Supplementary Document to:

Invincible Supply Chain

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

Administration Guide for SAP HotStandby liveCache with PowerHA 7.1.1

IBM

31/ July 2012

This Document

This administration 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 and other related documents can be found at the following website:

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

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

1 Table of Contents

2 Preface

2.1 Document Scope

This document is intended as an administration guide for the IBM implementation of HotStandby liveCache supported by PowerHA SystemMirror.
This document is supplementary to the design and implementation documents available at the same website. The latest versions of all these documents can be found here:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100677

2.2 Special Notices

2.3 Version Information

This is version 3.0 of this document.

2.4 Authors of this Document

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

3 Overview of the Solution Design

This section is an overview of the design for reference. The HotStandby functionality is based on an API provided by SAP for MaxDB and liveCache. The API is used by infrastructure providers to integrate the necessary infrastructure support required to meet the HotStandby functionality exported from liveCache. This chapter introduces the SAP functionality and the IBM implementation.

3.1 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.

3.1.1 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.

3.1.2 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

[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.

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.

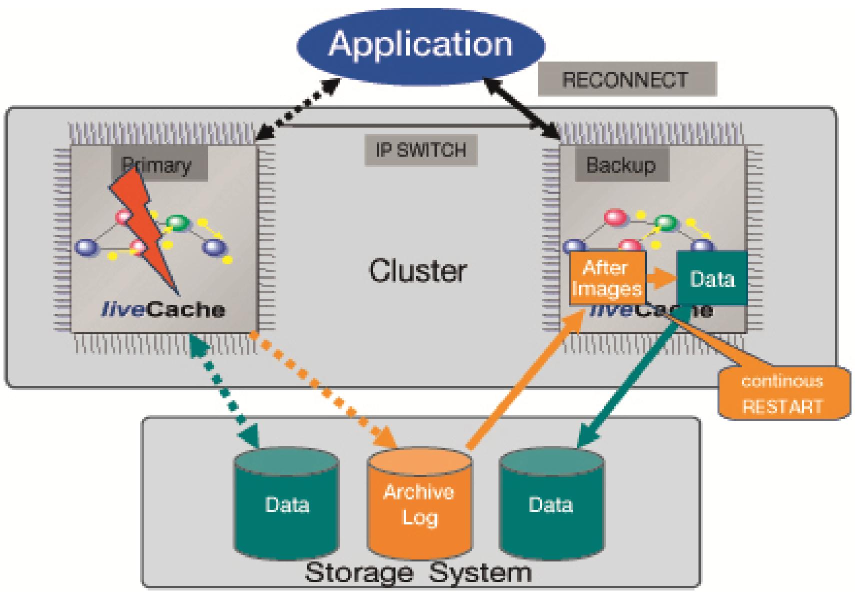


Figure 1: Overview of SAP liveCache HotStandby crash recovery provided by SAP

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 can move 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 single storage system. The control logic of the storage integration is implemented by a storage library (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 3.2.5.

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 base 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. Therfore 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. These parameters are visible in the extended parameters for a HotStandby liveCache in DBMGUI.

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. This is described in the implementation guide.

See the section 6.1 “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

3.1.3 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.

3.2 IBM HotStandby Implementation Overview

The implementation of HotStandby supported by IBM System Storage is based on IBM FlashCopy functionality. A complete, or clone FlashCopy with consistency grouping 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.

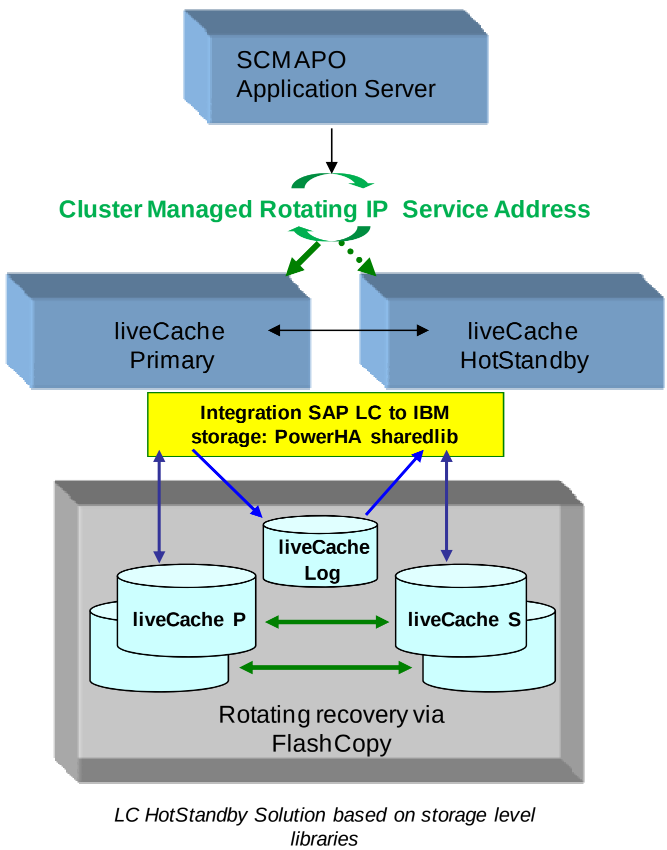


Figure 3: Overview of HotStandby with FlashCopy (green) and read/write of shared log (blue)

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.

3.2.1 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.

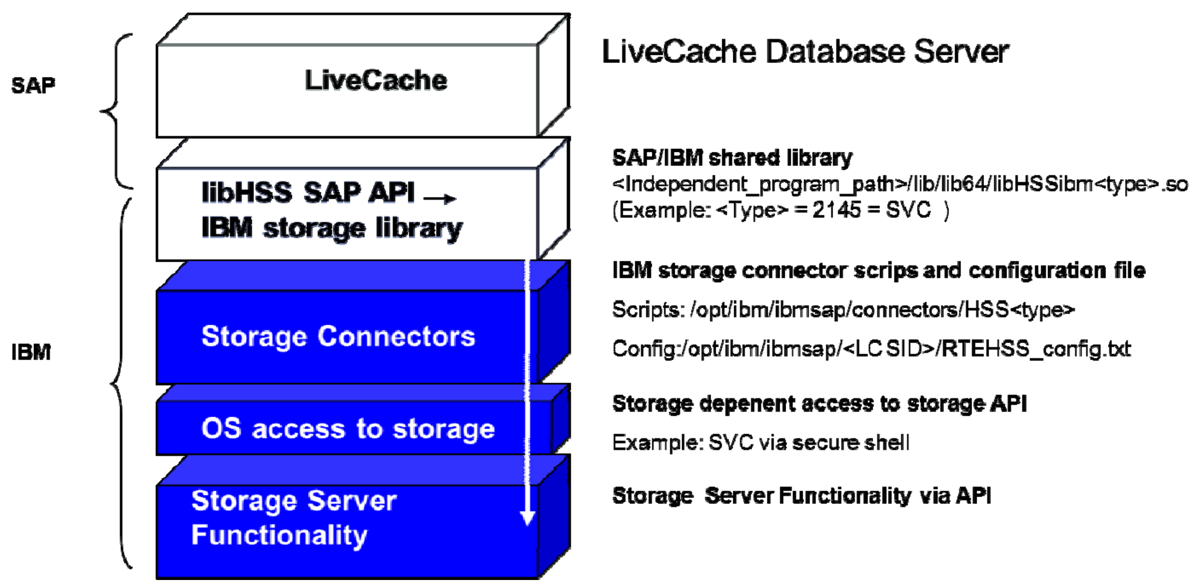


Figure 4: Overview of solution components

The shared library must be installed 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.

3.2.2 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 6.1 on related documents.

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.

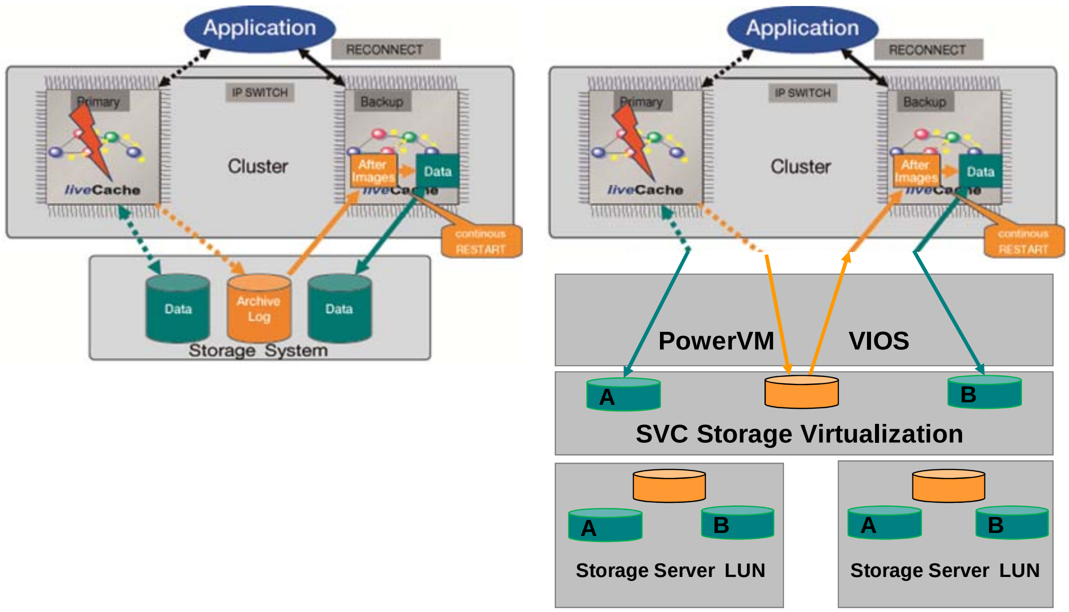


Figure 5: Implementation of virtualization layers

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 (dbmcli 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 communication.

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.

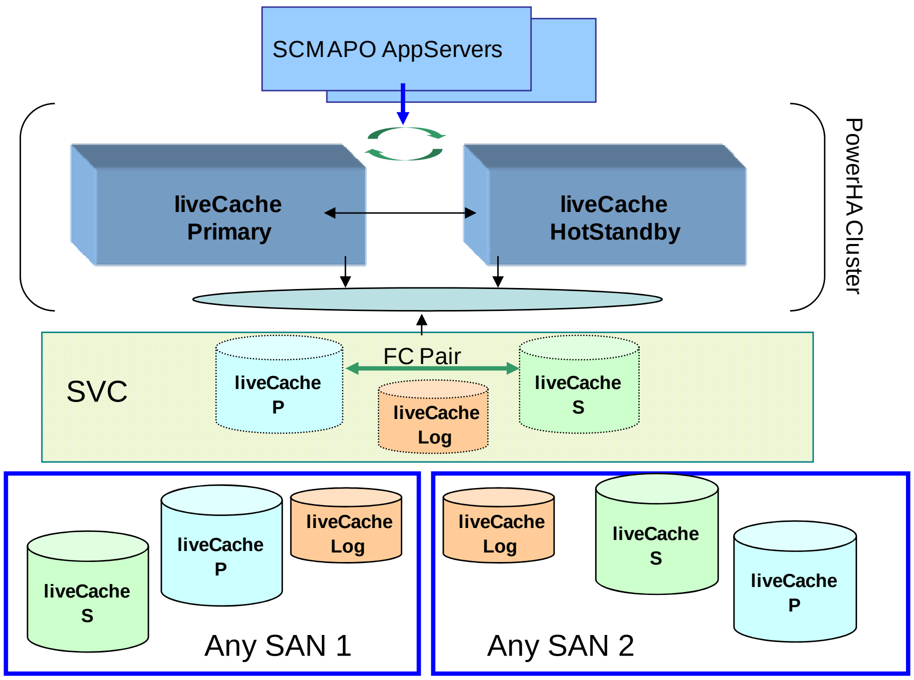


Figure 6: SAP liveCache with HotStandby with SVC mirrored storage

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. A SVC stretched cluster can be used to extend the configuration to span two sites.

3.2.3 Design of the PowerHA Cluster for SAP liveCache HotStandby

This section describes the general design with focus on the difference between the cluster infrastructure and the liveCache service, including the start, the stop and the monitor logic. Figure 7 shows an example design for PowerHA HotStandby cluster in an SAP landscape.

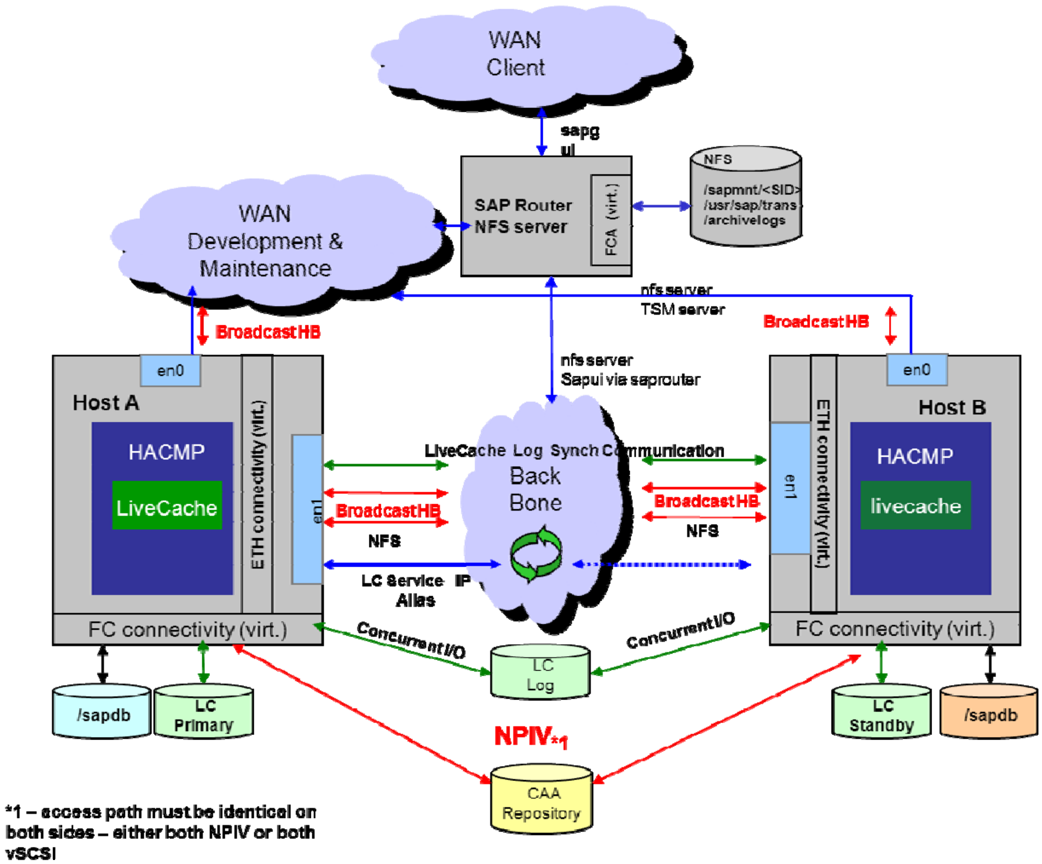


Figure 7: Overview of cluster concept used in reference architecture for proof of concept

3.2.4 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 volume groups, service IP (OFFICIAL_NODE), PowerHA application servers 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 the HotStandby but typical in an HA implementation.

3.2.5 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.

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 slave 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 slave 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 master or standby is in ADMIN mode the monitor accepts this state as well. For example a Master can be for a long time in ADMIN mode after a failover to redo outstanding logs. Not accepting this state the cluster would end in subsequent failover attempts.

In general, during a takeover, both resource groups switch nodes. In case of a one node situation the slave 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>).

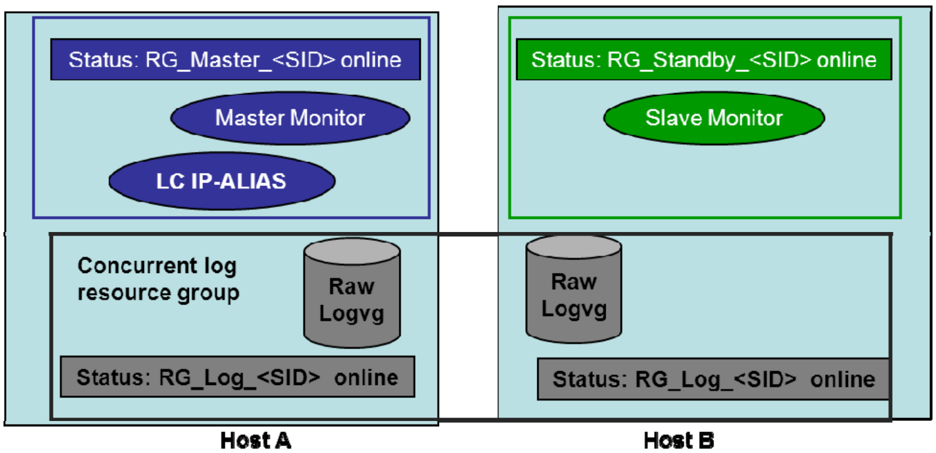


Figure 8: Overview of the PowerHA resource groups

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 that 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*	*is03d11*
	ONLINE	*is04d11*
RG_Master_<SID>	*ONLINE*	*is03d11*
	OFFLINE	*is04d11*
RG_Slave_<SID>	*ONLINE*	*is04d11*
	OFFLINE	*is03d11*

The 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. This information is maintained in a shared "lock directory" which must be setup at implementation and enabled for use by sdb to maintain this information – see Figure 9.

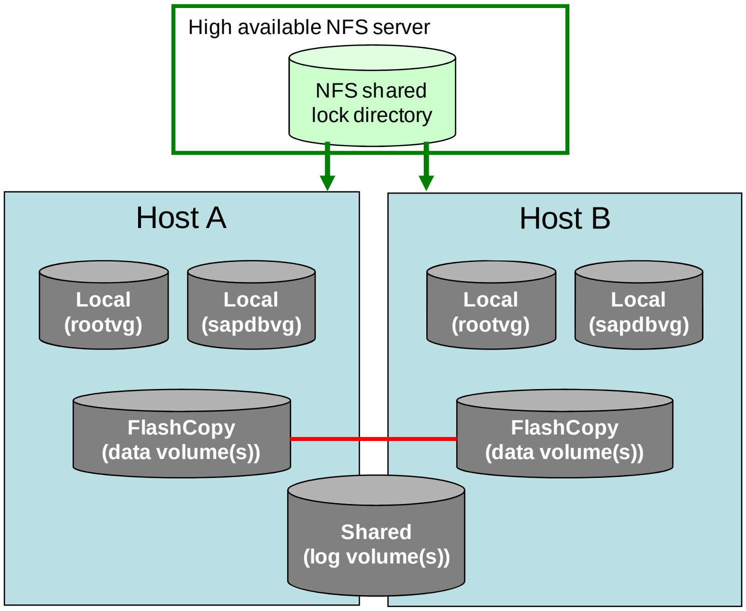


Figure 9: Overview of the storage components with NFS

3.2.5.1 SAP liveCache service

The APO application is responsible for the status of the SAP liveCache database service.
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 restarting 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.

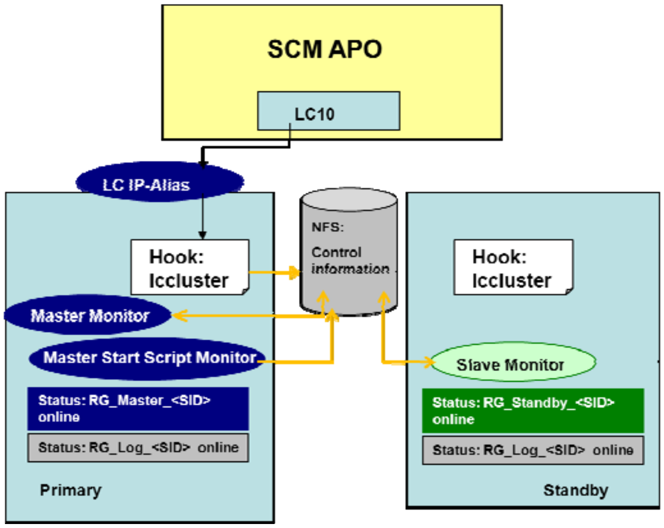


Figure 10: SAP APO control of SAP liveCache service

3.2.6 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.

3.2.6.1 Starting a clustered SAP liveCache from SAP APO

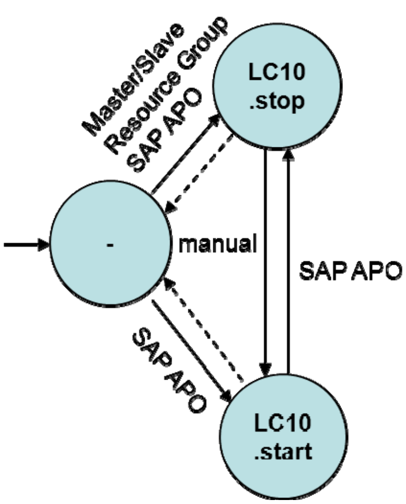


Figure 11: Logic of the SAP APO PowerHA integration

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 was stopped by APO, 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.

From the application monitors' or start scripts' perspective the lock directory can be empty; this would be treated as equivalent to LC10.stop and resulting in the initialization to LC10.stop. When the application is started by APO, this status changes to LC10.start and the cluster maintains an active liveCache.

3.2.6.2 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.

3.2.7 Monitoring of the Application

The application monitors maintain the liveCache service according to the requirements of APO. The focus is always on the SAP system requirements. Therefore a change in liveCache state which does not fit the expectation for continuous service to APO will result in a failover with the exception of the ADMIN state.

Long Admin on the Standby

The admin state can appear due to several reasons for an active or deactivated instance, but never for a Standby instance directly. The SAP liveCache application logic prevents bringing a liveCache instance from STANDBY to ADMIN mode.
The ADMIN states are assessed and managed by the application monitors. ADMIN can be an interim or long state during a start-up or transition. In the case of a standby resynchronization, the admin state can remain for an extended time. This is the time when the state of the data consistency is assessed and can may result in either an active period in which log entries are reapplied for resynchronization or a new FlashCopy is initiated to synchronize the data. This activity must not be interrupted. As this is the standby, there is no hectic to get the service back, the focus is on consistency.

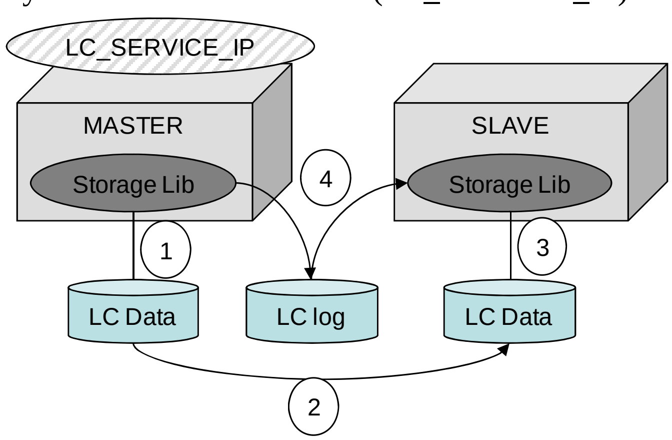
Interim ADMIN state or long ADMIN state on the Master
The interim ADMIN state is part of a normal start up, but here this is usually only a short transient state. There are situations however in which a "long ADMIN" may occur. For example, when a master fails over to a standby, there may still be outstanding logging activity which needs to be completed to finalize the synchronization before the instance can move into online mode for a master. Therefore an extended ADMIN mode is tolerated in the master prior to it having achieved the ONLINE status.

ADMIN STATE FOR MAINTENANCE
The liveCache master instance must be brought into admin state for administrative purposes. For a HotStandby this should be done in a controlled manner to ensure the synchronization and behaviour of the cluster. As maintenance activity on the liveCache will interrupt access from APO, the expectation is that such activities will be coordinated with the APO administrator, and that the liveCache set offline from APO. Setting the liveCache offline from APO will disarm the monitors and allow maintenance as usual. A restart of the liveCache from APO will reactivate all functionality of the HotStandby.

3.2.8 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.

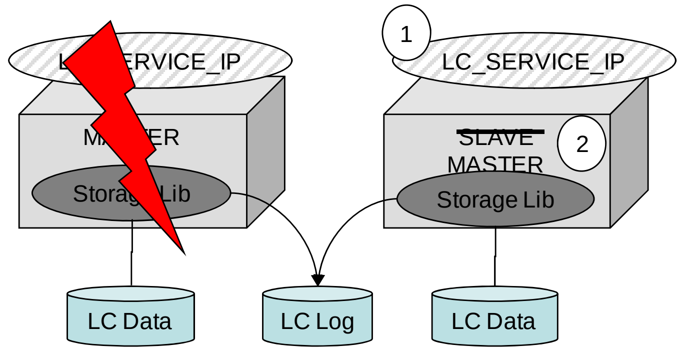
2. The implemented logic of PowerHA triggers via the master instance the startup of the standby instance. This may activate a FlashCopy which then initializes the persistent data of the standby if necessary. The data refresh is complete, from the view of the HotStandby, as soon the logical copy process of the FlashCopy 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 and replicates the changes in it own memory structures and persistent copy.

Phase 2: Failover of primary

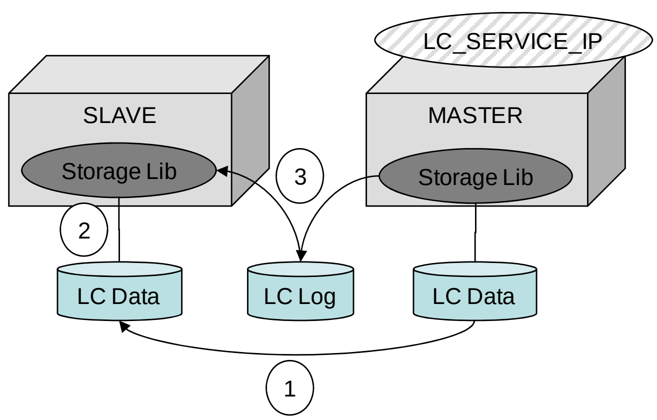
If the master instance is not operational, a failover is triggered by the cluster logic.



1. The communication daemon of the liveCache (x_server) is stopped on the Standby to avoid connection attempts during role swop. 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. Business can recommence.

Phase 3: Reactivation of a standby

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



1. The implemented PowerHA logic triggers the startup of the standby. This again starts a FlashCopy refresh if necessary. In case the physical copy from Phase2 is not yet 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.

3.2.9 Storage Prerequisites and best practices for storage planning and design

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

3.2.9.1 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.

The number of volume groups in this limitation refers to the first installation. This number is supported in the Smart Assist and Wizard. In this document there is also a procedure for increasing the size of the storage footprint for liveCache as part of the lifecycle. This procedure is not restricted by the installation tools and can include additional volume groups.

3.2.9.2 liveCache executables and work directories

liveCache has additional storage requirements in regards to the local storage for liveCache executables, 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 exists separately on each and is relevant only 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 filesystem 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*.

3.2.9.3 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 (executables, runlogs, error and kernel logs) and in this instance, space for archive logs which may be backed up periodically to a storage media. 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.

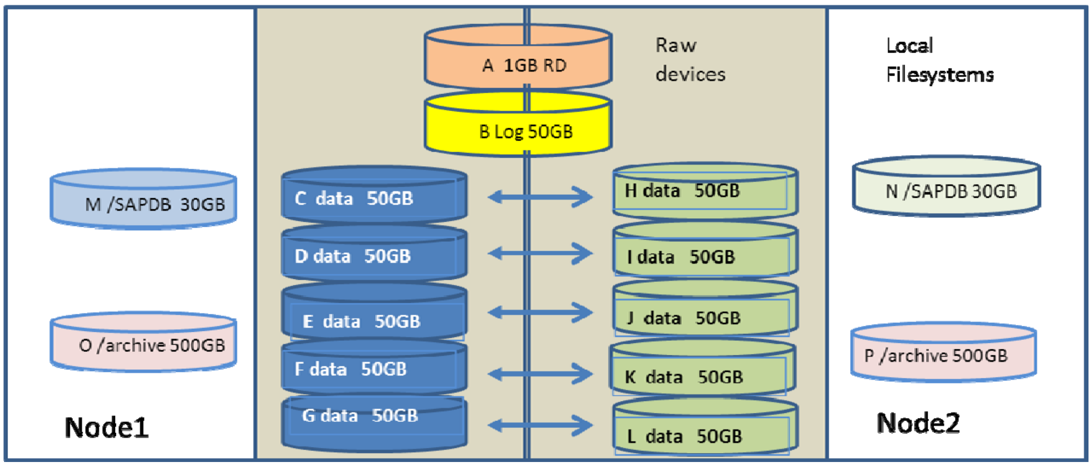


Figure 12: Example of a data layout for liveCache

Explanation and use of volumes in diagram:

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 filesystems 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 executables and local data for each liveCache instance. /sapdb must be available for the installation of the liveCache software.

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.

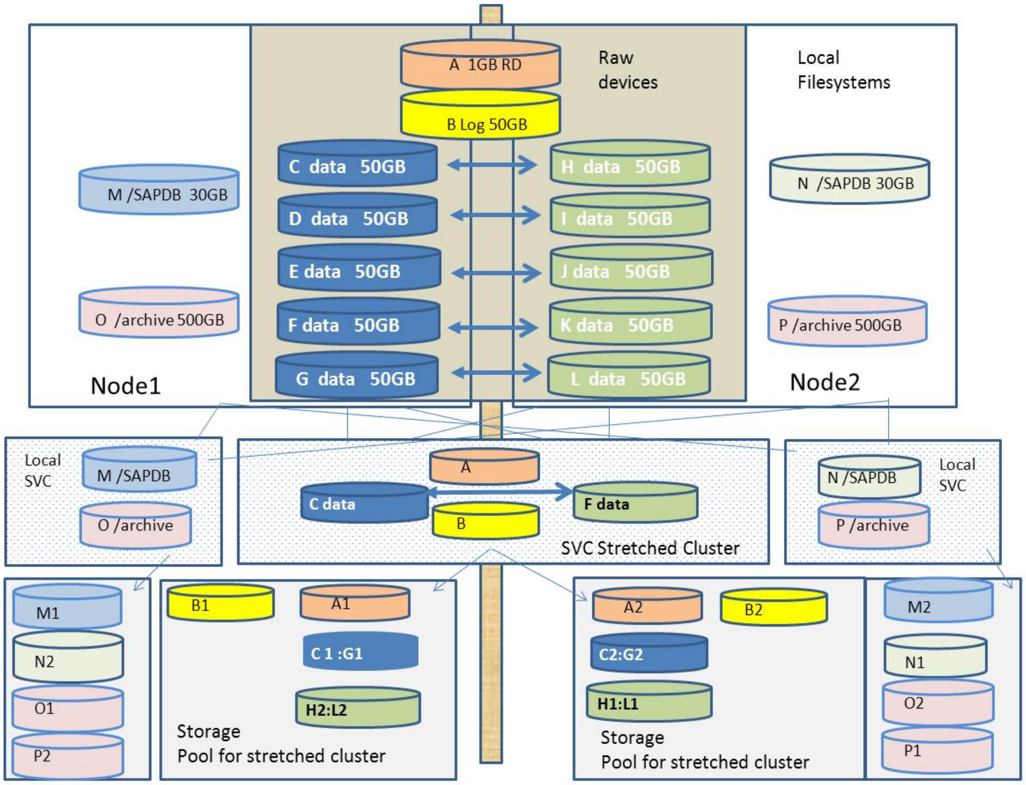


Figure 13: Example of storage design and volume group layout

The storage layout in Figure 13 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.

3.2.9.4 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.

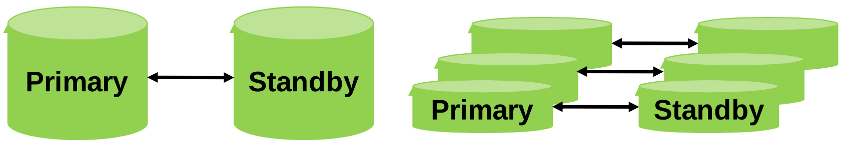


Figure 14: FlashCopy disk design – few vs. many small

3.2.9.5 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.

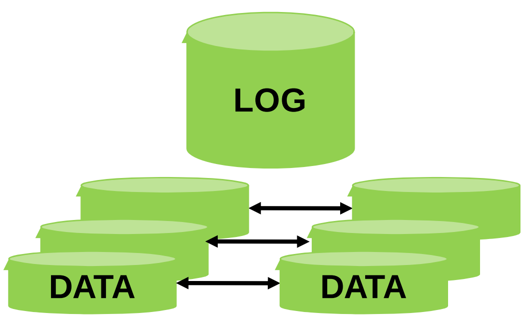


Figure 15: Log size

4 Administration of a HotStandby liveCache

This section focuses on the day to day management of a HotStandby liveCache as implemented by IBM. Maintenance activities in many cases must take into consideration that this is a clustered solution and that some additional steps may be required to ensure consistency is maintained. The solution can easily resynchronize itself but there are some requirements which enable this action.

In this document several roles have been anticipated. As there are sometimes several ways to achieve the same outcome, the approach is documented with a role. We would not necessarily expect the OS administrator to have access to APO or application tools. In the same sense we would necessarily expect the APO administrator to operate AIX clustering commands at OS level. In some cases there must be collaboration between the two roles as actions are required from both. For example, if the liveCache is taken offline, we would expect this to be done from APO administrator level as this has significant impact on the application as a whole.

ROLES:
APO – application administrator for APO
OS – AIX administrator with cluster know how and authorization to manage the cluster

4.1 Overview of Differences

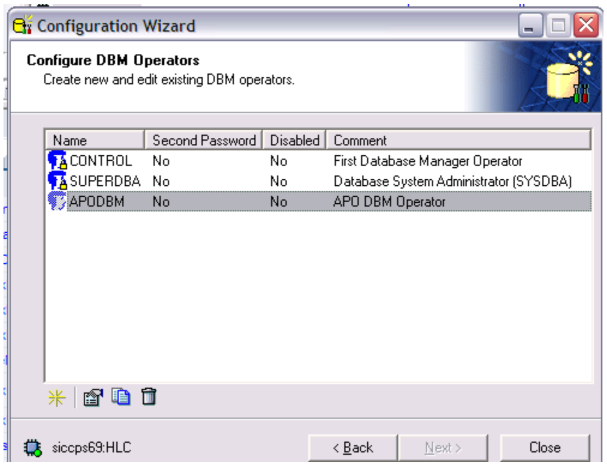
This section discusses the different view of the liveCache depending on the role and the tools each will use. In general, the liveCache HotStandby can be monitored and managed exactly like a normal liveCache. The major differences come with administration of changes which require a awareness of the two instances in the cluster. An easy administration and resynchronization method has been made available which ensures synchronization for tasks such as changing liveCache parameters, increasing the datacache size, and upgrading liveCache.

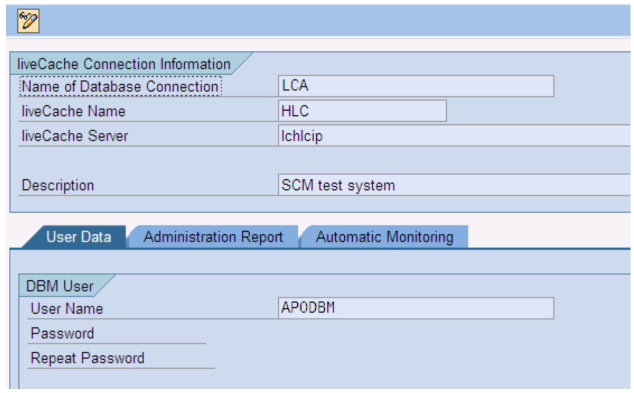
4.1.1 liveCache users which are critical - warning

The cluster solution is liveCache aware insofar as it must be able to access the liveCache instances and understand the status of the instances and control the start and stop of the standby service. In order to implement this, a control user is necessary. This user is selected during the installation of liveCache and once installed is not easily modified. The user and passwords are implemented as XUSER data for sdb. Ideally, these are not changed.

NOTE: If the cluster does not have a user which allows it to interface with liveCache the cluster and HotStandby behavior becomes unpredictable.

APO also requires a control user to manage the liveCache. This need not be the same control user. Rather than change the password, or make modifications to the cluster control user, it is easier to implement a new user for APO. This can be easily done with the dbmgui tool and then linked with the APO integration.





4.1.2 HotStandby and Archive Logging

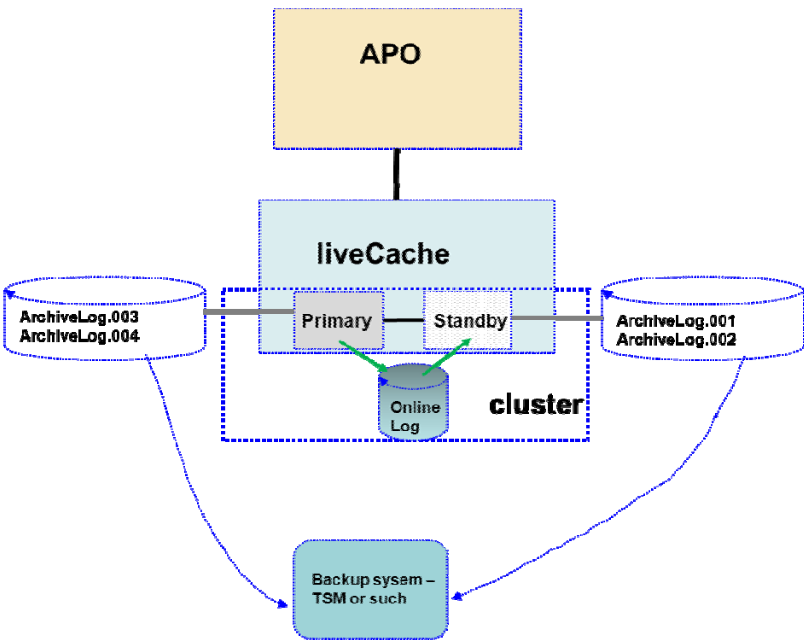


Figure 16: Active archive logging

For a HotStandby, only one liveCache instance is ever actively writing to the log and generating archive logs. However this role switches with the role of the instance. Only the active master has this role. When the standby takes over as master it takes over logging. This means that the the logs can be generated by either of the instances depending on the role. As the archive log speed has an effect on the liveCache performance, typically the first archive repository is local to the node. The log sequence is not repeated so that the consolidation of the two repositories will provide an ascending sequence of logs.

4.1.3 Application Administrator for APO and the Application View

Figure 17 shows the differences in the views to liveCache from the application level and from the instance level. APO is unaware of a clustered HotStandby; APO is only aware of the liveCache functionality which is either available or it is not available.

Using the DBMGUI, it is possible to see the state of each of the instances; which is the primary and which is in standby. Most liveCache administration is done from the DBMGUI to the primary liveCache.

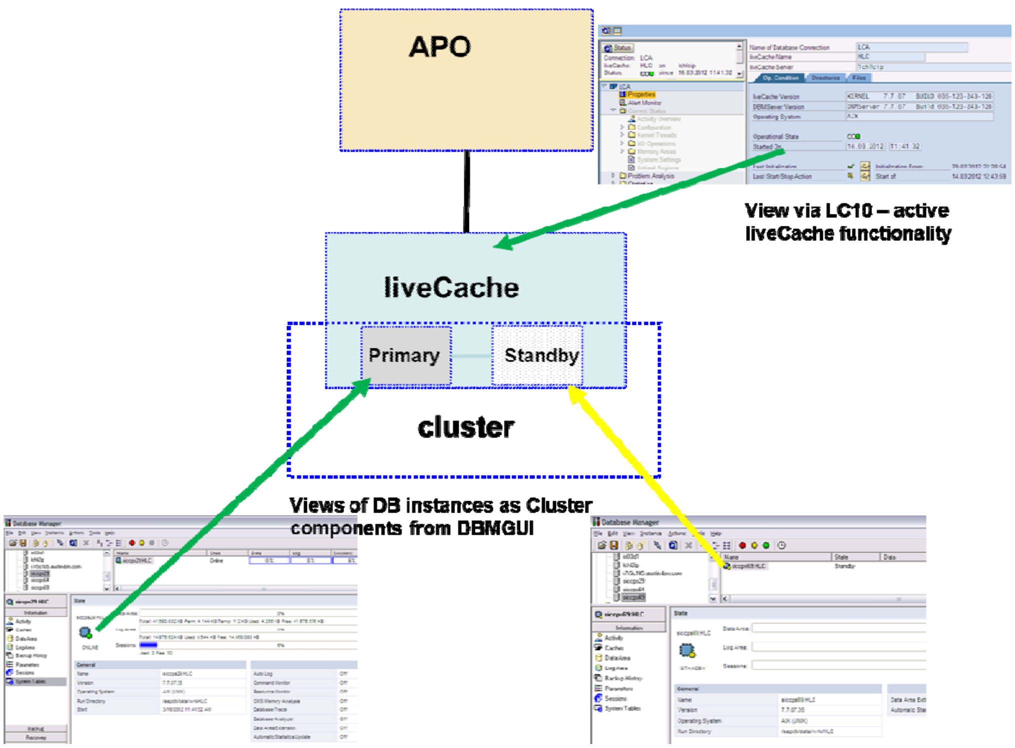


Figure 17: different views of the liveCache as function and as individual instances

Figure 17 shows the differences in the views to liveCache from the application level and from the instance level. APO is unaware of a clustered HotStandby; APO is only aware of the liveCache functionality which is either available or it is not available.
Using the DBMGUI, it is possible to see the state of each of the instances; which is the primary and which is in standby. Most liveCache administration is done from the DBMGUI to the primary liveCache.

APO retains control of the liveCache state. If APO shuts down the liveCache, the cluster will not attempt to reactivate it or take any action to switch to the standby. The cluster is aware of APO only insofar as it has knowledge of the state APO has set for the liveCache. The cluster waits in anticipation of APO restarting liveCache. Once restarted, the cluster will maintain it in active status through a failover to the standby if necessary. During the time liveCache is offline by APO request, the cluster continues to maintain all necessary prerequisites for restarting the liveCache when APO requests it. The cluster itself remains in active status. The cluster will failover the primary resource groups to the standby server if any prerequisite resource is lost on the node which the cluster is maintaining as the primary, but will not attempt to restart liveCache without direct order from APO. One of these resources is the IP address which is used by APO to contact the liveCache server and restart the liveCache service.

For example, if APO has shut down the liveCache, the server which was providing the active liveCache primary is maintained to restart as the active liveCache primary when liveCache is restarted. It retains the IP address and other prerequisites. If during the down time, this server is shutdown, the cluster will prepare the standby server to become the primary liveCache server. When APO restarts the liveCache activity, it will then be active on the server which was earlier the standby. This is totally transparent to APO.
When the second server is restarted, it will automatically reenter the cluster, and the cluster software will automatically re-integrate it as the new standby. This again is totally transparent to APO.

Where APO will see the effects of a failover is in the temporary loss of connection to the liveCache. This will cause work processes, which active at the time of the failover, to break and reconnect. In some cases this will lead to short dumps. Normally the first work process which realizes that a reconnect is necessary informs all the other WPs and they immediately initiate a reconnect.
Due to this behavior, the cluster temporarily stops the x_server until the service is online again. This allows the service to be reestablished quicker as the constant reconnect attempts do not reach the liveCache until it is able to accept them. This also ensures that no work process hangs, expecting a liveCache response that will never come due to the failover.

When APO shuts down the liveCache, it does this via a method which is aware of the HotStandby. LCINIT, initiated from LC10 will stop both primary and standby, and it will start both primary and standby. This is the only function with HotStandby awareness in APO and it is implemented via the provision of an additional script (see Figure 20 in section 4.3 for more information).

Based on this APO control, which is respected at cluster level, any administration task which focuses on liveCache alone can be carried out by stopping liveCache from APO. This disables the cluster reaction to any change of state in the liveCache instances and allows the master to be put into admin mode. Most of the maintenance procedures described here use this mechanism.

Switching the liveCache from Master to Standby Server

It is very simple to invoke a service switch from the current master to the current standby from the DBMGUI or from the command line. It is simply necessary to put the current Master to OFFLINE status (db_offline) and wait a minute or so until the monitor notices the state transgression and initiates a failover. As soon as the failover has taken place and the Standby has taken over as master, the old master will be reintegrated as the new standby. The change of states will be seen in DBMGUI. Note that a open DBMGUI connection to the standby will be broken as the x_server is purposely restarted during the takeover.

4.1.4 AIX Administrator and the Cluster View

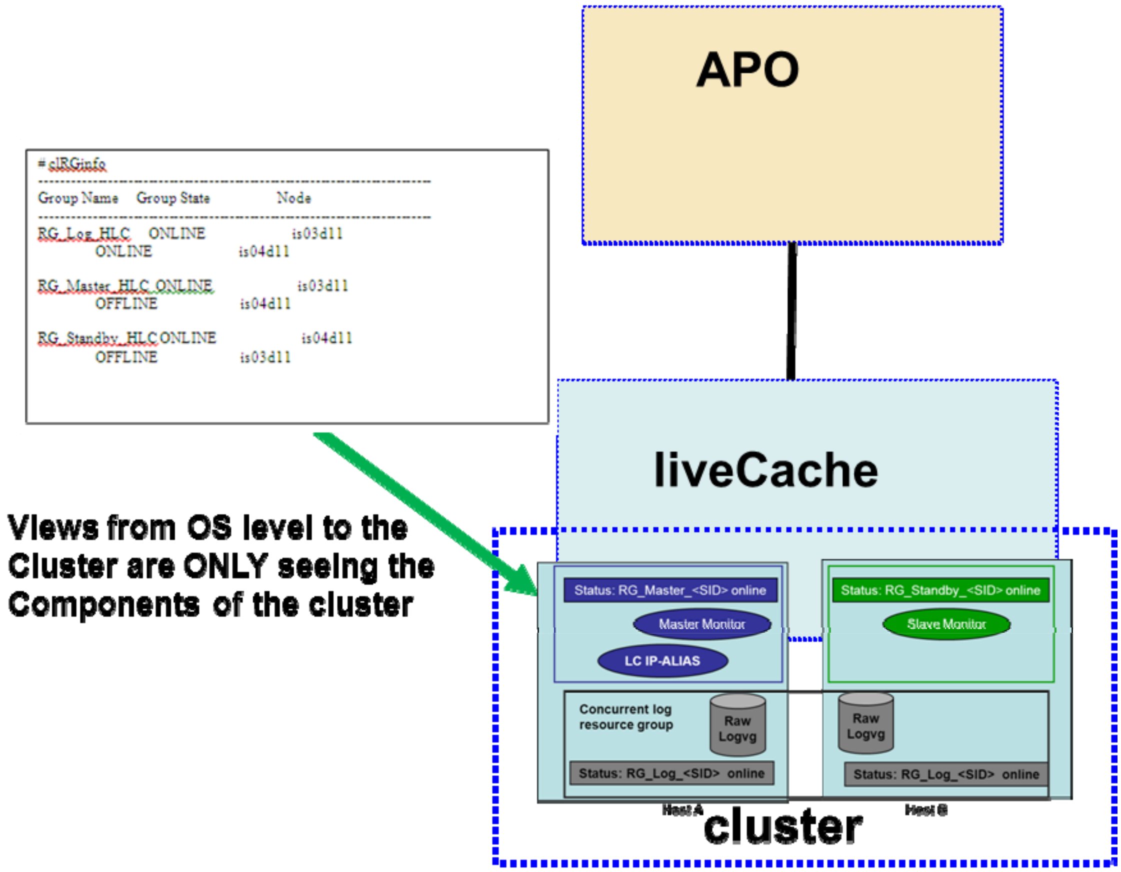


Figure 18: view of the cluster components

The tools used by the AIX administrator are looking at the state of the cluster only. The fact that the cluster is up and active does not mean that liveCache is active. If APO has stopped the liveCache, the cluster remains up and active. The view does not change.
To see that actual state of the liveCache DB instances, either the DBMGUI or the dbmcli command line can be used.

The IBM solution provides dedicated log files for the liveCache solution. The start, stop and monitor logs can be found in /var/hacmp/log/maxdbsa.log. The logging of the cluster hook is in /tmp/lccluster.log or /var/hacmp/log/lccluster.log.

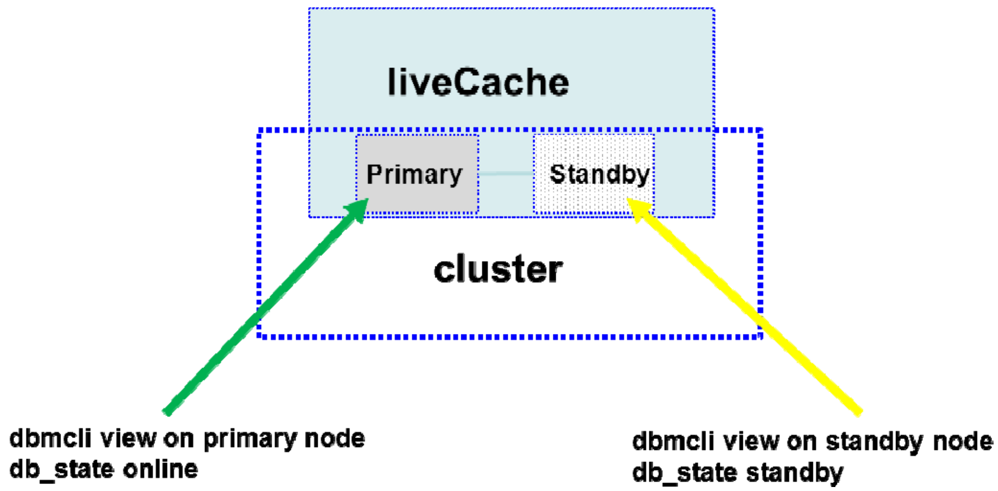


Figure 19: view of the liveCache status via database command line

In the example below, an XUSER has been provided to allow the OS administrator access to liveCache. The command line will deliver the state of the local instance – in this case the primary.

```
# dbmcli -U HLC_XUSER db_state
OK
State
ONLINE
```

Moving service to standby server (switching)

A Batch load will not survive this task, but can be directly restarted on base of a consistent liveCache. Therefore it is essential to plan for such tasks.

A graceful cluster switch is the less disruptive method to move the master instance to the standby node. In order to perform this a AIX administrator is required to move the resource groups. In the following a two step execution is provided:

1. The online on different node policy will not allow to directly move the master. The test implies to stop the node running the master through C-SPOC stop cluster services with “move resources” option.

```
                     Stop Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.
                                                        [Entry Fields]
* Stop now, on system restart or both              now
  Stop Cluster Services on these nodes             [is03d6]
  BROADCAST cluster shutdown?                      true
* Select an Action on Resource Groups              Move Resource Groups
```

2. Restart the node using C-SPOC

```
                      Start Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                  [Entry Fields]
* Start now, on system restart or both            now
  Start Cluster Services on these nodes           [is03d6]
* Manage Resource Groups                                Automatically
  BROADCAST message at startup?                         false
  Startup Cluster Information Daemon?             true
  Ignore verification errors?                     false
  Automatically correct errors found during       Interactively
  cluster start?
```

Expectation:
The rg_lc_standby will become STANDBY and establish a hot standby instance on is03d6, registered on Master.

To enable liveCache operations to perform this task as well a failover on base of injecting a failure to the master instance via dbmgui by stopping the service (db_offline) can be used. This causes the application monitors in the cluster to also cause a failover.
The difference to the graceful cluster switch is the time the APO requires to be back in production. The graceful cluster switch is about 30-60 seconds faster than doing this on base of injecting an error.

Stopping cluster services

Stopping PowerHA services can be done in different flavours.
Stop by bring all resources offline

This stops the infrastructure and the service. APO is not able to restart the service. DBMGUI can not connect to the instances.

Stop by unmanaging resources

This leaves the infrastructure up and leaves the service as is. Both are not protected anymore in sense of clustering. Both, APO and the DBMGUI can operate on the instances .

Stop with failover resources (only valid if one node remains up)

No matter if this task is performed on Standby or Master the master will be given priority and will survive. The standby will be brought offline.

4.2 Actions which cause a failover of the liveCache

4.2.1 Application Actions which cause a failover of the liveCache

Performing DBMGUI tasks in regards to change the state of the instance without stopping the application monitors will result in a failover in case it is done on the master node.
In case this is done on the Standby node the service will constantly try to recover the standby in place.
The same is also true when these tasks are performed via dbmcli.

4.2.2 OS level Actions which cause a failover of the liveCache

Killing the kernel processes of the instance will trigger monitor activity. If this is the master instance, a failover will result. If this is done on the standby, a recovery of the standby on the same will result. A standby never fails over on its own accord – it is only reintegrated on a recovering node after a failover of a master instance.

Stopping or rebooting the OS

Stopping the LPAR except when shutdown –Fr is used will cause a failover if done on the current master.

Depending how the OS is stopped the Cluster resource manager reacts differently.

a) shutdown –Fr
to be compliant with the request to shut down as quickly as possible, the resources are brought into unmanaged state and will be stopped along with the shutdown of the OS . This means the service is brought offline. No failover is initiated to the remaining node in case the node which is shutdown if the primary. At restart the cluster will attempt to reintegrate the node (if the cluster is set for auto restart which is recommended). If this was the master, it will restart as the master. In this case the failover to the prepared standby is bypassed and the master will need to rebuild the liveCache data structures in memory and refill the data.

NOTE: the master will not automatically reintegrate with the standby in this case!!! Therefore this is not a recommended approach. For automatic reintegration use one of the option in b.

b) All except shutdown –Fr (halt –q, shutdown –r, reboot)
In this case, the cluster will initiate a failover.

4.3 Concept of the primary in regards to resynchronization

Administration Procedures

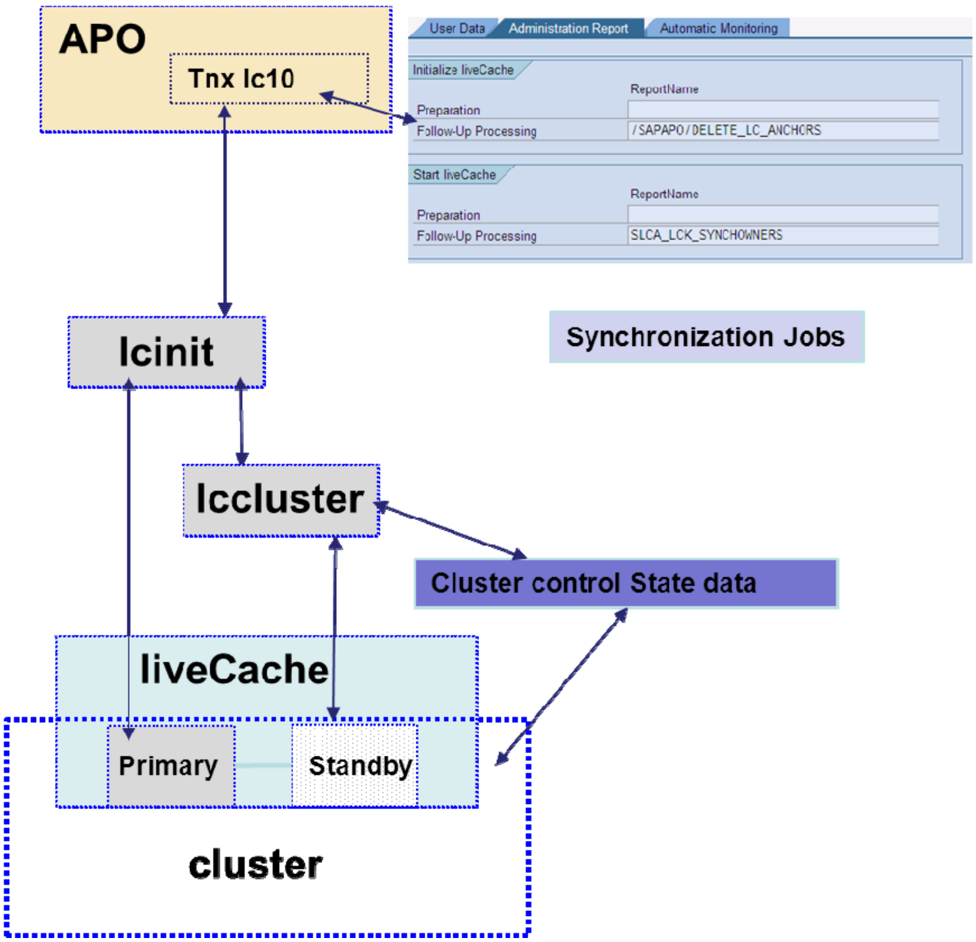


Figure 20: view of control path from APO to HotStandby

This section documents some of the most common maintenance procedures which have been tested to verify the cluster resilience and data consistency. The methods documented here can also be used for similar administration requirements.
Figure 20 depicts the flow of control from APO to liveCache. To ensure synchronization at the application level, the liveCache should be stopped and started from APO. The ensures that the follow-up processing reports are executed. Once shutdown, the liveCache can be administered using the command line or the dbmgui tool and when finished, it should be shut down and then restarted by APO.

APO invokes the script lcinit to manage liveCache. This script will manage the primary only as it is not aware of the cluster. To enable the cluster, lcinit will look for a cluster script and if one is found, it will pass activity information to this script. This allows management of the standby. The script lccluster maintains the cluster control status which determines whether the cluster reacts to a change in liveCache status or not.

If APO has requested the liveCache to be offline, the cluster will not react to changes in the liveCache status which in normal case would lead to a failover and service recovery.

NOTE: This is the recommended way to prepare for a maintenance window.

The script lccluster focuses on the state and the standby. It does not manage the primary. Therefore if this script is used, manual steps are required for the primary.

4.4 Migration from a single to a HotStandby liveCache

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 filesystem 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 on base of application monitoring. All cluster resource groups remain active.

4.5 Tested procedure for data transfer

The transition procedure described in this section can be used when moving from like to like, ie. from AIX to AIX. If the source liveCache, and the target liveCache are both on AIX, data transfer can be done via normal DB backup and restore. Where this is not possible, the APO upload/download path must be followed. This requires the data to be loaded from liveCache into the ABAP database of APO and then reloaded into the new liveCache.

The actual activities of the critical path insofar as downtime is concerned are indicated by the start and end of downtime markers.

4.5.1 Transfer Process Preparation

This procedure is based on using a backup of the current liveCache to load the new HotStandby liveCache. The new HotStandby is then attached to the production APO system.

Note: When doing a restore, it is necessary to start the restore activity from the liveCache instance which is the current active master. As the liveCache must be shut down from APO prior to starting this procedure, it should be first noted which is the actual master before shutdown. When the liveCache is stopped from APO, the IP alias will still remain on the node that was last master so this can also be used to determine the node which must be used for the data restore. When the liveCache is restarted from APO, the restart connection will target the node with this IP alias and this node should have the valid state of the restored data. The standby will then automatically resynchronize itself with the master such that the restored data is then propagated on both nodes. If the restore is done on the node which had been standby, then the attempt to restart liveCache will fail as the master node (with the IP alias) will have a data state which does not match the new log status and a master cannot resynchronize itself from a standby.

The target liveCache should be the node holding the IP alias as seen below in the netstat command output.

IP alias

```
# netstat -i
Name   Mtu    Network     Address            Ipkts Ierrs  Opkts     Oerrs   Coll
en0    1500   link#2      12.8c.9a.af.9b.2   2825766      0         351535  0   0
en0    1500   10.1        lchlcip            2825766      0         351535  0   0
en0    1500   9.153.164   is03d11            2825766      0         351535  0   0
:
:
```

4.5.2 Migration of an AIX Production liveCache to new HotStandby liveCache

>> Start of Downtime Marker – Beginning of DownTime <<<

1. **Backup of the source liveCache Data**

- Execute consistency check on APO with current liveCache to ensure valid state of the business data. (transaction /SAPAPO/OM17).
- Using transaction LC10, shutdown the current liveCache from APO – be sure to do this to avoid any unexpected connection attempts to liveCache from APO during the transition process.
- Using DBMGUI put the database into admin mode and backup the current liveCache as complete data backup.

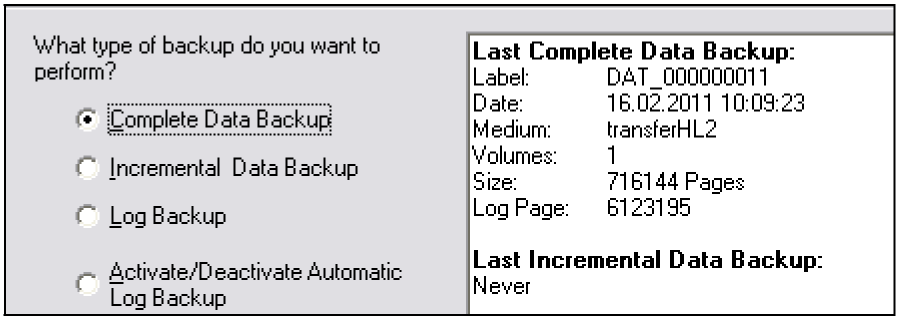
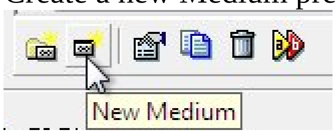


Figure 21: SAP DBMGUI, backup wizard

Create a new Medium press the following button shown in the picture below:



2. **Restore the liveCache data on the target (HotStandby) liveCache**

- Disable the cluster recovery logic either by shutting down from APO, or by simulating a shutdown from APO (see sections 5.1.1 and 5.1.2). As a result the cluster should be in following state:
  - Standby Service: OFFLINE
  - Master Service: ADMIN
  - LC10 state lock: LC10.stop
  - All RGs ONLINE

- Ensure the DefaultCodePage parameter is set to the same type as it was in the source. This liveCache parameter can either be ASCII or UNICODE. The default is UNICODE.

- Using DBMGUI, restore as image on new HotStandby liveCache.
  Using DBMGUI, restore the backup from the source liveCache to the HotStandby liveCache using the “Restore medium with initialization” option. Be sure to use initialization option to get rid of layout parameters which no longer match. When the restore is through select “Restart” and NOT “Close” to finish the task and bring the master instance online.

- Verify liveCache user settings
  There are two users which link the liveCache to APO. The DB administration user and the SAP user which is used for SQL and which owns the internal DB schema.
  The usual administrator is the user “control”. The SQL user is selected at installation time, the default is often sap<sid> or LCUSER. If the SQL user is different between the source and the target, some steps have to be taken to align this user for SQL access. The SQL user in the liveCache content must match the user in the APO integration. Both can be modified – changing the integration is easiest.

If control user and password are incorrect, these can be updated via DBMGUI. Delete any old instances and update the current with new user/password combination To do this you need superdba or DBM administrator authorization.

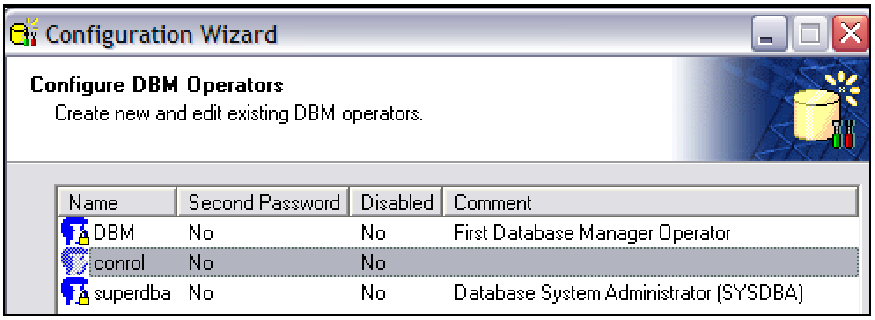


Figure 22: SAP DBMGUI –configuration wizard

Identify the DBM and liveCache user in APO transaction LC10 (Figure 23). In the PoC, the SQL user is LCUSER and the DBM user is CONTROL.

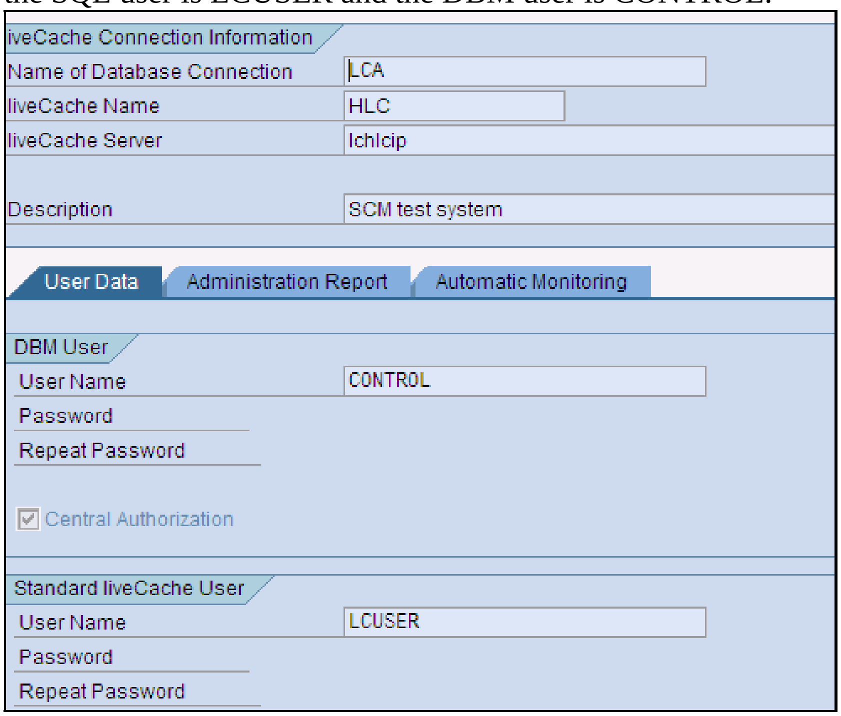


Figure 23: SAP APO transaction LC10 – integration configuration

In DBMGUI the database users can be displayed to verify the consistency between APO and liveCache database.

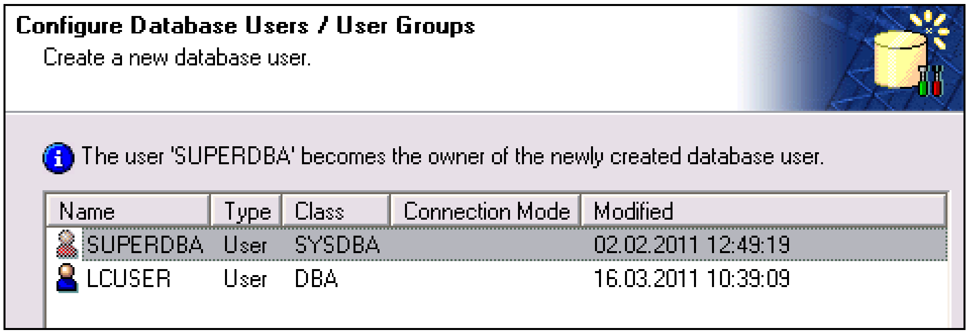


Figure 24: SAP DBMGUI – view of liveCache users

If a different user is desired as the APO SQL user press the Rename button in the "Configure Database users" panel:

Then modify the database schema by calling the following two commands from the dbmcli prompt

dbmcli > user_sap <sql user>, <pw sql user>
dbmcli > load_lcapps <sql user>,<pw sql user>

If the liveCache is to be connected to an APO system with a different SID then the one by which it was originally initialized, it is necessary to clear the old information. This could be the case if a test APO had been used during the liveCache verification tests. liveCache purposely rejects a connection by an unknown APO to avoid clones of the production APO system from automatically connecting to the production liveCache.

See SAP note number: 1015489 as reference.

- Shutdown the HotStandby for integration into APO.
Use dbmcli, issue command db_offline on the primary server. Standby should already be down from the beginning.

- From APO transaction LC10, modify the connection information.

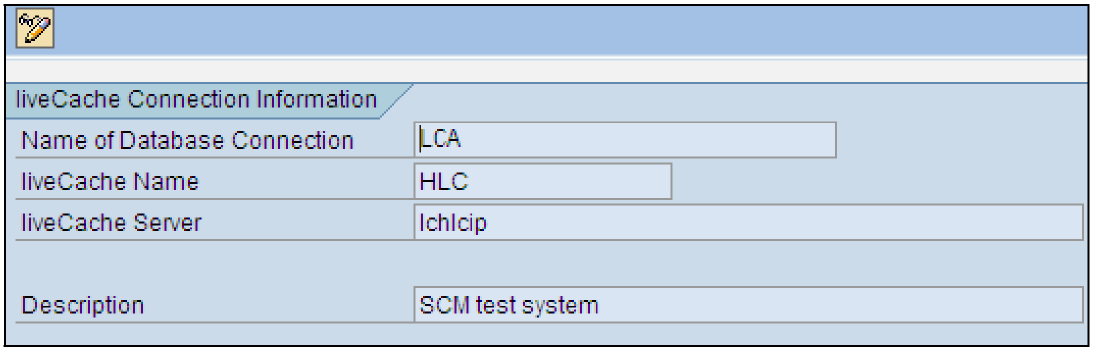


Figure 25: SAP APO transaction LC10 – connection configuration

For all three connections: LCA, LDA, LEA, change the liveCache name to the new HotStandby SID and use the service IP alias as the liveCache Server.

- From APO transaction LC10 start the liveCache
  When restarted, the standby will find itself out of synchronization (data does not match the new log status following the restore on the primary) and it will resynchronize itself via a new FlashCopy initiation. This should happen automatically.

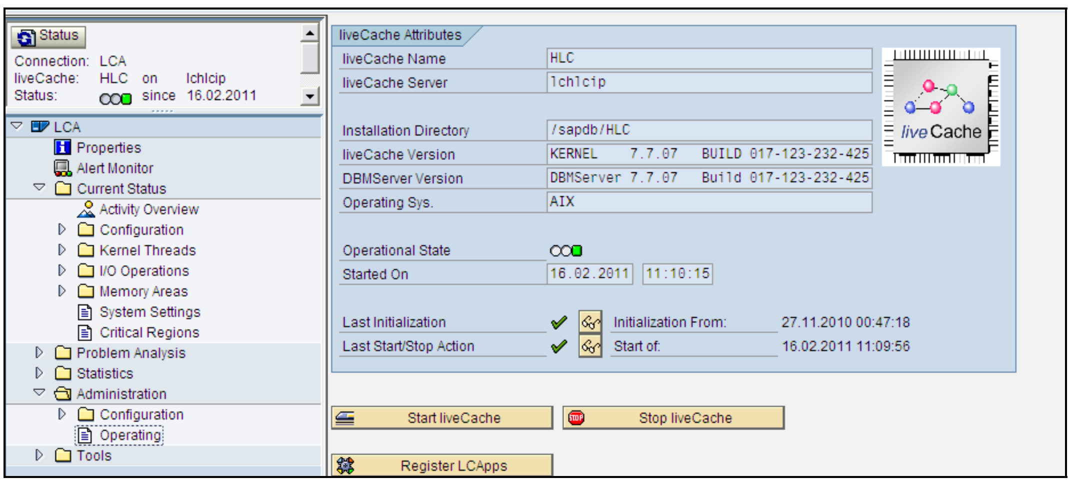


Figure 26: SAP APO transaction LC10 – liveCache operating

- In transaction LC10, select "Register LCApps", and reregister the applications for the new connection.

- Rerun the consistency checks to ensure the status of the liveCache is consistent (transaction /SAPAPO/OM17).

- Return APO to production if consistency checks are successful.
  If consistency checks bring irresolvable errors, the APO system can be reattached to the original liveCache by altering the integration parameters in LC10. This provides a failback strategy in case of a tight downtime window.

>>>> End of downtime marker – PRODUCTION CAN RESTART

4.5.3 Migration of Production non-AIX liveCache to new AIX HotStandby liveCache

To transfer data from a liveCache of different HW architecture to the AIX target HotStandby, the data must be moved via the APO application. The data is first loaded into the ABAP database. The source liveCache is then switched for the new HotStandby and then the data is loaded from the ABAP dataset into the target liveCache.
To do this, SAP requires that the liveCache logging be stopped to speed the loading process see Figure 27 from SAP.

NOTE: NEVER STOP LOGGING ON AN ACTIVE HOTSTANDBY PAIR. Before stopping the logging for any reason, be sure to shut down the standby first.

The HotStandby architecture and synchronization method relies on the shared log. If no logging is done, the standby has no chance or resynchronizing.
Special action must be taken before executing the SAP Method when the target is a HotStandby liveCache.

FROM SAP Method Description:
Section C: Upload (copying the master data and transaction data from the APO database to the liveCache)
Section D: Post processing tasks (perform these sometime after the upload)

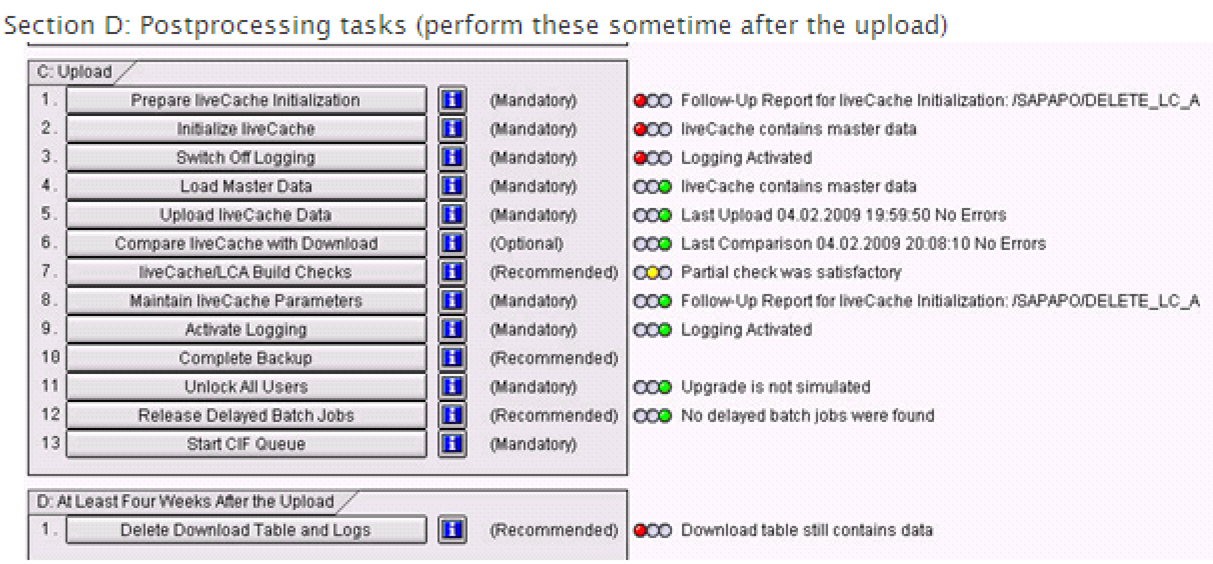


Figure 27: extract from the /SAPAPO/OM_LC_DOWNLOAD report overview

4.5.3.1 Performing /SAPAPO/OM_LC_DOWNLOAD on HotStandby

The standby must be disables and then reinitialized as it is not able to keep in synchronization without the log information. The difference between this procedure and the liveCache data restore from a backup is that in this case the master liveCache instance must be online to APO.

4.5.3.2 Necessary cluster status for upload to liveCache

This procedure differs from the backup/restore method in that the liveCache service must remain connected and available to APO. However with the logs disabled, we cannot have the standby service active.
In this case we need to leave the master in online state and connected to APO. On the standby, we need to suspend the standby application monitor and shut down the x_server. Alternatively, the standby LPAR or server can be shut down. The standby must stay down until the master has restarted logging. During the upload procedure, the liveCache master is stopped and restarted by APO.

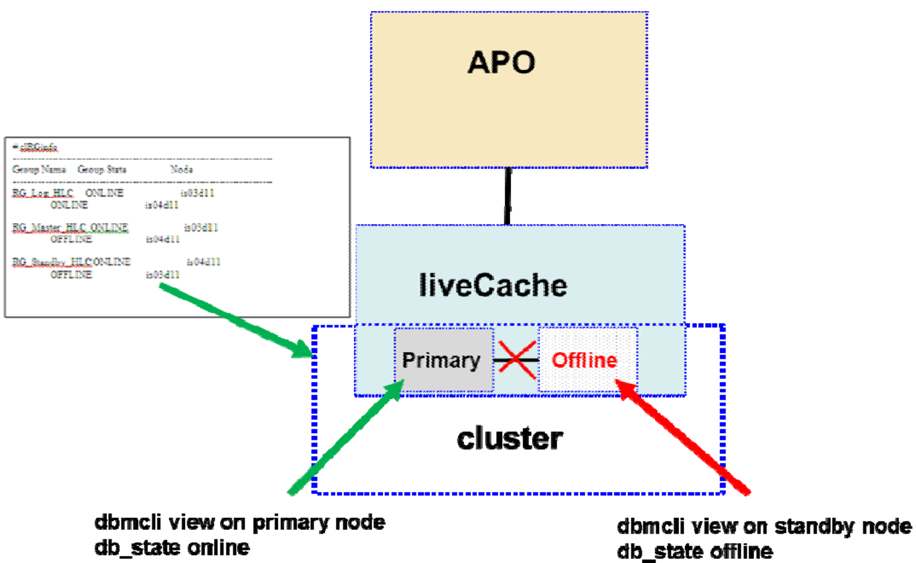


Figure 28: Status required for upload procedure

Deactivate the standby (OS Administrator)

- Under smitty-> hacmp->System Management (C-SPOC)->PowerHA System Mirror Services stop the cluster services for the node running the standby. This ensures no restart of the standby.
- Use the command line (dbmcli) or DBMGUI to stop the standby service.
- Stop x_server on the standby server
  This breaks the communications between the master and the standby. The master can now be used as a standalone instance for the purpose of data download. The procedure provided by SAP will need to restart the liveCache to deactivate the logging.

- Continue with the data load process to the point .When the data load is complete and the logging is restarted; the standby will need to be reinitialized. BE SURE LOGGING IS ACTIVE!
- Restart the x_server on the standby server
- Follow Reintegrating the Standby
  The reintegration process below will manage the re-initialization automatically.

Reintegrating the Standby

Restart the cluster services on the standby node with the option to automatically aquire resource groups. This will seamless reintegrate the standby.

```
                    Start Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

* Start now, on system restart or both              now
  Start Cluster Services on these nodes             [is04d11]
* Manage Resource Groups                            Automatically
  BROADCAST message at startup?                     false
  Startup Cluster Information Daemon?               true
  Ignore verification errors?                       false
  Automatically correct errors found during         Interactively
  cluster start?
```

4.6 APO Re-initialization of a liveCache with Hotstandby

This process requires collaboration between the APO administrator and the cluster or OS level administrator to ensure success. The reinitialization is used to force a new FlashCopy refresh on the standby side and to verify the FlashCopy mechanism.

The steps to be followed are the following:

1. APO Administrator – LC10 stop liveCache
   This step puts both the master and the standby offline and sets the LOCK information which informs the cluster that APO has stopped the liveCache.

2. OS Administrator - suspend the standby resource group monitors on the standby server. This is necessary as the re-initialization restarts liveCache before we are ready. This can be done by using the stop cluster services, and selecting the action "Unmanage Resource Groups".

```
                         Stop Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

  * Stop now, on system restart or both          now
  Stop Cluster Services on these nodes          [is04d11]
  BROADCAST cluster shutdown?                    true
* Select an Action on Resource Groups            Unmanage Resource Gro> +
```

3. Use dbmcli or dbmgui to stop the liveCache standby instance as the LC10 shutdown will not do this.

4. APO Administrator – LC10 Stop LC (puts all in a known state for restart)

5. OS Administrator – resume monitoring of standby resource group on standby server. This will prepare for the resynchronization when liveCache is restarted

6. APO Administrator – LC10 Start liveCache
   This action will reset the LOCK information informing the cluster that APO wants the liveCache online. It also activates the master liveCache and triggers the activity to restart the standby. This will lead to a resynchronization and bring the standby online with the master.

4.7 Manual Re-Initialization of a Standby (OS Administrator)

A manual re-integration can be used to force a new FlashCopy from primary to standby. This procedure is used in some procedures used to change the physical layout of the liveCache such as additional storage or log volumes.

The prerequisite of a manual re-initialization of a standby is the following:

a. No FlashCopy is active in the reverse direction (ie from standby to current master). If this is the case, you should wait until the FlashCopy is complete.
b. liveCache has been shut down from APO to avoid any cluster activity or APO intervention
c. Master is offline (this will be the result of APO shutdown)
d. Standby is offline
e. Logging is active on the master
f. Verify the x_server is online on the standby

The re-initialization of the standby is done from the master.
Enter dbmcli on the master instance and execute the following:

From the master instance, use the hss_execute commands to control the standby. From the active maters, bring the standby into admin mode, and then request the init.

```
dbmcli -n master_node ...
db_online (or verify that master is online with db_state)
hss_execute <standby_node> db_admin
hss_execute <standby_node> db_execute init standby
```

To verify that standby is in a status where it can be reconnected view the following:
Using <independent program path>/bin/protconv dump the kernel information in the LC work directory (work director example example /sapdb/data/wrk/<SID>)

Example path/program: /sapdb/programs/bin/protconv

Look in the output for: “RTEHSS <nr>: HSS Initialization succeeded.”
Look in SVC or DS8k flashcopy activities verify the newly initiated FC consistency group.

When we are sure that the FlashCopy has been successfully restarted (either of the above it true) we can restart the standby service as described below.

Stop and restart liveCache from APO. This will ensure that the HotStandby cluster is brought up with each node in the proper status.

4.8 Single node Operation

In the case that maintenance is planned for a server, operation of the liveCache can continue with a single server. Simply shut down the node, or shut down the cluster on the node that is being removed from service.

If this is the primary, a failover will be initiated which will bring the standby into service as primary liveCache server. Doing this will break the connection between APO and liveCache and force a reconnect, so this should be done in collaboration with the APO administrator to ensure that this causes no inconvenience. Open transactions will be broken by this disruption. Online users may get a short dump and can reenter the transaction as soon as the connection is reestablished. Batch jobs will need to be restarted, and therefore planned maintenance should be done outside the window used for batch.

If the node is the standby, then the removal of the standby from service is totally transparent to the application.

When finished, the server can be simply restarted and the automation will reintegrate the cluster and the liveCache standby. If the cluster has simply been shut down on the server, then a restart of the cluster will reintegrate the cluster and the standby service.

4.9 Upgrading a liveCache with HotStandby

A rolling upgrade for the liveCache code is not yet available. The current process for liveCache HotStandby application does require a short downtime and upgrade activity to be done on both database installations. In the current version (7.7.07) the master sends the current parameter set to the standby when it starts the standby. These parameter settings include the database version. The standby cannot start if the kernel version does not match the parameter setting provided by the master and therefore both must be on the same level to establish a HotStandby.

The following procedure from SAP was used to upgrade the HotStandby liveCache at level 7.7.07.

Procedure steps:

1. Stop liveCache from APO to stop any cluster activity
2. Start SDBUPD on master
3. Run SDBINST on standby
4. Restart liveCache from APO. This will cause the cluster to restart both master and standby.

liveCache 7.7 is used for SCM 5.1 and SCM 7.0 and SCM 7.01.
liveCache 7.9 is used for SCM 7.02 and 7.03 and possibly beyond.

Upgrades from a release < SCM 7.0 to SCM 7.0 or higher need the download-upload procedure as data layout has been changed over these releases. In this case see the procedure for Upload/Download.

For liveCache implementations from SCM 7.0 it is possible to upgrade to a higher Enhancement Packages (7.01, 7.02..) without initializing the liveCache and therefore the download-upload procedure is no longer needed.

NOTE: In this case we would recommend a backup before upgrade which is best practices for any database level upgrade.

4.10 Adding Disk Capacity to a HotStandby liveCache

Adding disk capacity to a HotStandby is one of the more elaborate procedures in the lifecycle administration. The recommendation is therefore (when possible) to build a HotStandby with sufficient capacity in the beginning, and then only add capacity as necessary in large increments. The easiest procedure to follow is to add a new storage volume group (VG) to the existing configuration, as this will allow the process to be done with minimal downtime. In this procedure, as documented here, there is one short planned downtime using this method.

This procedure requires collaboration between the AIX administrator, storage administrator, and the APO administrator for liveCache. The procedure to extend the storage by adding new storage devices requires new devices to be added into the configuration on both servers and synchronized. The activity is broken down into 3 main activities according to the collaboration requirements and timing. This procedure will require a single scheduled downtime in the liveCache. This is necessary to update the HotStandby library configuration which is only activated at start-up. Below is a summary of the activities which are detailed with examples in the following section.

4.10.1 Summary of Steps

This section lists the substeps within the main activities.

PREPARATION

OS and Storage Preparation (all this can be done in advance and has no influence on the HotStandby service)

1) Prepare the storage and attach the new storage (in FC pairs) to the two servers.
2) Prepare the storage layout on the primary, including the creation of the logical volumes which will be used as raw devices by liveCache.
3) Execute the initial FlashCopy of the prepared disks and then import the new volume group on the standby.
4) Add the new devices into the FlashCopy profile used by the HotStandby to ensure that these new disks are included in any storage refresh.
5) Verify authorizations for the new raw devices on both sides to ensure liveCache can access them.
6) Provide raw device information (name and size) to LC administrator.

Planned Down Time

APO or liveCache administrator in collaboration with APO administrator

7) Restart liveCache from APO (LC10) to activate the new FlashCopy parameters used in the HotStandby re-initialization.

The next steps can follow at any time when it is convenient. From this point we have periodic disruption to the standby, but primary continues undisturbed.

Admin Activities on the Standby

OS level cluster administrator, LC administrator support and, if possible, storage administration to view FlashCopy activity.

8) Disable the cluster on the standby server to avoid recovery activity
9) Set the standby liveCache offline using DBMgui or dbmcli
10) At this point it is recommended to test the new FlashCopy configuration by forcing a re-initialization of the standby. Verify that all FlashCopy pairs are active as expected, including the new pair(s).
    At this point you can restart the standby service and continue the next steps when convenient, or continue directly.

Adding and Configuring the New Volumes to LC

LiveCache Administrator

11) Disable the cluster and set the standby liveCache offline (if not already in this state from step 10)
12) Using DBMgui add the new LC devices to the primary. The new storage devices will be mapped onto the raw devices provided by the AIX administrator.
13) When all devices are configured on the primary, resynchronize the liveCache standby by forcing a re-initialization of the cluster.
14) Reactivate the cluster services on the standby and trigger the standby service recovery by putting the standby offline. The cluster will trigger recover the service.

Service is now fully operational.

4.10.2 Detailed Steps

This section shows details and examples of the steps required.

4.10.2.1 Preparation

AIX & Storage administrators

This section shows examples of the procedure summarized above. In this example the two servers are is03d11 and is04d11.

The storage preparation has been done providing a new volume to each of the LPARs. The volumes are identical in size and have the following mapping:

Server	Local Disk	SVC_Volume
Is04d11	hdisk1	siccps69_data01
Is03d11	hdisk3	siccps29_data01

Is04d11 (alias siccps69) is the primary server

```
Starting Status of Source Disk on IS04D11
          PVID                           Adapter          SVC_Volume
hdisk1   00f641d449fcd27a  None    siccps69_npiv    siccps69_data01
Volume id:  600507680185000D38000000000000FF
```

As this pair is already been used for a flashcopy, the PVID on both sides is currently the same. To simulate new disks, we reset the volume id to better track the FlashCopy success.

```
chdev -l hdisk1 -a pv=yes
hdisk1         00f641cdeb49c985
```

Create new volume group for this disk: The liveCache currently has lcdatavg01 and lcdatavg02. We add the volume group lcdatavg03.

```
# lsvg
rootvg
sapdbvg
caavg_private
lclogvg01
lclogvg02
lcdatavg01
lcdatavg02
lcdatavg03
```

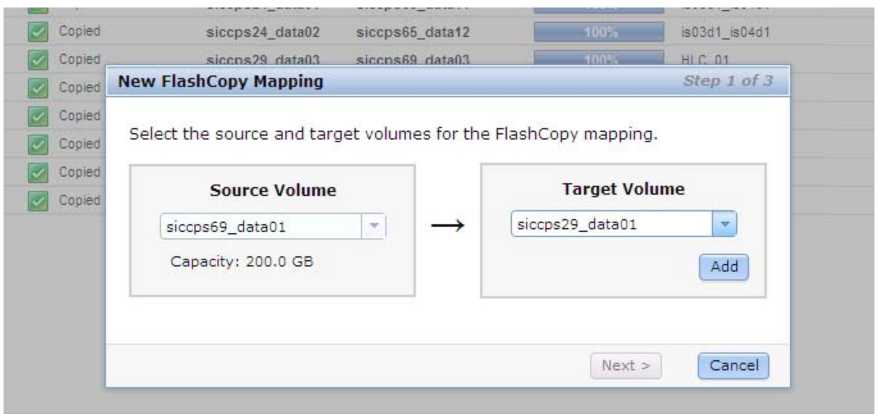
```
lcdatavg03:
PV_NAME   PV STATE TOTAL PPs   FREE PPs    FREE DISTRIBUTION
hdisk1      active        799             799         160..160..159..160..160
```

This volume group has a partition size of 256MBs and 799 partitions.

We then add the new logical volumes which will be used by liveCache. Each logical volume represents a device to liveCache and it is better to have several small devices then one large one. The number of I/O threads is calculated depending on the number of volumes. In this example each volume is 64GB in size. The volumes cannot exceed 128GB which is a liveCache limit.

```
# lspv -l hdisk1
hdisk1:
LV NAME              LPs     PPs     DISTRIBUTION          MOUNT POINT
lcdatalv05           250     250     160..00..00..00..90   N/A
lcdatalv04           250     250     00..00..69..160..21   N/A
lcdatalv03           250     250     00..160..90..00..00   N/A
#
```

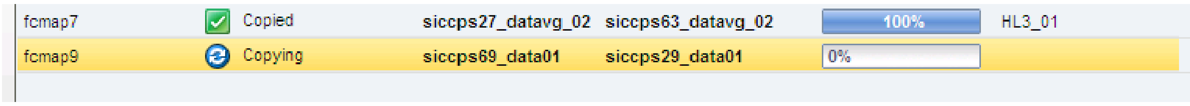
When all the structural changes are complete for the new volume(s) an initial FlashCopy is required to propagate the changes to the second node. In our example we have done this by creating a simple mapping on the SVC between the disk on which the changes have been done and the target disk on the 2nd node. This mapping and FlashCopy has no impact on the HotStandby at this point.



The FlashCopy must be a clone – a complete and self-sufficient copy of the original.

Once the FlashCopy is triggered, the new volume group can be imported on the second node. (**NOTE**: You do not need to wait until the copy is finished to continue, but it should be finished before any further FlashCopy activity is attempted which involves these volumes.)

This snapshot shows the copy activity on the SVC for the new disk pair.



Preparing the Second Node

On 2ndary, remove the device and rediscover it with the new PVID created by the FlashCopy. In our example, on is03d11, the target disk is hdisk3.

```
# rmdev –dl hdisk3
# cfgmgr
# importvg -y lcdatavg03 hdisk3
lcdatavg03
```

Now we can see the new volume group structure including the 3 new logical volumes.

```
# lsvg -l lcdatavg03
lcdatavg03:
LV NAME         TYPE     LPs    PPs    PVs  LV STATE        MOUNT POINT
lcdatalv03      jfs      250    250    1    closed/syncd    N/A
lcdatalv04      jfs      250    250    1    closed/syncd    N/A
lcdatalv05      jfs      250    250    1    closed/syncd    N/A
```

At this point, the basis work is finished and the volume groups are identical on both servers.

Preparing for liveCache Access to New Volumes

The new logical volumes will be used as raw devices by the liveCache. Therefore these raw devices must belong to the the liveCache OS user. **On both nodes**, set the necessary authorizations for these new raw devices. In the example below we see the existing devices with sdb access and the new devices with root access. Set the access for the new devices.

```
# cd /dev
# ls -ltr rlc*
crw-rw----   1 sdb    sdba     40,  1 Apr 16 10:14 rlcloglv02
crw-rw----   1 sdb    sdba     38,  1 Jun  6 13:06 rlcdatalv02
crw-rw----   1 sdb    sdba     37,  1 Jun  6 13:06 rlcdatalv01
crw-rw----   1 sdb    sdba     39,  1 Jun  6 13:06 rlcloglv01
crw-rw----   1 root   system   41,  3 Jun 14 16:16 rlcdatalv05
crw-rw----   1 root   system   41,  2 Jun 14 16:16 rlcdatalv04
crw-rw----   1 root   system   41,  1 Jun 14 16:16 rlcdatalv03
```

```
# chown sdb.sdba rlcd*
# ls -ltr rlc*
crw-rw----   1 sdb    sdba      40,  1 Apr 16 10:14 rlcloglv02
crw-rw----   1 sdb    sdba      38,  1 Jun  6 13:06 rlcdatalv02
crw-rw----   1 sdb    sdba      37,  1 Jun  6 13:06 rlcdatalv01
crw-rw----   1 sdb    sdba      39,  1 Jun  6 13:06 rlcloglv01
crw-rw----   1 sdb    sdba      41,  3 Jun 14 16:16 rlcdatalv05
crw-rw----   1 sdb    sdba      41,  2 Jun 14 16:16 rlcdatalv04
crw-rw----   1 sdb    sdba      41,  1 Jun 14 16:16 rlcdatalv03
```

4.10.2.2 Updating the Hotstandby Profile

Now add the new disk pair(s) into the hotstandby configuration file – enter the name of source and target as they are known on the SVC. To do this, edit the configuration file which is found at location /opt/ibm/ibmsap/<SID>. The file name is RTEHSS_config.txt

Modify the following lines adding matching pairs – MlC is primary, SlC is secondary. It does not matter which is actually doing this role at the time of setup, the FlashCopy can execute in either direction. This a logical separation, ensuring that all the disks are going in the same direction – from M to S or from S to M.

In our example we add the master volume siccps29_data01 and secondary volume siccps69_data01. Do no leave any spaces between the entries – they are delimited by comma only.

```
MlCDataVdiskID siccps29_data01,siccps29_data02,siccps29_data03

SlCDataVdiskID siccps69_data01,siccps69_data02,siccps69_data03
```

Make duplicate modifications on both nodes, or copy the profile from one node to the other. If you do this, be sure the authorizations are not changed in anyway.

NOTE: When the initial manual FlashCopy is finished, be sure that any FlashCopy mapping is cleaned up so that nothing is left which might inhibit the actual hotstandby resynchronization which will be using the FlashCopy pairs.

The preparation work is now complete. The next step is the activation of the configuration changes in the library, and this will require a planned downtime to stop and restart the liveCache cluster. This should be done from APO via LC10.

4.10.2.3 Admin Activities on the Standby

OS level cluster and LC administrators. Storage administrator to view FlashCopy activity.

1) Disable the cluster on the standby server to avoid recovery activity
2) Set the standby liveCache offline using DBMgui or dbmcli
3) At this point it is recommended to test the new FlashCopy configuration by forcing a re-initialization of the standby. Verify that all FlashCopy pairs are active as expected.
   At this point you can restart the standby service and continue the next steps when convenient, or continue directly.

Deactivate the standby (OS Administrator)

This is used for preparation for an administration task on the standby server. During an admin activity of this kind, the standby service is not available. The primary is not impacted.

Setting Cluster to Unmanaged

Under smitty-> hacmp->System Management (C-SPOC)->PowerHA System Mirror Services stop the cluster services for the node running the standby. This ensures no restart of the standby.

```
                                          Stop Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                          [Entry Fields]
* Stop now, on system restart or both                   now
  Stop Cluster Services on these nodes                 [is03d11]
  BROADCAST cluster shutdown?                           false
* Select an Action on Resource Groups                   Unmanage Resource Groups
```

Verify status is unmanaged and then continue with the test of the re-initialization.

clRGinfo –m

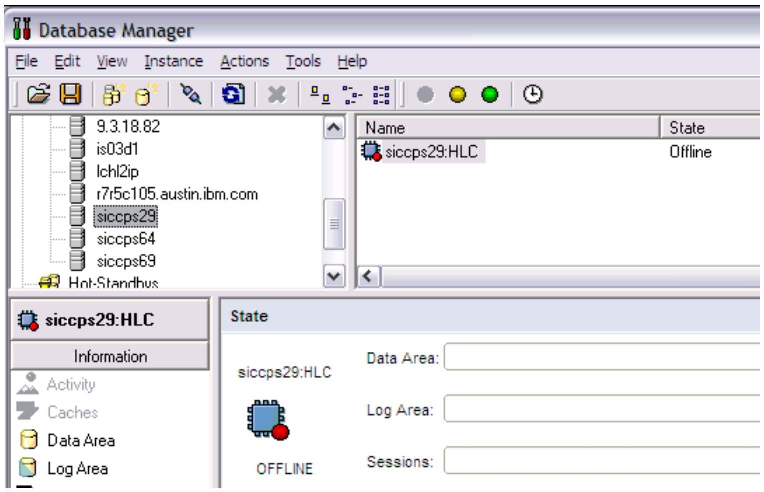
```
Group Name     Group State                  Application state           Node
----------------------------------------------------------------------------------
--
RG_Log_HLC     UNMANAGED                                                is03d11
 as_log_HLC                                  ONLINE MONITORED

RG_Log_HLC     ONLINE                                                   is04d11
 as_log_HLC                                  ONLINE MONITORED


RG_Master_HLC  ONLINE                                                   is04d11
 as_master_HLC                               OFFLINE


RG_Standby_HLC UNMANAGED                                                is04d11
 as_standby_HLC                              ONLINE MONITORED

RG_Standby_HLC UNMANAGED                                                is03d11
 as_standby_HLC                              ONLINE MONITORED
```

The cluster is now ready for a handover to the liveCache administrator for a re-initialization of the standby.

liveCache Administrator – Re-initialization of Standby

We would recommend now stopping the standby service to verify that the cluster is “at peace”. Once stopped under these conditions, it should remain stopped.

Use the command line (dbmcli) or DBMGUI to stop the standby service.



Test the Flash-Copy setup by invoking a re-initialization.

```
dbmcli on HLC>hss_execute is03d11 db_admin

OK

dbmcli on HLC>hss_execute is03d11 db_execute init standby

OK
```

This activity will force the new FlashCopy and all of the involved disk mappings should now be active. We are now ready to implement the new storage. This step can be done whenever convenient, there is no dependency. The standby service can be restarted or the next steps to implement the new storage and propagate the changes can be done immediately. If the standby is restarted, the new disks will be part of the FlashCopy refresh, but will simply not be in use by liveCache.

Mapping Name	Status	Source Volume	Target Volume	Progress	Group
fcmap0	Copied	siccps24_data01	siccps65_data11	100%	is03d1_is04d1
fcmap1	Copied	siccps24_data02	siccps65_data12	100%	is03d1_is04d1
fcmap2	Copying	siccps69_data02	siccps29_data02	73%	HLC_02
fcmap3	Copying	siccps69_data03	siccps29_data03	73%	HLC_02
fcmap4	Copied	siccps27_datavg_04	siccps63_datavg_04	100%	HL3_01
fcmap5	Copied	siccps27_datavg_03	siccps63_datavg_03	100%	HL3_01
fcmap6	Copied	siccps27_datavg_01	siccps63_datavg_01	100%	HL3_01
fcmap7	Copied	siccps27_datavg_02	siccps63_datavg_02	100%	HL3_01
fcmap8	Copying	siccps69_data01	siccps29_data01	7%	HLC_02

In our example, we see the result of the re-initialization in the successful FlashCopy activity of all three of the participating disks. FCmap8 is the new disk pair.

4.10.2.4 Restarting the Standby Service – AIX and liveCache Adminstrator

Once the FlashCopy has been started successfully, restart the HACMP services on the standby node. This will activate the monitoring.

```
                                             Start Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                                   [Entry Fields]
* Start now, on system restart or both                              now
  Start Cluster Services on these nodes                            [is03d11]
* Manage Resource Groups                                            Automatically
  BROADCAST message at startup?                                     false
  Startup Cluster Information Daemon?                               false
  Ignore verification errors?                                       false
  Automatically correct errors found during                         Interactively
  cluster start?
```

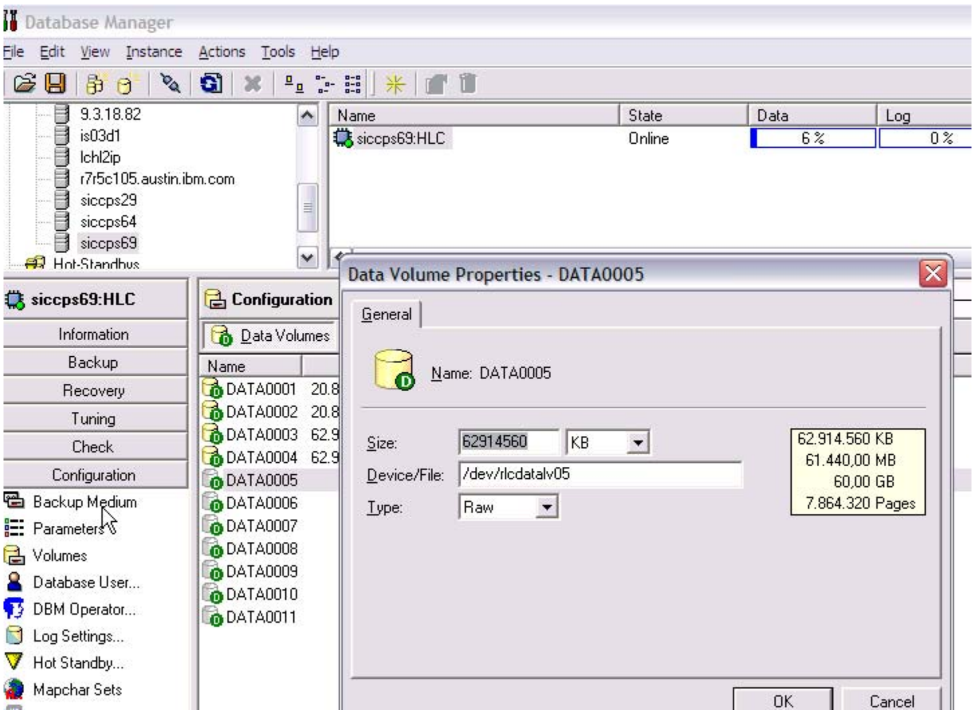
As the standby is in admin state following the re-initialization, it must be stopped in order to trigger automatic recovery. Using DBMgui or dbmcli and set the standby in in offline mode (db_stop). The cluster will recognize that the standby is in an invalid status and restart the service.

4.10.2.5 Activating the Storage on Primary – liveCache Administrator

The next two activities should be done in a period of low liveCache activity. Ideally we would like to implement and propagate the storage changes quickly to resume the standby service as soon as possible. The changes will take place on the primary and the standby service must be stopped as these changes will be incomplete on the standby side until the re-initialization propagates them. Consider that the raw devices themselves are known to the standby and the configuration information on their use will now be known to the standby (this information is cluster wide) but the actual formatting of the raw devices has happen only on the primary. If any data is placed on these new devices on the primary side, the standby cannot find it.

Standby: As in the preceding steps, set the standby offline by setting the cluster to unmanaged state and stopping the liveCache standby.

Primary: On primary, add new data volumes. Below is just an example to demonstrate the end-to- end activity. The actual size of the new devices must be properly calculated from the number of partitions in the device and the partition size. This information should have been provided by the OS administrator who setup the logical volumes. The volume cannot exceed 128GB as this is an LC limit.

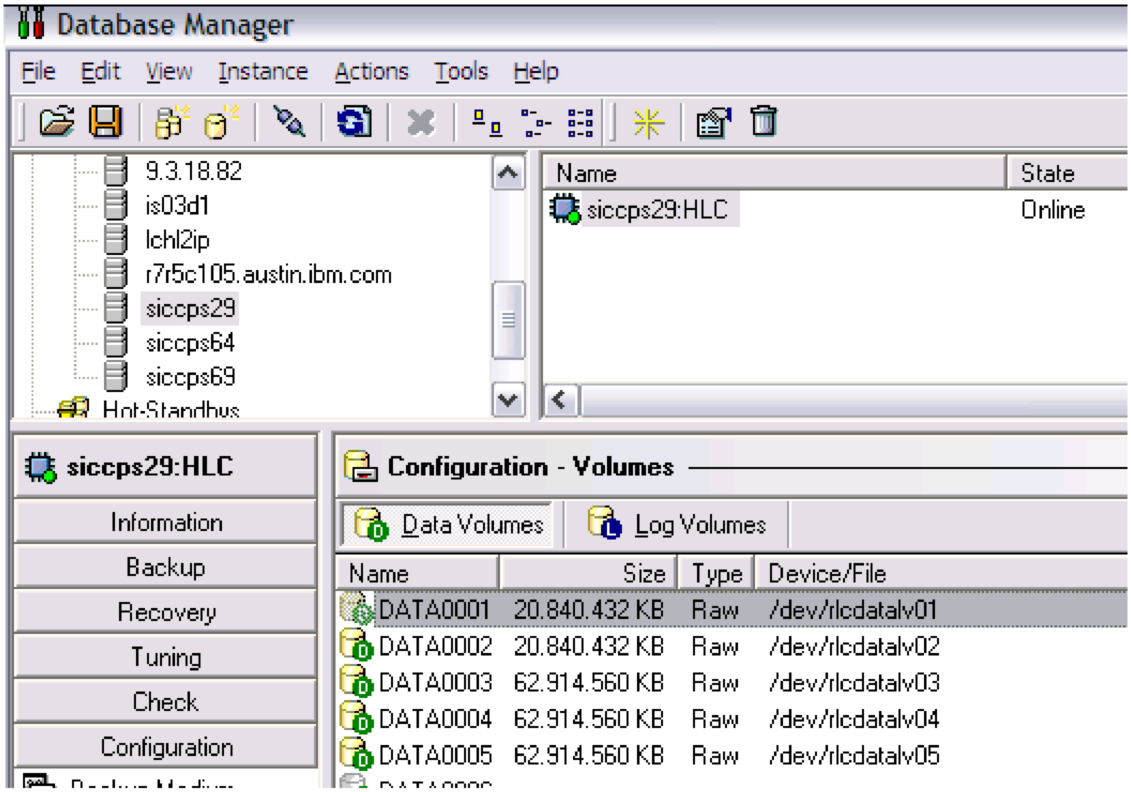


Primary: To ensure that the formatting action done on the volumes is carried over to the standby, a final re-initialization is necessary. As soon as the new device configuration is complete, repeat the steps to reinitialize the standby from the primary liveCache.

```
dbmcli on HLC>hss_execute is03d11 db_admin

OK

dbmcli on HLC>hss_execute is03d11 db_execute init standby

OK
```

Standby: Restart the cluster services on the standby server as done previously. In this case the standby will still be in admin status. Put the liveCache standby in offline status to signal to the cluster that a restart is required. The cluster will restart the standby services.

All modifications have now been propagated to the standby, and the service is fully restored.



View to the devices from the standby server.

4.11 Changing liveCache parameters

Changing liveCache parameters can be done as per usual. The new parameters are set in the primary liveCache instance using DBBGUI or dbmcli and these are reactivated via a restart of the liveCache. One the parameters are complete, wait for a refresh maintenance window and restart the liveCache vial APO transaction LC10. The parameters are set once and active for both instances.

5 General 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.

Normally the cluster will react to any unexpected change of the liveCache instance status or failures in the infrastructure which are not transparent to the application. The service monitor logic will view any change of an online master with suspicion and react accordingly. If the master suddenly changes status from online to offline or admin, the cluster will initiate a failover.
For the master, the ADMIN status is not tolerated once the master has achieved the ONLINE state as this has makes no sense to the APO application.

NOTE: The admin status is tolerated in unlimited fashion for a standby. This is done to ensure that nothing interrupts a long admin procedure such as resynchronization from log or FlashCopy initialization. If for whatever reason the standby instance is left in admin status, the cluster will NOT attempt to recover it. It is considered to be an intentional state. The one deviation to this is during a failover, when the cluster attempts to recover the master service on the standby. If the standby is in admin mode, the cluster will attempt to bring it online regardless as it has no other option for re-establishing the service.

5.1 Disable cluster recovery for maintenance (ADMIN mode)

This step assumes that a liveCache is running in an APO environment as a cluster and there is a window of time for administration requirements. This might be a step to restore the liveCache or to generate an offline backup using DBMGUI or command line to execute the administration tasks.
There are different methods for bringing the cluster resources into a “non-reaction” mode to allow for maintenance tasks.

The following resource group actions are considered as maintenance activities. The scripts try to protect the application against user errors during maintenance but cannot entirely achieve this. Therefore, to safely perform maintenance please keep the following aspects in mind:

a. The APO application is not aware of the cluster, just the service
b. The cluster has certain expectations of the application status which are necessary for the cluster to achieve the reintegration.

Therefore the sequence of activities is very important.

ATTENTION: Standby Node Maintenance via RG Offline
It is not recommended to control the service through the RGs as a PowerHA cluster administrator without performing a clean handover from the liveCache administrators. However, if there is some unavoidable reason for attempting this, care must be taken in the sequence to ensure that the standby can be reintegrated as a service after maintenance.
This sequence of events takes the standby out of active service insofar as APO support is concerned, leaving the master service online but not protected from a failure. It does however provide a method for a rolling maintenance.

If the standby service is still online when the standby RG is restarted, the application monitor will detect the state STANDBY and will not call the start script. This results in the following error condition:

- A missing x_server (the database listener) resulting in lost connectivity.
- No connected between standby and master resulting in a cold restart in case of failovers

Never bring the standby RG offline without ensuring the liveCache service is offline as well. To bring the standby service offline the follow options are possible in case it has to be enforced:

1. Procedure using RGs
    - Bring the standby RG offline to disable and restart attempt
    - Either use dbmcli or DBMGUI to stop the standby, or bring the log RG offline on the standby node – this will stop standby service as well. The actual stop of the service is performed by the stop script of the log.

2. Procedure using Cluster Services control
   If maintenance on the Standby node is performed and the log is offline, this can expose the master to a potential failover to a node with no log access. This will fail and result in a ping pong of this resource between nodes, potentially interrupting the maintenance tasks as well. In cases where this node is not a valid failover node for a master, the cluster services for this node should be shut down.

Best Practice: maintenance in a clustered liveCache environment

The procedure to follow is to

- deactivate the HA protection of the service to avoid unexpected restarts or failovers. There are two options:
    - Set LC10.stop by stopping through APO or simulate this as described in the following sections → This disables the reaction of the application monitors. The cluster will still react on infrastructure failures.
    - Unmanage Cluster RGs (section 5.1.3) → This suspends the reaction to application failures AND infrastructure failures.
- perform the work to be done

- reactivate the HA protection and verify that the hot standby pair is functional by reverting the task done to stop cluster surveillance.
    - Restart through APO. Note: The start must be mandatory done through APO to ensure the initialization tasks required to validate the content.
    - Manage the RGs in the same order as they have been before or ensure RG startup sequence.
- Ensure the Standby is restarted and reconnected to a working Master with a consistent content (sanity check).

There are different maintenance tasks and not all require to fully unmanaged the RGs. In the following sections different options are described.

5.1.1 Invoking LCINIT to simulate APO transaction LC10

It is possible to use LCINIT directly to shut down or start the liveCache for simulation of the path from APO to the cluster. This will NOT synchronize with APO as it will not invoke any of the processing reports. This is only to be used from user sdb - if called from root it can can leave information behind that sdb cannot remove.
This procedure is used to test a liveCache which is not yet attached to an APO system to verify the behavior and the end to end access path.

```
/<dependent path>/sap/lcinit
    -u                 username,passwd      (DBM user)
    -U                 [userkey]            (default DBMUSR)
    -uUTL              [username,passwd]    (UTILITY session)
    -uSQL              [username,passwd]    (SQL session)
    -uSRV              (Service session)
    -V                 (show version of server)
    -d   dbname        (set dbname)
    -R   dbroot        (set dbroot)
    -n   node          (name of servernode)
    -e   encryption    (encryption method)
                       (use only with -n)
                       (valid methods: 'SSL')
    -i   inputfile     (Default stdin)
    -ic  inputfile     (Default stdin)
    -o   outputfile    (Default stdout)
    -t   protocolfile  (writes subsequent protocol)
    -s                 (local mode, don't use with -n)
    -key               (wait for any key on exit)

  <DBMServer-Command>:
    Everything after the options will be sent to DBMServer.
    For more information about the DBMServer-Commands use
```

the DBMServer-Command help.

Example for SID= HLC DB administrator = control.

lcinit HLC stop -uDBM control,<password>
lcinit HLC restart -uDBM control,<password>

5.1.2 Simulating Start or Stop of liveCache from APO through lccluster

APO invokes LCINIT. This function checks for the existence of a script called lccluster and calls it with specific parameters. This script controls the cluster logic by maintaining the application status. This status modification can also be done by invoking lccluster which can also be invoked manually. Care must be taken that this is ONLY done as user sdb so as to maintain the correct authorizations for subsequent APO access. Actions on the liveCache should be done in collaboration with the APO administrator such that a restart is not attempted from that level during the maintenance activities.
This procedure is often used to test when a new liveCache is not yet attached to APO.

Advantage: Easy to do and quick recovery
Disadvantage: APO can restart the service if this is not coordinated!

NOTE: Script lccluster is found in the dependent path. This is an example below. Check for the valid dependent path in the current environment.

Set LC10.stop
As SDB user (su – sdb)

To simulate an APO shutdown completely, invoke the following command using the parameters as shown:

/sapdb/<SID>/db/sap/lccluster stopping req

Using dbmgui or dbmcli, stop the master database

/sapdb/<SID>/db/sap/lccluster stopping ok

Primary is now available for maintenance activities.

Set LC10.start
As SDB user (su – sdb)
To simulate the full APO startup, invoke the following command using the parameters as shown:

/sapdb/<SID>/sap/lccluster starting req

Using dbmgui or dbmcli, bring the master database online

/sapdb/<SID>/sap/lccluster starting ok

From this point, the cluster should recover and resynchronize the standby.

To ensure complete synchronization, stop and restart the liveCache from APO after a major maintenance step such as a restore of the database content.

To verify the success/failure of the lccluster script executions, look in /tmp/lccluster.log or /var/hacmp/lccluster.log.

5.1.3 Disabling the PowerHA cluster Responses for Maintenance

The OS level cluster administrator of a HotStandby must also respect the authority of the APO application. Doing normal cluster maintenance without coordination with the APO administrator can lead to very unhappy results. Therefore, before starting any cluster level administration the liveCache should first be shut down from APO by the APO administrator. APO has no concept of the cluster and there the hierarchy or control is purposely implemented to be driven from APO. This maintains the synchronization of the application. Starting and stopping liveCache from APO triggers activities in the APO system which supports the synchronization.

NOTE: Always stop the liveCache service from APO before embarking on cluster level maintenance activities.

5.1.3.1 Suspending PowerHA Application Monitoring

To suspend the cluster protection of the service, the following procedure can be used to suspend the monitors in PowerHA as an alternative to LC10 state setting.

This can be done in *smitty cl_admin → Resource Groups and Applications → Suspend/Resume Application Monitoring*.

- Advantage: PowerHA will entirely stop application monitoring.
- Disadvantage: Infrastructure failures will trigger failover.

5.1.3.2 PowerHA Unmanage Resource Groups feature

This is the most reliable approach to suspend cluster activity while keeping resources such as IP and disks online.

- Advantage: Infrastructure failures will not trigger a failover
- Disadvantage: Infrastructure failures will not trigger a failover!

Unmanage

Smitty cl_admin → PowerHA SystemMirror Services → Stop Cluster Services

```
                    Stop Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                        [Entry Fields]
* Stop now, on system restart or both   now
  Stop Cluster Services on these nodes  [is03d1,is04d1]
  BROADCAST cluster shutdown?           true
* Select an Action on Resource Groups   Unmanage Resource Groups
```

Figure 29: Stop Cluster node(s) with Unmanage Resource Groups option for maintenance.

After setting the rescource groups in unmanaged state the cluster manager will show following state:

 # lssrc -ls clstrmgrES
Current state: ST_STABLE
[…]

Using the "*Unmanage Resource Group*" option to stop the cluster services, the Cluster Manager puts the resource groups into the UNMANAGED state but continues to run under the covers thus leaving the RSCT services up and running under this condition.

One of the reasons that PowerHA does *not* stop the RSCT services from running when you stop cluster services is because *not* only HACMP but also the Enhanced Concurrent Mode (ECM) volume groups use RSCT services. Stopping RSCT services would vary off the ECM volume group and would affect the application that is using it.

There could be rare cases when you need to stop RSCT, for example, to perform an RSCT upgrade. If you need to upgrade RSCT, you can stop and restart it, using SMIT options under the HACMP Problem Determination Tools menu. For the steps needed to stop, restart and upgrade RSCT, see the PowerHA *Troubleshooting Guide* Redbook.

Reactivation

Start cluster services smitty cl_admin → PowerHA SystemMirror Services → Start Cluster Services

Start Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

	[Entry Fields]
* Start now, on system restart or both	now
Start Cluster Services on these nodes	[is03d1, is04d1]
* Manage Resource Groups	Automatically
BROADCAST message at startup?	false
Startup Cluster Information Daemon?	true
Ignore verification errors?	false
Automatically correct errors found during cluster start?	Interactively

Figure 30: PowerHA – reactivate cluster control for unmanaged resource groups

5.2 Administrative moves of instances for infrastructure maintenance

In case the AIX requires a reboot for maintenance the following steps need to be performed to manually trigger a takeover of the master in hot standby fashion. Anyhow, this should be done during a low load window and not during along Batch run.
Due to the relationships and dependencies between the resource groups, the master cannot be simply moved to the standby node.

The following steps show the required sequence:
Execute:

The online on different node policy will not allow to directly move the master. The test implies to stop the node running the master through C-SPOC stop cluster services with “move resources” option.

```
                    Stop Cluster Services

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                    [Entry Fields]
* Stop now, on system restart or both            now                 +
  Stop Cluster Services on these nodes           [is03d6]            +
  BROADCAST cluster shutdown?                    true                +
* Select an Action on Resource Groups            Move Resource Groups   +
```

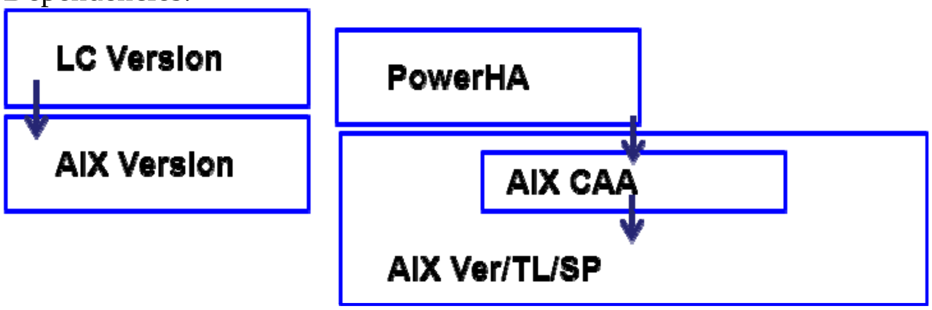
Expectation:

The rg_lc_master will relocate the service of the master to the active standby node and activate the standby as master. There will be no standby service thereafter and rg_lc_standby will become “Offline due to lack of node”. The kernel processes of standby will be reused and NOT restarted.

To reintegrate the standby start the just stopped cluster node.

5.3 Support matrix for Implementation Stack

Dependencies:



SAP and Platform dependecies
Typically, liveCache has a formal support matrix based on the AIX version. This is available in the SAP PAM (platform availability matrix). This is certified per each AIX version (AIX 5.3, AIX 6.1, AIX 7.1, etc).
SAP liveCache is supported already to version 7.9 on AIX 6.1 and from version 7.7 is supported to AIX 7.1. This matrix can be reviewed to verify the current platforms supported for each change in the liveCache version in the usual manner that SAP components stacks are normally verified. HotStandby is supported from liveCache version 7.5.

AIX and PowerHA dependencies
Support for the HotStandby liveCache begins with PowerHA 7.1.1 SP1. This version supports liveCache from LC version 7.7. There is little interdependencies between the liveCache version and PowerHA. liveCache is a mature implementation and the API which supports the HotStandby feature has been consistent since the introduction in version 7.5. Any questions or concerns in regards to a new version and HotStandby should be directed to SAP.

With the integration of PowerHA with the OS feature Cluster Aware AIX, PowerHA has a greater dependency on features which may be implemented in AIX CAA. In this case, PowerHA will set a prerequisite dependency for an upgrade if this is necessary. Any prerequisites will be documented in the release notes.

As for AIX, it is expected to provide back level compatibility for PowerHA such that an update of AIX can be done without consideration or dependencies on the PowerHA version. PowerHA (unless noted otherwise in the release notes associated with AIX).

6 Appendix

6.1 Related documents and sources of further information

Master Document – "Invincible Supply Chain" reference architecture for top-to-bottom HA
Also at this location you will find the latest version of this document and the installation guide for the HotStandby.
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

6.2 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

7 Trademarks and disclaimers

7.1 Copyrights and trademarks

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.

7.2 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.
COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs

ANY INFORMATION HEREIN IS PROVIDED "AS IS" WITHOUT WARRANTY OR INDEMNIFICATION OF ANY KIND BY IBM AND DO NOT ANY EXPRESS OR IMPLIED, WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS OR USAGE FOR PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.