. It depends on the size of uncommitted data which is holding SQL level row locks. 300000 – 600000 is recommended.

SAP HotStandby supports prefetching of the memory pages by the Standby server. To activate this recommended behavior, set the following parameter. The prefetching ensures that there is no I/O bottleneck at failover which will restrict the performance while the memory structure is filled.

- **Preload**
  PreloadObjectsInStandbyMode should be set to yes.

10.2 Using 64K Pages

For Large liveCache systems in HotStandby, the liveCache can be set to manage the data cache in 64K pages rather than 4K pages. This reduces the system overhead. This is implemented as a liveCache parameter setting in the following manner:

The switch to 64K pages is set using the liveCache parameter directly from dbmcli

dbm_configset KnlSetenv01
LDR_CNTRL=DATAPSIZE=64K@TEXTPSIZE=64K@STACKPSIZE=64K@SHMPSIZE=64K

To display the parameter use:
dbm_configget

To reset the liveCache to default 4K pages, set the parameter to null:
dbm_configset KnlSetenv01

Care should be taken in implementing large pages if AME is planned. AME with 64K pages is covered in the document "SAP SCM on Power7" in section 13.1.

NOTE: a restart of liveCache is required to activate new parameters

10.3 Best Performance for LUNs

The 3-tier configuration used a Hot Standby liveCache. This liveCache was configured with raw devices and two physical volumes over storage virtualization. To maximize the throughput on the volumes the following settings were used in AIX 6.1.

- The NPIV (virtual SAN adapters) were modified to increase the num_cmd_elems :
  chdev -l fcsn -a num_cmd_elems=2048 –P
- The hdisks which housed the liveCache data were changed to increase the queue_depth:
  chdev -l hdisk<n> -a queue_depth=256 –P

10.4 OS Considerations

HotStandby and Forking

The current implementation by SAP drives the communications between the cluster nodes via dbmcli. This causes a fork to be initiated from the primary liveCache thread and this can cause unnecessary overhead. To reduce the fork overhead, the following restricted parameter should be set:

Change Forking value - AIX: vmo -p -o vmm_fork_policy=0
The liveCache should be performance optimized by changing the following parameter to use multiple CPUs:

dbmcli > param_directput ALLOW_MULTIPLE_SERVERTASK_UKTS YES
OK
dbmcli > param_checkall
OK

10.5 Autologging

It is recommended to have autologging active for any production liveCache. This ensures that the online log is archived so space is available for continuous logging.

NOTE: Do not forget to enable auto logging in liveCache (easiest through the dbmgui):

• Create full data backup using the backup wizard (ADMIN state)
• Enable Automatic auto logging using the backup wizard (ONLINE state)

10.6 DB Analyzer

The DB Analyzer can be automatically started when liveCache is started. This is the normal behavior when activating the liveCache from the APO transaction LC10. The anaylzer can be stopped and started from APO as desired. In the case of a HotStandby takeover due to failure of the primary, the DB analyzer can be set to automatically restarted on the new primary by activating the automatic function.

You activate or deactivate the automatic function for starting DB Analyzer whenever the database is started. When you activate the function while DB Analyzer is not running, DB Analyzer is started immediately. When you deactivate the function, Database Anaylzer is stopped.

You also use this command to determine whether the function is ON or OFF. The option can be executed from DBMCLI.

Prerequisites

You have the DBInfoRead server permission.

Structure

```
auto_dbanalyzer <option>

<option> :: = ON | OFF | SHOW
```

For futher information please see SAP page on auto dbanalyzer
http://help.sap.com/saphelp_maxdb77/helpdata/en/f5/ea88a0229f4bf796ff7099d122d7c2/content.htm

11 Integration liveCache into APO

In the case of moving from a single liveCache instance to a HotStandby, the likelihood is that the customer will chose to build a new liveCache and bring the new liveCache into production, rather than try to reconfigure the current liveCache. Very likely the current liveCache is file system based and the HotStandby requires raw devices and a very special disk layout. Building a new HotStandby liveCache and following a staged migration will allow all failover testing to be done on the new liveCache before actually activating it into production.

This method will also provide a failback in case of problems in the transition.
It will reduce the cutover time. The downtime required is the time it takes to create a backup of the current liveCache and restore it on the new HotStandby liveCache.
At this time APO should not be actively using the liveCache or modifying data. Ideally, APO should be taken out of production for this cutover.

The method below can be tested with an online backup of the production liveCache. This will help reduce the actual cutover path as the steps which must be following will be known and tested in advance. In the procedure below, there are some options which depend on the state of the source and the requirements of the target systems.

When administrate or maintain a clustered liveCache, it should be first shutdown from APO level to ensure that no failover activity takes place. By stopping the liveCache from APO, the application lock is set which deactivates the failover. How to prevent cluster activity during a maintenance window in a HotStandby liveCache is described in detail in the administration guide.

The detailed procedure to transfer the data into the clustered liveCache is described in the Admin guide: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

12 Maintenance Procedures and Best Practices

liveCache maintenance activities on a cluster have to take into consideration the behaviour of the cluster and how it will react to a change of liveCache status and other changes in the liveCache layout. The following methods are therefore provided to help in managing a liveCache cluster.

The detailed procedure to transfer the data into the clustered liveCache is described in the Admin guide: http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

13 Appendix

13.1 Related documents and sources of further information

Master Document – "Invincible Supply Chain" reference architecture for top-to-bottom HA
http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

Administration Guide for HotStandby liveCache
http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

An IBM RedBook is available online with details on the functionality of PowerHA 7.1.
http://www.redbooks.ibm.com/abstracts/sg247845.html?Open

SAP liveCache Hot Standby command set can be found at the website below:
http://maxdb.sap.com/doc/7_7/45/0f77bbe82f29efe10000000a114a6b/frameset.htm

Further information on the SAP liveCache command line and DBMGUI tool.
Command Line (dbmcli):
http://help.sap.com/saphelp_nwpi711/helpdata/en/a3/b2462a9ef05c41922b8092257a2e2c/frameset.htm

DBMGUI:
http://help.sap.com/saphelp_nwpi711/helpdata/en/a3/b2462a9ef05c41922b8092257a2e2c/frameset.htm
(Use version 7.6 and higher)

Smart Assist for liveCache HotStandby
http://publib.boulder.ibm.com/infocenter/aix/v7r1/index.jsp?topic=%2Fcom.ibm.aix.powerha.smartassist%2Fsmart_live_main.htm

SAP SCM on Power7 – performance and implementation recommendations
http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102031

SAP Core Systems with DB2 HA/DR on Power7
http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP102036

13.2 SAP liveCache versions tested

The proof of concept was executed using two different version of SAP liveCache:

7.7.07.17
7.7.07.26
7.7.07.39

Upgrade to SP26 (7.7.07.26) was done due to a known problem in failover recovery in versions prior to support pack 16 that which did occur in the testing. Upgrade to build 39 was done to overcome a known error leading to a deadlock situation in the HotStandby.

Application Levels used for high level testing:

Support Package Level for Installed Software Components

Component	Software C	SP-Level	Support Package	Short description of the component
SAP_ABA	701	0006	SAPKA70106	Cross-Application Component
SAP_BASIS	701	0006	SAPKB70106	SAP Basis Component
PI_BASIS	701	0006	SAPK-70106INPIBASIS	Basis Plug-In
ST-PI	2008_1_700	0000	-	SAP Solution Tools Plug-In
SAP_BS_FND	701	0006	SAPK-70106INSAPBSFND	SAP Business Suite Foundation
SAP_BW	701	0006	SAPKW70106	SAP Business Warehouse
SAP_AP	700	0019	SAPKNA7019	SAP Application Platform 7.00
EA-IPPE	400	0017	SAPKGPID17	SAP iPPE (EA-IPPE) 400
SCEMSRV	700	0006	SAPK-70006INSCEMSRV	SCEMSRV 7.0
SCM	700	0006	SAPKY70006	Supply Chain Management 7.0
SCMBPLUS	700	0006	SAPK-70006INSCMBPLUS	SCM Basis PLUS 7.0
SCMEWM	700	0006	SAPK-70006INSCMEWM	Extended Warehouse Management 7.0
SCMSNC	700	0006	SAPK-70006INSCMSNC	SCMSNC 7.0
SCM_BASIS	700	0006	SAPK-70006INSCMBASIS	SCM Basis 7.0
BI_CONT	704	0006	SAPK-70406INBICONT	Business Intelligence Content
QIE	200	0006	SAPK-20006INQIE	Quality Inspection Engine 2.00

14 Trademarks and disclaimers

14.1 Copyrights and trademarks

 References in this document to IBM products or services do not imply that IBM intends to make them available in every country.

The following terms are trademarks of International Business Machines Corporation in the United States and/or other countries: Advanced Micro-Partitioning, AIX/L(logo), AIX 5L, DB2 Universal Database, eServer, i5/OS, IBM Virtualization Engine, Micro-Partitioning, iSeries, POWER, POWER4, POWER4+, POWER5, POWER5+, POWER6. POWER7

A full list of U.S. trademarks owned by IBM may be found at: http://www.ibm.com/legal/copytrade.shtml

UNIX is a registered trademark of The Open Group in the United States and other countries. Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

SAP, the SAP Logo, SAP, R/3 is trademarks or registered trademarks of SAP AG in Germany and many other countries.

More information about SAP trademarks can be found at: http://www.sap.com/company/legal/copyright/trademark.asp

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Other company, product or service names may be trademarks or service marks of others.

Information is provided „AS IS“ without warranty of any kind.

Information concerning non-IBM products was obtained from a supplier of these products, published announcement material, or other publicly available sources and does not constitute an endorsement of such products by IBM. Sources for non-IBM list prices and performance numbers are taken from publicly available information, including vendor announcements and vendor worldwide homepages. IBM has not tested these products and cannot confirm the accuracy of performance, capability, or any other claims related to non-IBM products. Questions on the capability of non-IBM products should be addressed to the supplier of those products.

14.2 Disclaimer and special notices

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.
Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems.
Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.
Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.
This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products.
All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.