



Pure Storage for SAP HANA® - Setup and Best Practices Guide

September 2015

© 2015 Pure Storage, Inc. All rights reserved.

Portions of this documentation are © 2015 SAP SE or an SAP affiliate company. All rights reserved. Screenshots used with the permission of SAP SE.

Pure Storage, the "P" Logo, and Pure1 are trademarks or registered trademarks of Pure Storage, Inc. in the U.S. and other countries. SAP and SAP Hana are registered trademarks of SAP in the U.S. and other countries. Oracle is a registered trademark of Oracle Corporation in the U.S. and other countries. The Pure Storage product described in this documentation is distributed under a license agreement and may be used only in accordance with the terms of the agreement. The license agreement restricts its use, copying, distribution, decompilation, and reverse engineering. No part of this documentation may be reproduced in any form by any means without prior written authorization from Pure Storage, Inc. and its licensors, if any.

THE DOCUMENTATION IS PROVIDED "AS IS" AND ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE DISCLAIMED, EXCEPT TO THE EXTENT THAT SUCH DISCLAIMERS ARE HELD TO BE LEGALLY INVALID. PURE STORAGE SHALL NOT BE LIABLE FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES IN CONNECTION WITH THE FURNISHING, PERFORMANCE, OR USE OF THIS DOCUMENTATION. THE INFORMATION CONTAINED IN THIS DOCUMENTATION IS SUBJECT TO CHANGE WITHOUT NOTICE.

Contributors: Krishna Satyavarapu
Pure Storage, Inc. 650 Castro Street, Mountain View, CA 94041
<http://www.purestorage.com>

Table of Contents

Executive Summary	4
Pure Storage Introduction	5
FlashArray//m Specifications.....	7
Purity Operating Environment	7
Pure1	8
Experience Evergreen Storage	8
Goals and Objectives	9
Audience	9
Solution Architecture	9
Pure Storage configuration:.....	10
Cluster Setup:.....	12
OCFS2 setup:.....	17
Configuration of udev rules:	21
Multipathing Configuration:	23
Additional Performance enhancements	31
HANA Configuration Files:.....	34
Conclusion	38

Executive Summary

The SAP HANA database takes advantage of main memory (RAM), the data processing capabilities of multicore processors to deliver better performance for analytical and transactional applications. It offers a multiengine query processing environment that allows it to support relational data (with both row and column oriented physical representations in a hybrid engine) as well as graph and text processing for semi structured and unstructured data management within the same system. The SAP HANA database is 100 percent compliant with atomicity, consistency, isolation, and durability (ACID) requirements.

For more information about SAP HANA, go to the SAP help portal: <http://help.sap.com/hana/>.

The performance of the underlying infrastructure, and in particular storage performance is the key issue that can hold back a HANA project. Traditional disk-based storage cannot meet the performance needs of HANA and is inherently too complicated to manage. The Pure Storage FlashArray//m can, not only deliver the required performance, but also the resiliency and management simplicity required by SAP HANA environments, at a cost this is often lower than that of disk based and hybrid arrays.

Specifically, enterprises running SAP HANA on the certified Pure Storage platform can benefit in the following ways:

- High performance storage with sub millisecond latency
- Highly efficient data reduction (inline compression and de-duplication)
- 5 9's high availability with no single points of failure
- Simplified management with rapid provisioning and replication capabilities

There are two ways to implementing SAP HANA:

- An "Appliance" option where the customers can purchase pre-configured systems for their HANA implementation. These configurations often tend to be less flexible as they come with fixed server, storage and network components.
- The SAP HANA Tailored Datacenter Integration (TDI) option, where customers can choose their server, storage and network components as long as they are all certified by SAP. The customers seem to prefer this option as they can leverage existing server, storage and networking infrastructure for their SAP HANA implementation.

The Pure Storage FlashArray//m series has been validated as a certified SAP HANA hardware platform as per SAP's "Certified Enterprise Storage" program. Customer can use Pure Storage's FlashArray//m series as a part of their SAP HANA TDI implementation.

Pure Storage Introduction

Who knew that moving to all-flash storage could help reduce the cost of IT? FlashArray//m makes server and workload investments more productive, while also lowering storage spend. With FlashArray//m, organizations can dramatically reduce the complexity of storage to make IT more agile and efficient, accelerating your journey to the cloud.



FlashArray//m's performance can also make your business smarter by unleashing the power of real-time analytics, driving customer loyalty, and creating new, innovative customer experiences that simply weren't possible with disk. All by Transforming Your Storage with FlashArray//m.

FlashArray//m enables you to transform your data center, cloud, or entire business with an affordable all-flash array capable of consolidating and accelerating all your key applications.

Mini Size—Reduce power, space and complexity by 90%

- 3U base chassis with 15-120+ TBs usable
- ~1kW of power
- 6 cables

Mighty Performance—Transform your datacenter, cloud, or entire business

- Up to 300,000 32K IOPS
- Up to 9 GB/s bandwidth
- <1ms average latency

Modular Scale—Scale FlashArray//m inside and outside of the chassis for generations

- Expandable to ~½ PB usable via expansion shelves
- Upgrade controllers and drives to expand performance and/or capacity

Meaningful Simplicity—Appliance-like deployment with worry-free operations

- Plug-and-go deployment that takes minutes, not days
- Non-disruptive upgrades and hot-swap everything
- Less parts = more reliability

The FlashArray//m expands upon the FlashArray's modular, stateless architecture, designed to enable expandability and upgradability for generations. The FlashArray//m leverages a chassis-based design with customizable modules, enabling both capacity and performance to be independently improved over time with advances in compute and flash, to meet your business' needs today and tomorrow.

The Pure Storage FlashArray is ideal for:

Accelerating Databases and Applications Speed transactions by 10x with consistent low latency, enable online data analytics across wide datasets, and mix production, analytics, dev/test, and backup workloads without fear.

Virtualizing and Consolidating Workloads Easily accommodate the most IO-hungry Tier 1 workloads, increase consolidation rates (thereby reducing servers), simplify VI administration, and accelerate common administrative tasks.

Delivering the Ultimate Virtual Desktop Experience Support demanding users with better performance than physical desktops, scale without disruption from pilot to >1000's of users, and experience all-flash performance for under \$100/desktop.

Protecting and Recovering Vital Data Assets Provide an always-on protection for business-critical data, maintain performance even under failure conditions, and recover instantly with FlashRecover.

Pure Storage FlashArray sets the benchmark for all-flash enterprise storage arrays. It delivers:

Consistent Performance FlashArray delivers consistent <1ms average latency. Performance is optimized for the real-world applications workloads that are dominated by I/O sizes of 32K or larger vs. 4K/8K hero performance benchmarks. Full performance is maintained even under failures/updates.

Less Cost than Disk Inline de-duplication and compression deliver 5 – 10x space savings across a broad set of I/O workloads including Databases, Virtual Machines and Virtual Desktop Infrastructure.

Mission-Critical Resiliency FlashArray delivers >99.999% proven availability, as measured across the Pure Storage installed base and does so with non-disruptive everything without performance impact.

Disaster Recovery Built-In FlashArray offers native, fully-integrated, data reduction-optimized backup and disaster recovery at no additional cost. Setup disaster recovery with policy-based automation within minutes. And, recover instantly from local, space-efficient snapshots or remote replicas.

Simplicity Built-In FlashArray offers game-changing management simplicity that makes storage installation, configuration, provisioning and migration a snap. No more managing performance, RAID, tiers or caching. Achieve optimal application performance without any tuning at any layer. Manage the FlashArray the way you like it: Web-based GUI, CLI, VMware vCenter, Rest API, or OpenStack.

//m20
15 – 120 TBs in 3U
150,000 32K IOPS



//m50
Up to 250 TBs in 7U
220,000 32K IOPS



//m70
Up to 400 TBs in 11U
300,000 32K IOPS



	//m20	//m50	//m70
Capacity	<ul style="list-style-type: none"> Up to 120+ TBs effective capacity* 5 – 40TBs raw capacity (base chassis) 	<ul style="list-style-type: none"> Up to 250+ TBs effective capacity* 30 – 88TBs raw capacity(w/shelves) 	<ul style="list-style-type: none"> Up to 400+ TBs effective capacity* 44 – 136TBs raw capacity (w/shelves)
Performance	<ul style="list-style-type: none"> Up to 150,000 32KIOPS** <1ms average latency Up to 5 GB/s bandwidth 	<ul style="list-style-type: none"> Up to 220,000 32KIOPS** <1ms average latency Up to 7 GB/s bandwidth 	<ul style="list-style-type: none"> Up to 300,000 32K IOPS** <1ms average latency Up to 9 GB/s bandwidth
Connectivity	<ul style="list-style-type: none"> 8 Gb/s Fibre Channel 10 Gb/s Ethernet iSCSI Management and Replication ports 	<ul style="list-style-type: none"> 16 Gb/s Fibre Channel 10 Gb/s Ethernet iSCSI Management and Replication ports 	<ul style="list-style-type: none"> 16 Gb/s Fibre Channel 10 Gb/s Ethernet iSCSI Management and Replication ports
Physical	<ul style="list-style-type: none"> 3U 742 Watts (nominal draw) 110 lbs. (49.9 kg) fully loaded 5.12" x 18.94" x 29.72" FlashArray//m chassis 	<ul style="list-style-type: none"> 3U – 7U 1007 - 1447 Watts (nominal draw) 110 lbs. (49.9 kg) fully loaded + 44 lbs. per expansion shelf 5.12" x 18.94" x 29.72" FlashArray//m chassis 	<ul style="list-style-type: none"> 5U – 11U 1439 – 2099 Watts (nominal draw) 110 lbs. (49.9 kg) fully loaded + 44 lbs. per expansion shelf 5.12" x 18.94" x 29.72" FlashArray//m chassis

FlashArray//m Specifications

* Effective capacity assumes HA, RAID, and metadata overhead, GB-to-GiB conversion, and includes the benefit of data reduction with always-on inline deduplication, compression, and pattern removal. Average data reduction is calculated at 5-to-1, below the global average of the FlashArray user base.

** Why does Pure Storage quote 32K, not 4K IOPS? The industry commonly markets 4K IOPS, but real-world environments are dominated by IO sizes of 32K or larger. FlashArray//m adapts automatically to 512B-32KB IO for superior performance, scalability, and data reduction.

Table 1. Pure Storage FlashArray//m Series.

Purity Operating Environment

Purity implements advanced data reduction, storage management and flash management features, and all features of Purity are included in the base cost of the FlashArray//m.

Storage Software Built for Flash—The FlashCare technology virtualizes the entire pool of flash within the FlashArray, and allows Purity to both extend the life and ensure the maximum performance of consumer-grade MLC flash.

Granular and Adaptive—Purity Core is based upon a 512-byte variable block size metadata layer. This fine-grain metadata enables all of Purity’s data and flash management services to operate at the highest efficiency.

Best Data Reduction Available—FlashReduce implements five forms of inline and post-process data reduction to offer the most complete data reduction in the industry. Data reduction operates at a 512-byte aligned variable block size, to enable effective reduction across a wide range of mixed workloads without tuning.

Highly Available and Resilient—FlashProtect implements high availability, dual-parity RAID-3D, non-disruptive upgrades, and encryption, all of which are designed to deliver full performance to the FlashArray during any failure or maintenance event.

Backup and Disaster Recovery Built In—FlashRecover combines space-saving snapshots, replication, and protection policies into an end-to-end data protection and recovery solution that protects data against loss locally and globally. All FlashProtect services are fully-integrated in the FlashArray and leverage the native data reduction capabilities.



Pure1 Manage—By combining local web-based management with cloud-based monitoring, Pure1 Manage allows you to manage your FlashArray wherever you are – with just a web browser.

Pure1 Connect—A rich set of APIs, plugin-is, application connectors, and automation toolkits enable you to connect FlashArray//m to all your data center and cloud monitoring, management, and orchestration tools.

Pure1 Support—FlashArray//m is constantly cloud-connected, enabling Pure Storage to deliver the most proactive support experience possible. Highly trained staff combined with big data analytics help resolve problems before they start.

Pure1 Collaborate—Extend your development and support experience online, leveraging the Pure1 Collaborate community to get peer-based support, and to share tips, tricks, and scripts.

Experience Evergreen Storage



Tired of the 3-5 year array replacement merry-go-round? The move to FlashArray//m can be your last data migration. Purchase and deploy storage once and once only – then expand capacity and performance incrementally in conjunction with your business needs and without downtime. Pure Storage’s vision for Evergreen Storage is delivered by a combination of the FlashArray’s stateless, modular architecture and the ForeverFlash business model, enabling you to extend the lifecycle of storage from 3-5 years to a decade or more.

Goals and Objectives

The purpose of this document is to describe a test plan and best practices for deploying a high performance SAP HANA environment with a Pure Storage FlashArray. This document will serve a guide for customers going through the process of configuring HANA clusters, underlying file systems and storage area networks for HANA.

Audience

This document is intended for Database Administrators (DBAs), Storage Administrators, System Administrators and anybody who wants to build a SAP HANA environment on a Pure Storage FlashArray with the best results. This is a technical paper that assumes familiarity with SAP HANA administration as well as basic Linux system and storage administration tasks as required for a typical database installation, such as setting up file systems, LUN configuration etc.

Solution Architecture

The SAP HANA certification infrastructure uses a 2+1 cluster configuration. Three Cisco UCS B440 M2 blades are used with SUSE 11 SP3 OS, including High Availability Option and OCFS to create the environment.

The Pure Storage FlashArray used in this configuration is the //m20. There are four 8GB Fibre channel ports from the array attached to the SAN Fabric. Each node has two Fibre channel ports for storage and four 10 GBPS ports for network.

All the servers boot from SAN with dedicated drive for their OS related partitions. All other LUNs are shared across the nodes. The /hana/shared partition is used for storing SAP binaries and is mounted in parallel simultaneously.

The HANA clustering controls the DATA and LOG LUNs that are needed by the master and the worker nodes. These are mounted in the appropriate node with the role of master or worker.

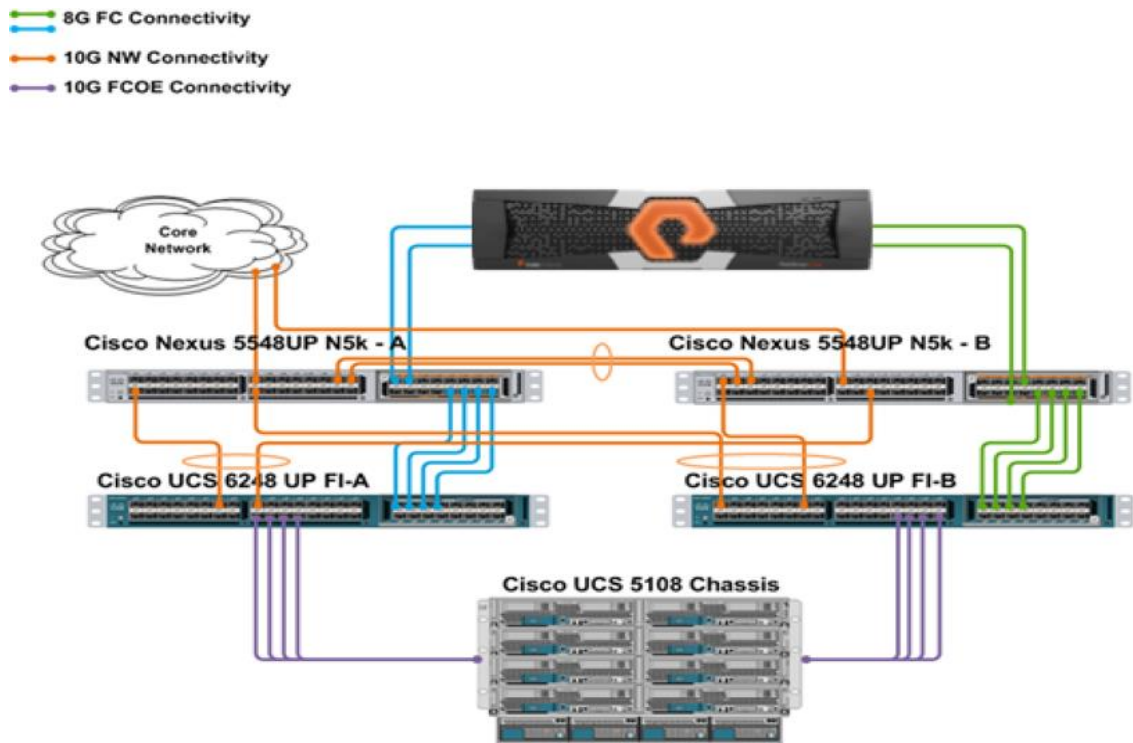


Figure 1: Pure Storage HANA Certification setup

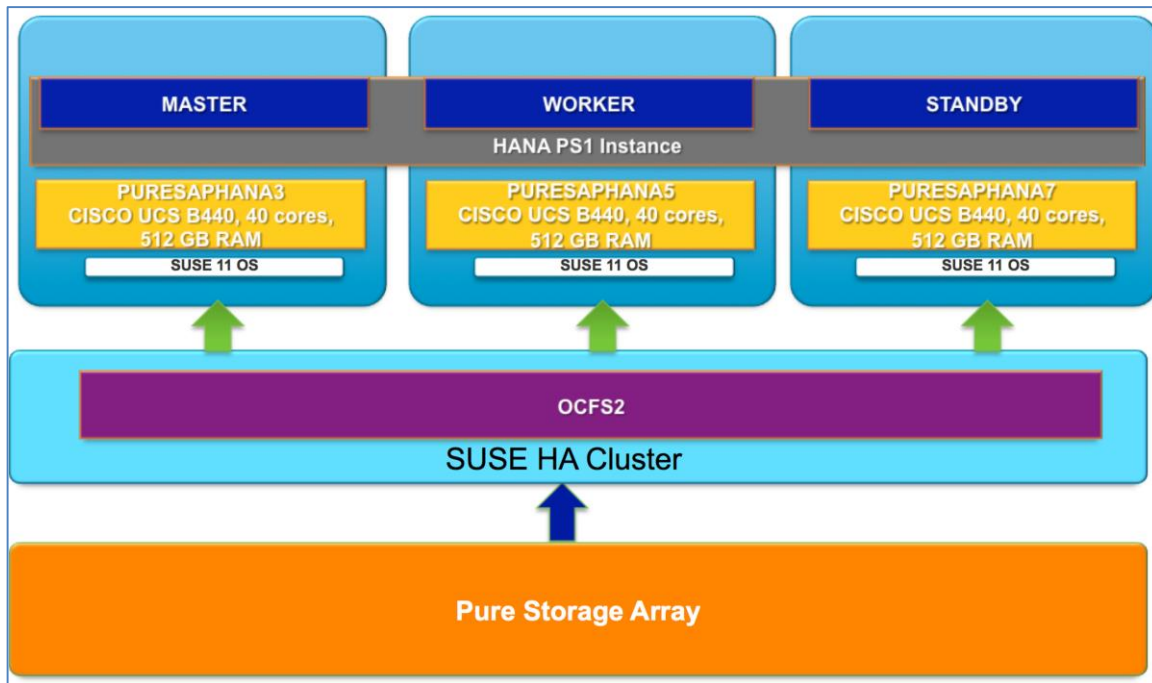


Figure 2: Logical architecture of the Pure Storage HANA setup

Pure Storage configuration:

The Pure Storage array FA405 is used in the configuration. The raw capacity of the array is 20 TB, but with Pure Storage de-duplication and compression capabilities the usable capacity is many fold.

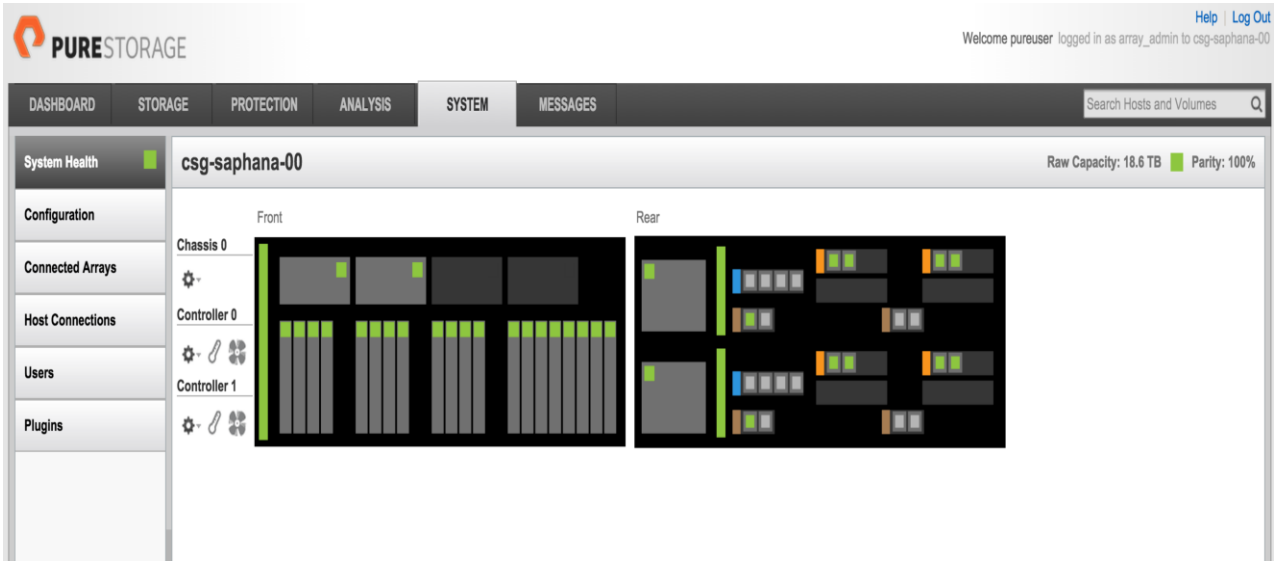


Figure 3: Pure Storage Array System Dashboard

The three Cisco UCS servers used for the HANA configuration are attached to the SAN fabric and zoned to the array. The three nodes are seen in the array as shown below.

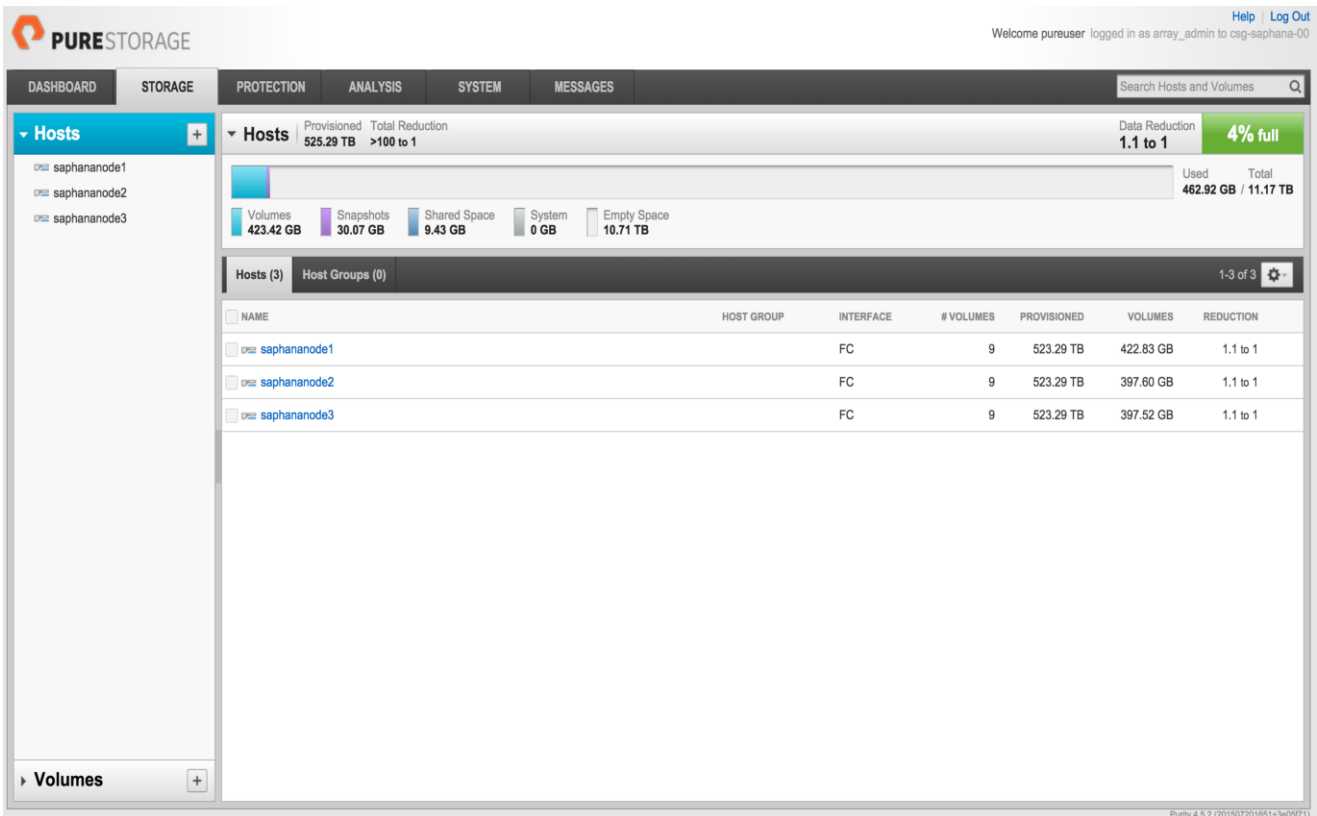


Figure 4: Hosts zoned to the array

LUN configuration:

Multiple LUNs are configured in the Pure storage array for use in the HANA setup. There are three boot LUNs as shown that are zoned to the individual hosts. All other LUNs are shared between the three nodes. The data and log LUNs are used by the HANA database. The HANA_SHARED_HOME LUN is used for the OCFS2 partition that is always mounted on all the nodes simultaneously and house the HANA binaries.

The quorum device is also a shared LUN (SBD_LUN) that is used for fencing hosts in the event of contention in the cluster

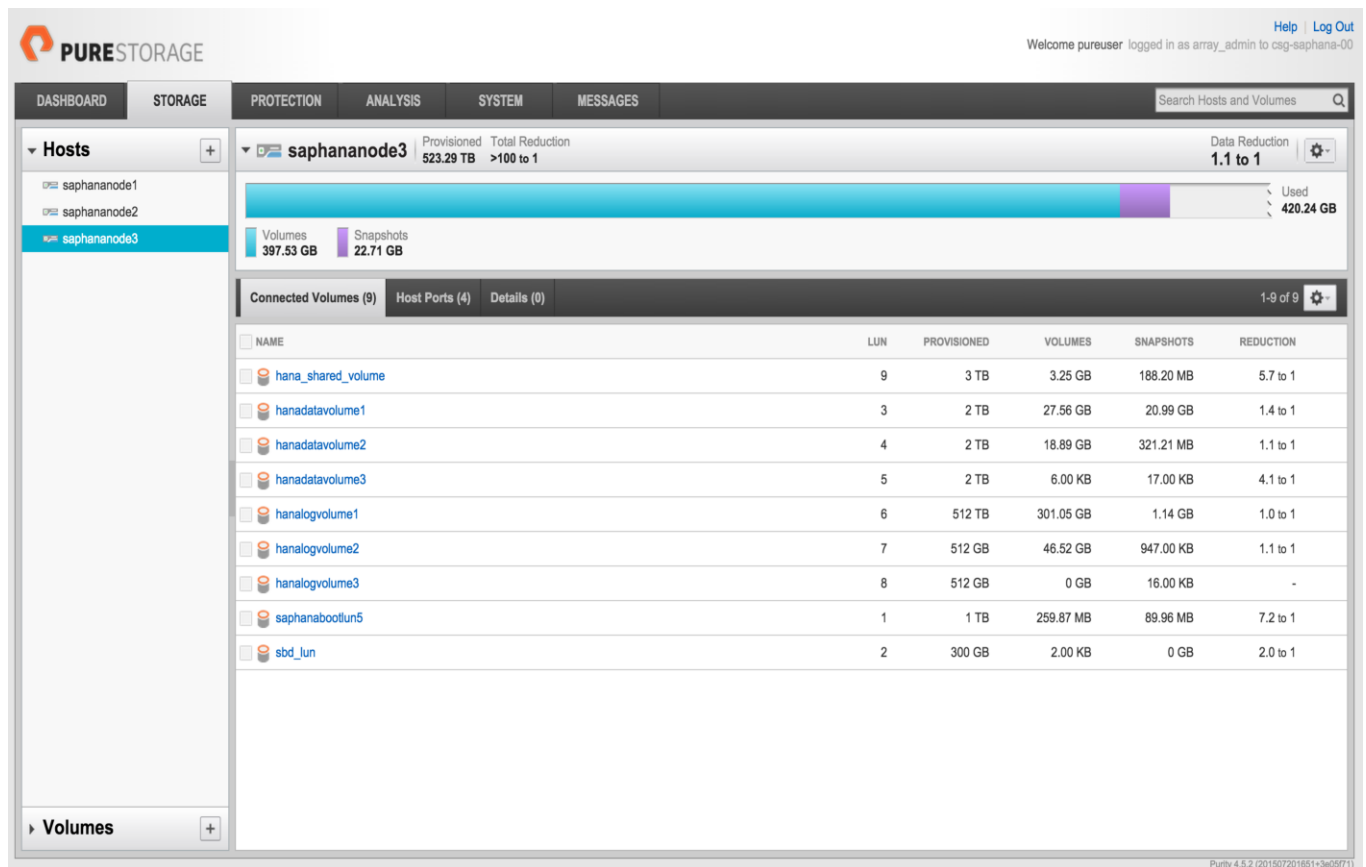


Figure 5: Pure Storage LUNS used in the setup

Cluster Setup:

- Enabled SSH for password less access on all three nodes.
- Installed SLES High availability components:
- Some of the high availability components are directly available by YasT.
- The other components are installed by downloading the two ISO images representing the HA SW.

Invoke YAST and install additional third party components. Choose the ISO images as the source of the software. Cluster components such as Corosync, Pacemaker, etc. are installed.

Install SLEHA Bootstrap and INIT and initialize cluster:

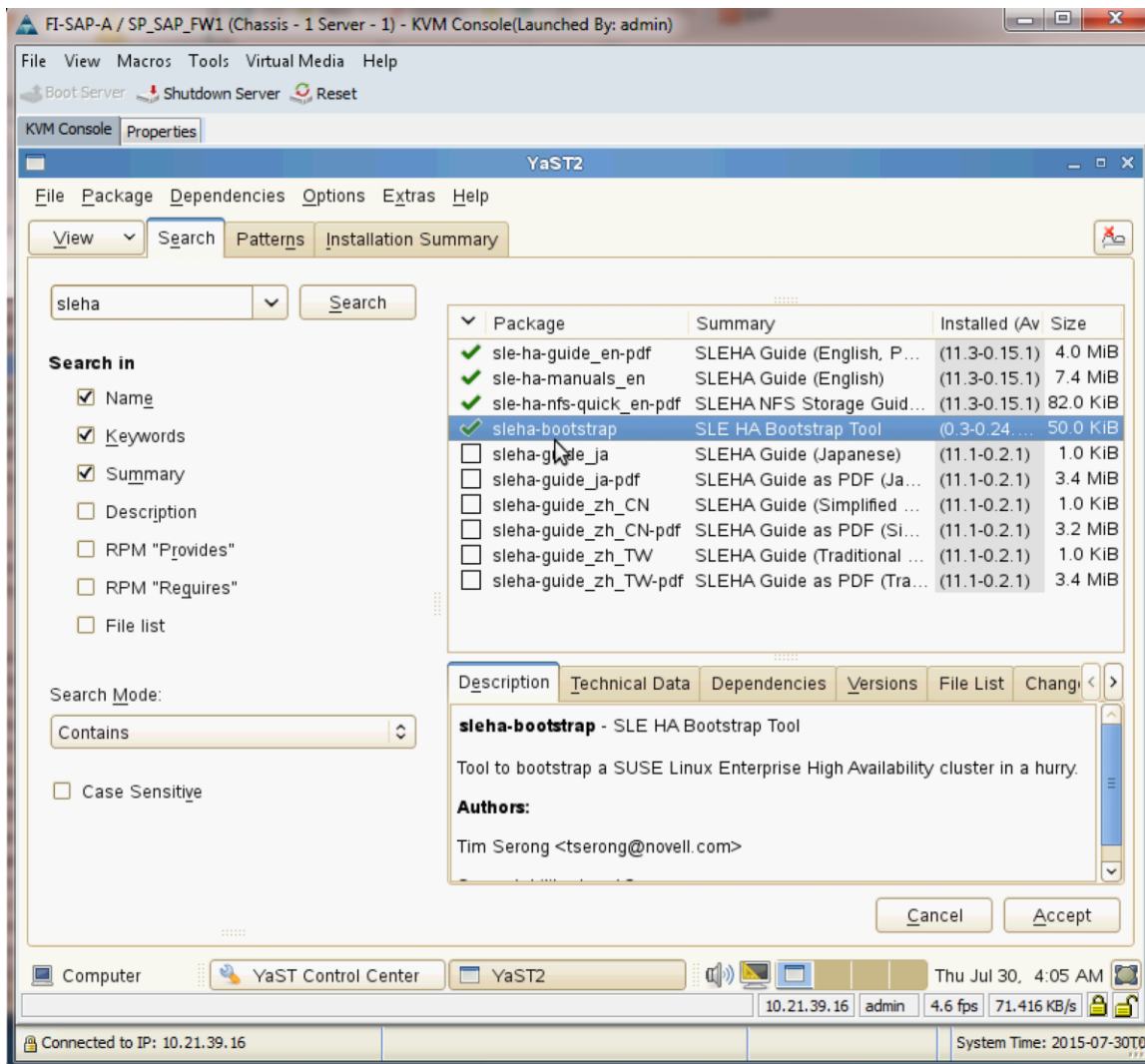


Figure 6: SLES HA install

Create sbd on a separate lun using the create sbd on a shared lun.

```
saphananode1:/dev/mapper # sbd -d /dev/mapper/3624a93707d4c758cacd1f75000011057 create
Initializing device /dev/mapper/3624a93707d4c758cacd1f75000011057
Creating version 2.1 header on device 3 (uuid: 0d05b48c-9bd4-45ad-a2b8-27d3079838b7)
Initializing 255 slots on device 3
Device /dev/mapper/3624a93707d4c758cacd1f75000011057 is initialized.
saphananode1:/dev/mapper #
```

Then Run the sleha-init command and choose communication modes and other parameters in the first node. The stonith SBD device is automatically recognized and leveraged as the quorum device for the cluster.

```
saphananode1:/dev/mapper # sleha-init
Enabling sshd service
Generating ssh key
Configuring csync2
Generating csync2 shared key (this may take a while)...done
Enabling csync2 service
Enabling xinetd service
csync2 checking files

Configure Corosync:
This will configure the cluster messaging layer. You will need
to specify a network address over which to communicate (default
is eth0's network, but you can use the network address of any
active interface), a multicast address and multicast port.

Network address to bind to (e.g.: 192.168.1.0) [10.21.32.0] 10.21.39.48
Multicast address (e.g.: 239.x.x.x) [239.88.219.67]
Multicast port [5405]

Configure SBD:
If you have shared storage, for example a SAN or iSCSI target,
you can use it to avoid split-brain scenarios by configuring SBD.
This requires a 1 MB partition, accessible to all nodes in the
cluster. The device path must be persistent and consistent
across all nodes in the cluster, so /dev/disk/by-id/* devices
are a good choice. Note that all data on the partition you
specify here will be destroyed.

Do you wish to use SBD? [y/N] y
Path to storage device (e.g. /dev/disk/by-id/...) [] /dev/mapper/3624a93707d4c758cacd1f75000011057
All data on /dev/mapper/3624a93707d4c758cacd1f75000011057 will be destroyed
Are you sure you wish to use this device [y/N] y
Initializing SBD.....done
Enabling hawk service
HA Web Konsole is now running, to see cluster status go to:
https://10.21.39.45:7630/
Log in with username 'hacluster', password 'linux'
WARNING: You should change the hacluster password to something more secure!
Enabling openais service
Waiting for cluster.....done
Loading initial configuration
Done (log saved to /var/log/sleha-bootstrap.log)
```

Figure 7: SUSE HA Init

- Run sleha-join in the other two nodes.
- SLES HA by default leverages Multicast, which might or not be supported on the existing infrastructure. Reconfigure the communication channel to use Unicast instead to ensure that the nodes are able to communicate with each other.
- Use the “yast2 cluster” command to invoke the configuration utility.
- Configure the setting using yast2 cluster command. Enter UDPU and multiple member addresses.

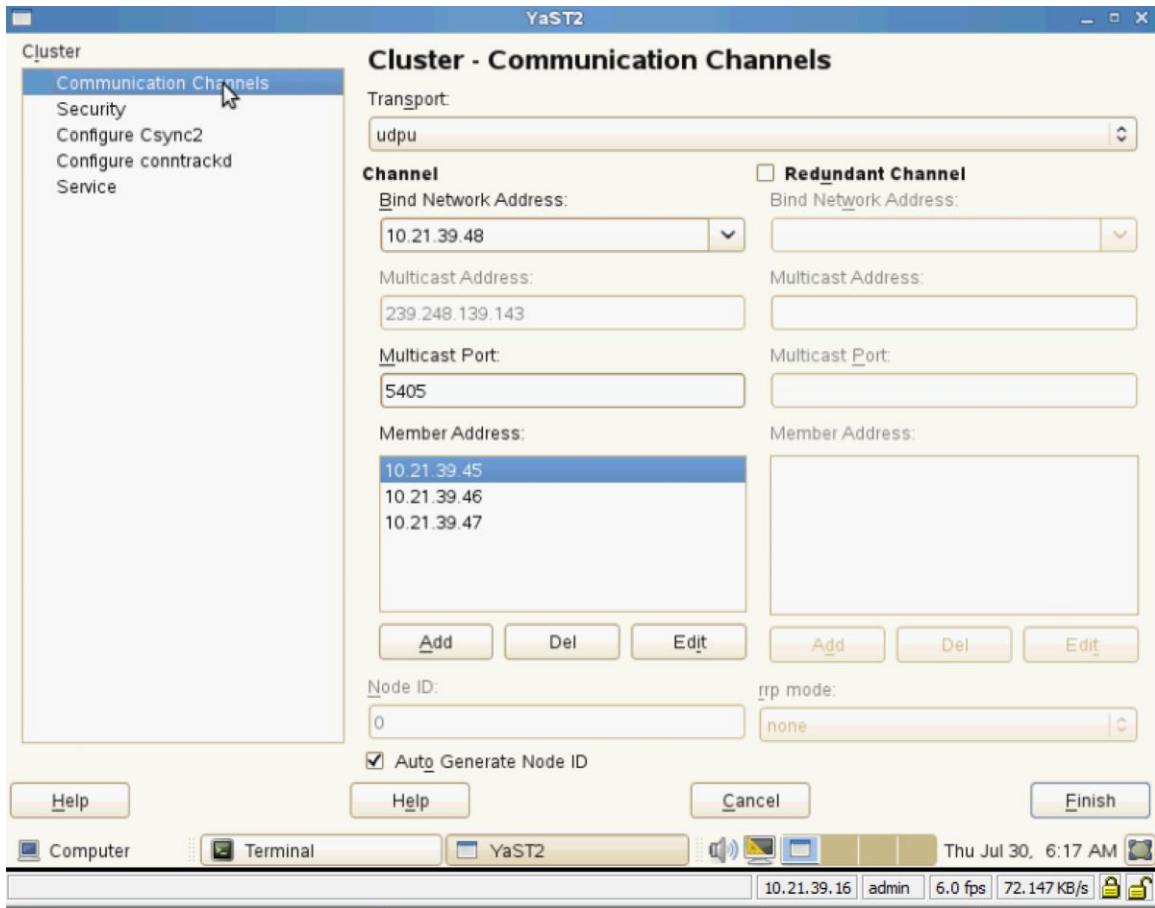


Figure 8: Cluster communications configuration

- Ensure that the corosync.conf file is copied to all nodes in the cluster from the first node. Restart Openais to apply all changes.

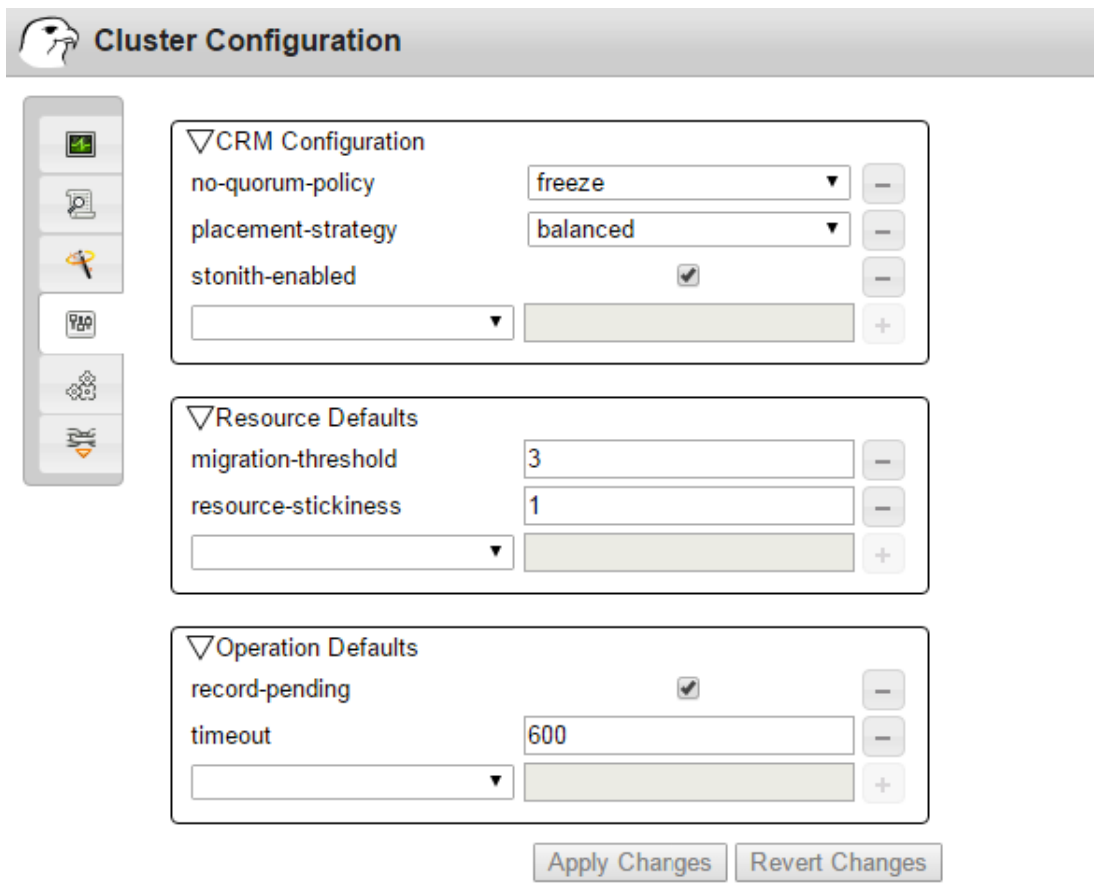


Figure 9: Enable Quorum on Stonith

- Update cluster configuration using the HAWK portal to enable quorum using the Stonith SBD device.

Cluster Status after Setup:

The command line utility `crm` can be used for verifying that the cluster has been setup correctly. After the cluster is setup run this command to verify cluster operation and components.

```
saphananode1:~ # crm status
```

```
Last updated: Wed Sep 2 22:07:12 2015
```

```
Last change: Wed Sep 2 22:00:19 2015 by hacluster via cibadmin on saphananode1
```

```
Stack: classic openais (with plugin)
```

```
Current DC: saphananode3 - partition with quorum
```

```
Version: 1.1.9-2db99f1
```


3 Nodes configured, 3 expected votes

10 Resources configured.

Online: [saphananode1 saphananode2 saphananode3]

stonith-sbd (stonith:external/sbd): Started saphananode2

Clone Set: base-clone [base-group]

Started: [saphananode1 saphananode2 saphananode3]

OCFS2 setup:

Install all OCFS2 components except the developer tools using Yast on all the servers. Reboot the servers to enable the kernel module.

OCFS Steps:

- Install OCFS2 packages (installed with the HA Extension by default)
- Configure a DLM clone and O2CB clone cluster resources
- Format SAN volume with an OCFS2 file system
- Configure an OCFS2 volume clone cluster resource

OCFS2 Solution Components:

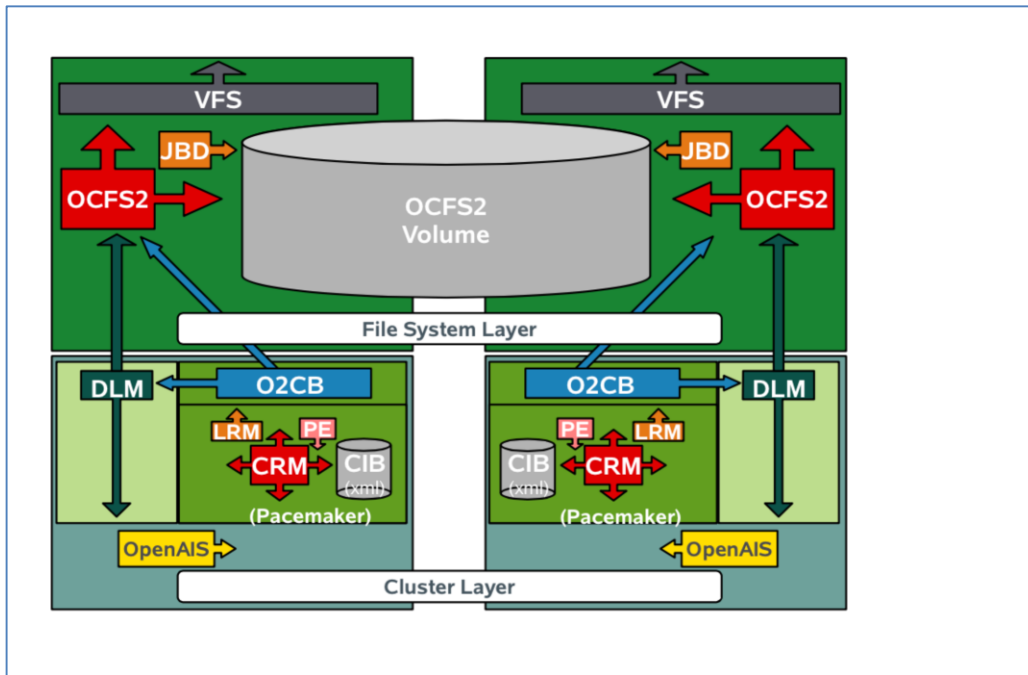


Figure 10: OCFS2 Solution Components

Install OCFS2 Packages:

Launch YaST and search for OCFS2. Install all packages except the developer package on all of the servers

Perform the following steps to install OCFS components.

```
yast2 sw_single
```

Configure DLM and O2CB cluster resources:

The command “crm configure” should be used to configure these resources. The steps that were followed:

```
saphanode1:~ # crm configure
```

```
crm(live)configure# primitive dlm ocf:pacemaker:controld op monitor interval="60" timeout="60"
crm(live)configure# primitive o2cb ocf:ocfs2:o2cb op monitor interval="60" timeout="60"
crm(live)configure# group base-group dlm o2cb
```

```
crm(live)configure# primitive dlm ocf:pacemaker:controld op monitor interval="60" timeout="60"
crm(live)configure# primitive o2cb ocf:ocfs2:o2cb op monitor interval="60" timeout="60"
```

Format SAN volume with an OCFS2 file system:

- The ‘parted’ tool is used for partitioning shared HANA home volume.

```
saphananode1:~# parted /dev/mapper/3624a93707d4c758cacd1f7500001105e
```

```
GNU Parted 2.3
```

```
Using /dev/dm-0
```

```
Welcome to GNU Parted! Type 'help' to view a list of commands.
```

```
(parted) mklabel gpt
```

```
(parted) print
```

```
Model: Linux device-mapper (multipath) (dm)
```

```
Disk /dev/dm-0: 3299GB
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt
```

```
Number Start End Size File system Name Flags
```

```
(parted) mkpart primary 0GB 3299GB
```

```
(parted) print
```

```
Model: Linux device-mapper (multipath) (dm)
```

```
Disk /dev/dm-0: 3299GB
```

```
Sector size (logical/physical): 512B/512B
```

```
Partition Table: gpt
```

```
Number Start End Size File system Name Flags
```

```
1 4194kB 3299GB 3299GB primary
```

```
mkfs.ocfs2 -N /dev/mapper/3624a93707d4c758cacd1f7500001105e_part1
```

```
mkfs.ocfs2 1.8.2
```

```
Cluster stack: pcmk
```

```
Cluster name: pacemaker
```

Stack Flags: 0x0

NOTE: Feature extended slot map may be enabled

Label:

Features: sparse extended-slotmap backup-super unwritten inline-data strict-journal-super metaecc xattr indexed-dirs
refcount discontig-bg

Block size: 4096 (12 bits)

Cluster size: 4096 (12 bits)

Volume size: 3298526494720 (805304320 clusters) (805304320 blocks)

Cluster groups: 24967 (tail covers 1024 clusters, rest cover 32256 clusters)

Extent allocator size: 1103101952 (263 groups)

Journal size: 268435456

Node slots: 3

Creating bitmaps: done

Initializing superblock: done

Writing system files: done

Writing superblock: done

Writing backup superblock: 6 block(s)

Formatting Journals: done

Growing extent allocator: done

Formatting slot map: done

Formatting quota files: done

Writing lost+found: done

mkfs.ocfs2 successful

Mount manually and confirm:

- Create mountpoint directory individually in each of the nodes
`mkdir /hana/shared`
- Mount on one of the nodes to confirm.

`saphananode1:~# mount /dev/mapper/3624a93707d4c758cacd1f7500001105e_part1 /hana/shared`

Unmount filesystem after check:

```
umount /hana/shared
```

Configure an OCFS2 volume clone cluster resource:

The command `crm configure` or the HAWK interface can be used to configure the volume cluster resource.

```
crm configure
```

```
primitive ocfs2-1 ocf:heartbeat:Filesystem params
device="/dev/mapper/3624a93707d4c758cacd1f7500001105e_part1" directory="/hana/shared" fstype
="ocfs2" options="acl" op monitor interval="20" timeout="40"
```

On committing these changes in CRM, the `/hana/shared` filesystem is mounted on all nodes in the cluster concurrently.

Hawk Interface view of ocfs2-1 configuration:

Resource ID	Class	Provider	Type
ocfs2-1	ocf	heartbeat	Filesystem
<input type="checkbox"/> Use Template			
Parameters			
device	/dev/mapper/3624a93707d4c7		-
directory	/hana/shared		-
fstype	ocfs2		-
options	acl		-
			+
Operations			
monitor	timeout: 40 interval: 20		-
			+
Meta Attributes			

Figure 11: Setting up shared filesystem

Configuration of udev rules:

The device manager of the kernel needs to be configured as shown below. Most important parameters to be changed are nr_requests and scheduler. Please set parameters has shown below for Pure Storage in the udev/rules.d as shown below:

```
saphananode1:/etc/udev/rules.d # ls
40-alsa.rules  51-packagekit-firmware.rules  70-kpartx.rules  77-network.rules  85-usb_autosuspend_devices.rules  99-ocsc_lite.rules
51-dlm.rules   65-drbd.rules                 70-persistent-cd.rules  81-mount.rules   85-usb_elotouch_wakeup.rules      99-pure-storage.rules
51-ocfs2.rules 70-kdump.rules                70-persistent-net.rules 81-mptctl.rules  99-iwlwifi-led.rules
```

Figure 12: Pure Storage udev Rules

99-pure-storage.rules

Recommended settings for Pure Storage FlashArray.

Use noop scheduler for high-performance solid-state storage

```
ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/scheduler}="noop"
```

Reduce CPU overhead due to entropy collection

```
ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/add_random}="o"
```

Spread CPU load by redirecting completions to originating CPU

```
ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/rq_affinity}="2"
```

```
ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/nr_requests}="1024"
```

```
ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/nomerges}="1"
```

```
#ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/max_sectors_kb}="4096"
```

```
ACTION=="add|change", KERNEL=="sd*[:!o-9]", SUBSYSTEM=="block", ENV{ID_VENDOR}=="PURE",
ATTR{queue/rotational}="o"
```

```
ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?*\"", ATTR{queue/scheduler}="noop"
```

```
ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?*\"", ATTR{queue/rotational}="o"
```

```
ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?*\"", ATTR{queue/nr_requests}="4096"
```

```
ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?*\"", ATTR{queue/rq_affinity}="2"
```

```

ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?* ", ATTR{queue/nomerges}="1"
#ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?* ", ATTR{queue/max_sectors_kb}="4096"
ACTION=="add|change", ENV{DM_UUID}=="mpath-3624a937?* ", ATTR{queue/add_random}="0"
ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/scheduler}="noop"
ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/rotational}="0"
ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/nr_requests}="4096"
ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/rq_affinity}="2"
ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/nomerges}="1"
#ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/max_sectors_kb}="4096"
ACTION=="add|change", KERNEL=="dm-*", ATTR{queue/add_random}="0"
# SCSI timeout in line with Customer Best Practices
ACTION=="add", SUBSYSTEM=="scsi", ATTRS{model}=="FlashArray ", RUN+="/bin/sh -c 'echo 60 > /sys$DEVPATH/timeout'"

```

Multipathing Configuration:

Multipathing needs to be setup to do round robin for all PURE LUNs by setting it up in /etc/multipath.conf. Install sysstat package with Yast2 to get iostat in the servers. It is very important to set the path-selector in multipath configuration to **queue-length**.

The file contents of multipath.conf are shown here:

```

saphananode1:~ # cat /etc/multipath.conf

blacklist_exceptions {

device {

vendor "PURE"

}

}

## Use user friendly names, instead of using WWIDs as names.

defaults {

```

```

user_friendly_names no
max_fds max
flush_on_last_del yes
queue_without_daemon no
}
blacklist {
devnode "^hd[a-z]"
devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
#devnode "^(ram|raw|loop|fd|md|sr|scd|st)[0-9]*"
}
blacklist {
device {
vendor "HP"
}
}
devices {
device {
vendor "PURE"
path_selector "'queue-length 0'"
path_grouping_policy multibus
rr_min_io 1
path_checker tur
}
}
pu

```


After making these changes reboot the server and the output will look like the following:

```
saphananode1:~ # multipath -ll
3624a93707d4c758cacd1f7500001105e dm-10 PURE ,FlashArray
size=3.0T features='o' hwhandler='o' wp=rw
`+- policy='queue-length 0' prio=1 status=active
|- 1:0:0:9 sdan 66:112 active ready running
|- 1:0:2:9 sdbx 68:176 active ready running
|- 1:0:3:9 sddb 70:144 active ready running
|- 1:0:7:9 sdej 128:176 active ready running
|- 2:0:0:9 sdr 65:16 active ready running
|- 2:0:1:9 sdbc 67:96 active ready running
|- 2:0:3:9 sdcr 69:240 active ready running
|- 2:0:4:9 sddz 128:16 active ready running
|- 3:0:0:9 sdaw 67:0 active ready running
|- 3:0:1:9 sdch 69:80 active ready running
|- 3:0:2:9 sddo 71:96 active ready running
|- 3:0:7:9 sden 128:240 active ready running
|- 4:0:0:9 sdaf 65:240 active ready running
|- 4:0:1:9 sdbm 68:0 active ready running
|- 4:0:2:9 sdcs 70:0 active ready running
`- 4:0:4:9 sdek 128:192 active ready running
3624a93707d4c758cacd1f7500001105d dm-9 PURE ,FlashArray
size=512G features='o' hwhandler='o' wp=rw
`+- policy='round-robin 0' prio=1 status=active
|- 1:0:0:8 sdai 66:32 active ready running
```

```

|- 1:0:2:8 sdbu 68:128 active ready running
|- 1:0:3:8 sdcw 70:64 active ready running
|- 1:0:7:8 sdef 128:112 active ready running
|- 2:0:0:8 sdp 8:240 active ready running
|- 2:0:1:8 sday 67:32 active ready running
|- 2:0:3:8 sdck 69:128 active ready running
|- 2:0:4:8 sddw 71:224 active ready running
|- 3:0:0:8 sdat 66:208 active ready running
|- 3:0:1:8 sdcd 69:16 active ready running
|- 3:0:2:8 sddm 71:64 active ready running
|- 3:0:7:8 sdem 128:224 active ready running
|- 4:0:0:8 sdy 65:128 active ready running
|- 4:0:1:8 sdbi 67:192 active ready running
|- 4:0:2:8 sdco 69:192 active ready running
`- 4:0:4:8 sdeh 128:144 active ready running
3624a93707d4c758cacd1f7500001105c dm-8 PURE ,FlashArray
size=512G features='o' hwhandler='o' wp=rw
`+- policy='round-robin o' prio=1 status=active
|- 1:0:0:7 sdae 65:224 active ready running
|- 1:0:2:7 sdbo 68:32 active ready running
|- 1:0:3:7 sdcv 70:48 active ready running
|- 1:0:7:7 sdec 128:64 active ready running
|- 2:0:0:7 sdl 8:176 active ready running
|- 2:0:1:7 sdav 66:240 active ready running
|- 2:0:3:7 sdcg 69:64 active ready running
|- 2:0:4:7 sddt 71:176 active ready running

```

```
| 3:0:0:7 sdap 66:144 active ready running
| 3:0:1:7 sdbz 68:208 active ready running
| 3:0:2:7 sddg 70:224 active ready running
| 3:0:7:7 sdel 128:208 active ready running
| 4:0:0:7 sdu 65:64 active ready running
| 4:0:1:7 sdbd 67:112 active ready running
| 4:0:2:7 sdcn 69:160 active ready running
` 4:0:4:7 sdee 128:96 active ready running
3624a93707d4c758cacd1f7500001105b dm-7 PURE ,FlashArray
size=512T features='o' hwhandler='o' wp=rw
`+- policy='round-robin 0' prio=1 status=active
| 1:0:0:6 sdaa 65:160 active ready running
| 1:0:2:6 sdbk 67:224 active ready running
| 1:0:3:6 sdct 70:16 active ready running
| 1:0:7:6 sdea 128:32 active ready running
| 2:0:0:6 sdk 8:160 active ready running
| 2:0:1:6 sdar 66:176 active ready running
| 2:0:3:6 sdcc 69:0 active ready running
| 2:0:4:6 sddp 71:112 active ready running
| 3:0:0:6 sdal 66:80 active ready running
| 3:0:1:6 sdbv 68:144 active ready running
| 3:0:2:6 sdde 70:192 active ready running
| 3:0:7:6 sdei 128:160 active ready running
| 4:0:0:6 sdq 65:0 active ready running
| 4:0:1:6 sdba 67:64 active ready running
| 4:0:2:6 sdcj 69:112 active ready running
```

```

`- 4:0:4:6 sdeb 128:48 active ready running
3624a93707d4c758cacd1f7500001105a dm-6 PURE ,FlashArray
size=2.0T features='o' hwhandler='o' wp=rw
`+- policy='round-robin o' prio=1 status=active
|- 1:0:0:5 sdv 65:80 active ready running
|- 1:0:2:5 sdbh 67:176 active ready running
|- 1:0:3:5 sdcv 69:208 active ready running
|- 1:0:7:5 sddu 71:192 active ready running
|- 2:0:0:5 sdh 8:112 active ready running
|- 2:0:1:5 sdam 66:96 active ready running
|- 2:0:3:5 sdbv 68:192 active ready running
|- 2:0:4:5 sddl 71:48 active ready running
|- 3:0:0:5 sdah 66:16 active ready running
|- 3:0:1:5 sdbv 68:80 active ready running
|- 3:0:2:5 sdda 70:128 active ready running
|- 3:0:7:5 sdeg 128:128 active ready running
|- 4:0:0:5 sdn 8:208 active ready running
|- 4:0:1:5 sdax 67:16 active ready running
|- 4:0:2:5 sdcf 69:48 active ready running
`- 4:0:4:5 sddx 71:240 active ready running
3624a93707d4c758cacd1f7500001103c dm-o PURE ,FlashArray
size=1.0T features='o' hwhandler='o' wp=rw
`+- policy='round-robin o' prio=1 status=active
|- 1:0:0:1 sdi 8:128 active ready running
|- 1:0:2:1 sdaq 66:160 active ready running
|- 1:0:3:1 sdcv 68:240 active ready running

```

```

|- 1:0:7:1 sddd 70:176 active ready running
|- 2:0:0:1 sda 8:0 active ready running
|- 2:0:1:1 sdw 65:96 active ready running
|- 2:0:3:1 sdbg 67:160 active ready running
|- 2:0:4:1 sdcy 70:96 active ready running
|- 3:0:0:1 sdt 65:48 active ready running
|- 3:0:1:1 sdbb 67:80 active ready running
|- 3:0:2:1 sdcn 69:176 active ready running
|- 3:0:7:1 sddr 71:144 active ready running
|- 4:0:0:1 sdc 8:32 active ready running
|- 4:0:1:1 sdag 66:0 active ready running
|- 4:0:2:1 sdbp 68:48 active ready running
`- 4:0:4:1 sddi 71:0 active ready running
3624a93707d4c758cacd1f75000011059 dm-5 PURE ,FlashArray
size=2.0T features='o' hwhandler='o' wp=rw
`+- policy='round-robin o' prio=1 status=active
|- 1:0:0:4 sds 65:32 active ready running
|- 1:0:2:4 sdbe 67:128 active ready running
|- 1:0:3:4 sdcl 69:144 active ready running
|- 1:0:7:4 sddq 71:128 active ready running
|- 2:0:0:4 sdf 8:80 active ready running
|- 2:0:1:4 sdaj 66:48 active ready running
|- 2:0:3:4 sdbt 68:112 active ready running
|- 2:0:4:4 sddh 70:240 active ready running
|- 3:0:0:4 sdad 65:208 active ready running
|- 3:0:1:4 sdbn 68:16 active ready running

```

```

|- 3:0:2:4 sdcx 70:80 active ready running
|- 3:0:7:4 sded 128:80 active ready running
|- 4:0:0:4 sdj 8:144 active ready running
|- 4:0:1:4 sdas 66:192 active ready running
|- 4:0:2:4 sdca 68:224 active ready running
`- 4:0:4:4 sdds 71:160 active ready running
3624a93707d4c758cacd1f75000011058 dm-4 PURE ,FlashArray
size=2.0T features='o' hwhandler='o' wp=rw
`+- policy='round-robin o' prio=1 status=active
|- 1:0:0:3 sdo 8:224 active ready running
|- 1:0:2:3 sdaz 67:48 active ready running
|- 1:0:3:3 sdcj 69:96 active ready running
|- 1:0:7:3 sddk 71:32 active ready running
|- 2:0:0:3 sdd 8:48 active ready running
|- 2:0:1:3 sdac 65:192 active ready running
|- 2:0:3:3 sdbq 68:64 active ready running
|- 2:0:4:3 sddc 70:160 active ready running
|- 3:0:0:3 sdz 65:144 active ready running
|- 3:0:1:3 sdbj 67:208 active ready running
|- 3:0:2:3 sdcu 70:32 active ready running
|- 3:0:7:3 sddy 128:0 active ready running
|- 4:0:0:3 sdg 8:96 active ready running
|- 4:0:1:3 sdao 66:128 active ready running
|- 4:0:2:3 sdbw 68:160 active ready running
`- 4:0:4:3 sddn 71:80 active ready running
3624a93707d4c758cacd1f75000011057 dm-1 PURE ,FlashArray

```

```
size=300G features='o' hwhandler='o' wp=rw
`+- policy='round-robin o' prio=1 status=active
|- 1:0:0:2 sdm 8:192 active ready running
|- 1:0:2:2 sdau 66:224 active ready running
|- 1:0:3:2 sdce 69:32 active ready running
|- 1:0:7:2 sddf 70:208 active ready running
|- 2:0:0:2 sdb 8:16 active ready running
|- 2:0:1:2 sdab 65:176 active ready running
|- 2:0:3:2 sdbl 67:240 active ready running
|- 2:0:4:2 sdcz 70:112 active ready running
|- 3:0:0:2 sdx 65:112 active ready running
|- 3:0:1:2 sdbf 67:144 active ready running
|- 3:0:2:2 sdcq 69:224 active ready running
|- 3:0:7:2 sddv 71:208 active ready running
|- 4:0:0:2 sde 8:64 active ready running
|- 4:0:1:2 sdak 66:64 active ready running
|- 4:0:2:2 sdbs 68:96 active ready running
`- 4:0:4:2 sddj 71:16 active ready running
```

Additional Performance enhancements

Boot.local:

```
saphananode1:~ # cat /etc/init.d/boot.local
#!/bin/sh
#
# Copyright (c) 2002 SuSE Linux AG Nuernberg, Germany. All rights reserved.
#
```

```
# Author: Werner Fink <werner@suse.de>, 1996
#   Burchard Steinbild, 1996
#
# /etc/init.d/boot.local
#
# script with local commands to be executed from init on system startup
#
# Here you should add things, that should happen directly after booting
# before we're going to the first run level.
#
modprobe softdog
transparent_hugepage=never
```

Boot file with kernel settings: /boot/grub/menu.lst

```
# Modified by YaST2. Last modification on Wed Jul 28 19:44:21 UTC 2015
default 4
timeout 8
##YaST - generic_mbr
gfxmenu (hdo,o)/message
##YaST - activate

###Don't change this comment - YaST2 identifier: Original name: linux###
title Trace -- SLES for SAP Applications - 3.0.101-0.46
    root (hdo,o)
```

```
kernel /vmlinuz-3.0.101-0.46-trace root=/dev/system/root_lv instmode=cd resume=/dev/system/swap_lv splash=silent
showopts vga=0x314 intel_idle.max_cstate=0 processor.max_cstate=0 transparent_hugepage=never
```



```
initrd /initrd-3.0.101-0.46-trace
```

```
###Don't change this comment - YaST2 identifier: Original name: linux###
```

```
title SLES for SAP Applications - 3.0.101-0.46 (bigsm)
```

```
root (hdo,o)
```

```
kernel /vmlinuz-3.0.101-0.46-bigsm root=/dev/system/root_lv instmode=cd resume=/dev/system/swap_lv splash=silent showopts vga=0x314 intel_idle.max_cstate=0 processor.max_cstate=0 transparent_hugepage=never
```

```
initrd /initrd-3.0.101-0.46-bigsm
```

```
###Don't change this comment - YaST2 identifier: Original name: failsafe###
```

```
title Failsafe -- SLES for SAP Applications - 3.0.101-0.46
```

```
root (hdo,o)
```

```
kernel /vmlinuz-3.0.101-0.46-bigsm root=/dev/system/root_lv showopts ide=nodma apm=off noresume edd=off powersaved=off nohz=off highres=off processor.max_cstate=1 nomodeset x11failsafe vga=0x314 intel_idle.max_cstate=0 processor.max_cstate=0 transparent_hugepage=never
```

```
initrd /initrd-3.0.101-0.46-bigsm
```

```
###Don't change this comment - YaST2 identifier: Original name: xen###
```

```
title Xen -- SLES for SAP Applications - 3.0.101-0.46
```

```
root (hdo,o)
```

```
kernel /xen.gz vga=mode-0x314
```

```
module /vmlinuz-3.0.101-0.46-xen root=/dev/system/root_lv instmode=cd resume=/dev/system/swap_lv splash=silent showopts vga=0x314 intel_idle.max_cstate=0 processor.max_cstate=0 transparent_hugepage=never
```

```
module /initrd-3.0.101-0.46-xen
```

```
###Don't change this comment - YaST2 identifier: Original name: linux###
```

```
title SLES for SAP Applications - 3.0.101-0.40
```

```
root (hdo,o)
```

```
kernel /vmlinuz-3.0.101-0.40-default root=/dev/system/root_lv instmode=cd resume=/dev/system/swap_lv  
splash=silent showopts vga=0x3140 intel_idle.max_cstate=0 processor.max_cstate=0 transparent_hugepage=never
```

```
initrd /initrd-3.0.101-0.40-default
```

```
###Don't change this comment - YaST2 identifier: Original name: failsafe###
```

```
title Failsafe -- SLES for SAP Applications - 3.0.101-0.40
```

```
root (hdo,o)
```

```
kernel /vmlinuz-3.0.101-0.40-default root=/dev/system/root_lv showopts ide=nodma apm=off noresume edd=off  
powersaved=off nohz=off highres=off processor.max_cstate=1 nomodeset x11failsafe vga=0x314  
intel_idle.max_cstate=0 processor.max_cstate=0 transparent_hugepage=never
```

```
initrd /initrd-3.0.101-0.40-default
```

HANA Configuration Files:

The following are the configuration

HANA Global.ini file:

```
[system_information]
```

```
usage=test
```

[communication]

listeninterface = .global

[persistence]

basepath_datavolumes=/hana/data/PS1

basepath_logvolumes=/hana/log/PS1

use_mountpoints = yes

basepath_shared = yes

[storage]

ha_provider = hdb_ha.fcClient

partition_*_*__prtype = 5

partition_1_data__wwid = 3624a93707d4c758cacd1f75000011058

partition_1_log__wwid = 3624a93707d4c758cacd1f7500001105b

partition_2_data__wwid = 3624a93707d4c758cacd1f75000011059

partition_2_log__wwid = 3624a93707d4c758cacd1f7500001105c

[trace]

ha_fcclient = info

Nameserver.ini:

[landscape]

id = 55bc4a10-32fe-2f3b-e100-00000a15272d

master = saphananode1:30001 saphananode2:30001 saphananode3:30001

worker = saphananode1 saphananode2

active_master = saphananode1:30001

```
standby = saphananode3
roles_saphananode1 = worker
roles_saphananode3 = standby
roles_saphananode2 = worker
```

Saprofile.ini:

```
# Basis
OS_UNICODE=nuc
SAPSYSTEM=oo
SAPSYSTEMNAME=PS1
INSTANCE_NAME=HDB$(SAPSYSTEM)
SAPLOCALHOST=saphananode1
ipc/shm_permission_01=600
```

HWCCT: fsperf parameters

The following parameters were set for the new performance test tool from SAP HWCCT tool fsperf. This would change I/O behavior and to enhance the database to work with the file system and storage.

- `async_read_submit=off`
- `async_write_submit_blocks=new`

For more information regarding these parameters please refer to SAP Note 1943937. In order to use these parameters in SAP HANA you need to execute the following commands in the Linux shell as <sid>adm user.

```
hdbparam -paraset fileio "[<path>]" . async_write_submit_blocks=new
```

To set `async_write_submit_blocks` for Data persistence: (Check the select query on the view `M_VOLUME_IO_TOTAL_STATISTICS` shown below to get the information on the path and trigger ratios)

```
hdbparam -paraset fileio "[/hana/data/PS1/mnt0001/hdb0003/]" . async_write_submit_blocks=new
```

The command would return lines indicating the success of the parameter setting for all the services like NameServer, Preprocessor, IndexServer etc.

To find the path execute the following query as shown below:

SQL		Result		
<pre>select HOST, PORT, TYPE, PATH, CONFIGURATION, TRIGGER_WRITE_RATIO, TRIGGER_READ_RATIO from SYS.M_VOLUME_IO_TOTAL_STATISTICS</pre>				
	HOST	PORT	TYPE	PATH
1	saphananode1	30,003	DATA_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
2	saphananode1	30,003	LOG_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
3	saphananode1	30,003	LOG	/hana/log/PS1/mnt00001/hdb00003/
4	saphananode1	30,003	DATA	/hana/data/PS1/mnt00001/hdb00003/
5	saphananode1	30,003	LOG_BACKUP	/usr/sap/PS1/HDB00/backup/log/
6	saphananode2	30,003	DATA_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
7	saphananode2	30,003	LOG_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
8	saphananode2	30,003	LOG	/hana/log/PS1/mnt00002/hdb00004/
9	saphananode2	30,003	DATA	/hana/data/PS1/mnt00002/hdb00004/
10	saphananode2	30,003	LOG_BACKUP	/usr/sap/PS1/HDB00/backup/log/
11	saphananode1	30,007	DATA_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
12	saphananode1	30,007	LOG_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
13	saphananode1	30,007	LOG	/hana/log/PS1/mnt00001/hdb00002/
14	saphananode1	30,007	DATA	/hana/data/PS1/mnt00001/hdb00002/
15	saphananode1	30,007	LOG_BACKUP	/usr/sap/PS1/HDB00/backup/log/
16	saphananode1	30,001	DATA_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
17	saphananode1	30,001	LOG_BACKUP	/usr/sap/PS1/SYS/global/hdb/backupint/
18	saphananode1	30,001	LOG	/hana/log/PS1/mnt00001/hdb00001/
19	saphananode1	30,001	DATA	/hana/data/PS1/mnt00001/hdb00001/
20	saphananode1	30,001	LOG_BACKUP	/usr/sap/PS1/HDB00/backup/log/

Figure 12: Select query for Volume IO statistics

Conclusion

Unlike traditional storage architectures, the Pure Storage FlashArray does not require customers to compromise on resiliency and simplicity for the sake of performance. Whether you are looking to get started with SAP HANA, or migrating an existing HANA environment, the FlashArray is the ideal platform for maximizing your HANA investment.

Pure Storage recommends following guidelines for sizing customer HANA environments. Please consult with your Pure Storage sales team for more detailed information.

AFA Model	Internal Shelf	No. Extension Shelves	SAP HANA Nodes
//m20	1	1	Up to 6
//m50	1	2	Up to 9
//m70	1	4	Up to 12

Table 2. Sizing Guidelines for Pure Storage FlashArray//m.

References

1. Pure Storage FlashArray Data Sheet - http://www.purestorage.com/pdf/Pure_Storage_FlashArray_Datasheet.pdf
2. Compare Flash Array with other Flash alternatives - http://www.purestorage.com/pdf/Pure_Storage_Flash_Array_vs_Flash_Alternatives.pdf
3. Pure Storage Forever Flash - http://www.purestorage.com/pdf/Pure_Storage_Forever_Flash_Datasheet.pdf



Pure Storage, Inc.
Twitter: [@purestorage](https://twitter.com/purestorage)
www.purestorage.com

650 Castro Street, Suite #260
Mountain View, CA 94041

T: 650-290-6088
F: 650-625-9667

Sales: sales@purestorage.com
Support: support@purestorage.com
Media: pr@purestorage.com
General: info@purestorage.com