Violin MEMORY

White Paper

# SAP HANA Appliances with Violin Memory

Reference Architecture for Installing and Configuring Violin Memory Arrays with SAP HANA

Version 3.0

***Abstract***

*SAP HANA enables organizations to optimize their business operations by analyzing large amounts of data in real time. It achieves very high performance without requiring any tuning. This document shows you the benefits of Violin Memory Flash Storage Systems in a HANA environment and guides through the setup of the solution.*

## Table of Contents

# 1. Introduction

Companies deploying HANA today are increasingly demanding simplicity, lowest total cost of ownership, and no compromise in performance and latency. Having timely access to information is critical, and application performance is constantly under pressure to access and process data in near-real time speeds. The reality of today's business is that transactions are processed 24/7, reports run and are generated 24 hours per day, and by everyone in the business – not just in finance or for executives.

Timely business insights ultimately translate into higher revenue and increased customer satisfaction, whether the customer is a partner, a client or an internal stakeholder. Downtime caused by ever-increasing complexity destroys profitability and trust, and limits the ability to grow.

To better enable decision makers, the IT infrastructure needs to support the volumes of data, access points, users that are currently growing exponentially. The ability to make ad hoc decisions based on queries that you can run dynamically provides the competitive advantage against those that have to wait weeks to gain access to reports. This means that the storage back-end supporting the SAP ecosystem needs to deliver real- time near zero-latency data access for continuous loading and storage. It also needs to support multiple reports that are run by multiple individuals at the same time – without performance degradation, without impact to other business functions, and at economies of scale.

Flash arrays ensure the lowest latency data access, provide high-bandwidth connections and scale to hundreds of TB's so even the largest databases can be stored in memory. With the ability to manage multiple workloads in a single all-flash array, Violin is fully certified on both SAP ASE/IQ and HANA and as such, users can drastically reduce hardware storage costs, consolidating and accelerating any element of the SAP ecosystem where performance is constrained by traditional disk-based storage.

Whether you are running an ERP application, HANA or ASE/IQ database or predictive analytics, running your SAP deployments on the Violin platform enables predictable performance and linearly scalability – all at extremely reduced lower total cost of ownership.

Running HANA on the certified Violin Memory platform gives you control over SAP architecture with:

- Uncompromised performance at up to 10X compared to legacy storage
- Consistent low-latency and throughput across scale-out and scale-up deployments
- Built-in, enterprise-grade high availability redundancy
- Exponential reduction of administration, installation and maintenance overhead

## 2. The Violin Memory 6000 Storage Array

The Violin 6000 Memory Array represents the consolidation of a rack of data center storage equipment into a single 3U platform and provides the benefits of “a rack in a box.” This consolidation is made possible through the ongoing development of flash memory and Violin’s specific flash memory technologies, including both aggregation and vRAID.

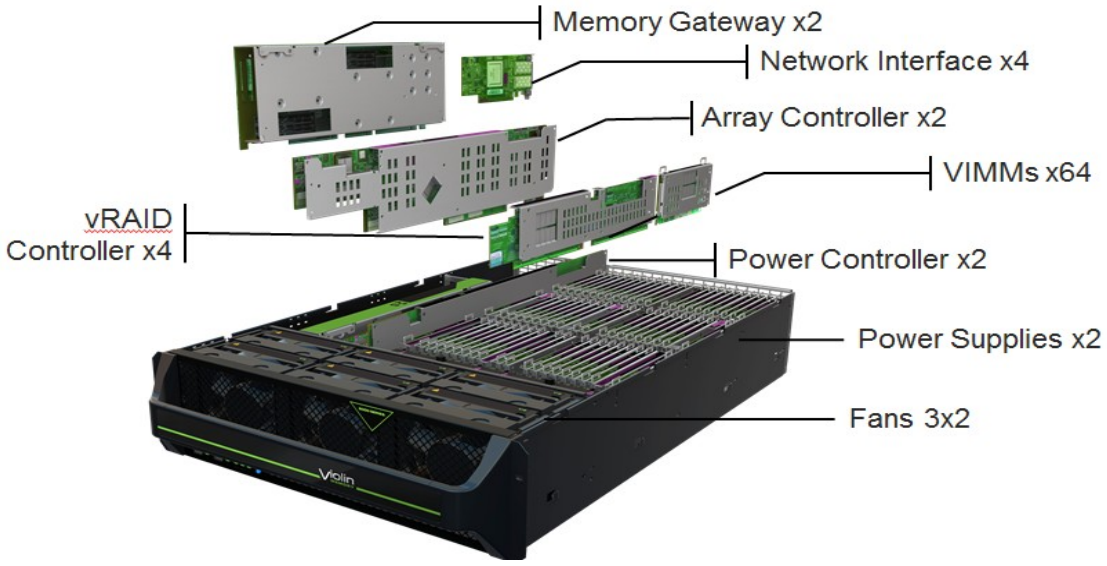
With no single point of failure, the Violin 6000 Memory Array is suitable for use as mission-critical primary flash storage. By moving from HDD storage to Violin 6000 Memory Arrays, the primary benefits are:

- 80% reduction in power and rack space
- 90% reduction in latency
- 80% reduction in Cost per IOP
- 50x increase in IOPS per shelf

Here is a concise list of the key elements of the Violin 6000 flash Memory Array Platform Architecture and capability along with a supporting graphical description.

- Violin switched memory (VXM) interconnect provides Performance, Scalability, Density, Availability
- vRAID controllers for non-blocking erasure, massively parallel reads/writes, and sustained performance
- Redundant active/active gateways
- Non-disruptive firmware upgrade (2QCY2012)
- Fail-in-place Flash Memory on VIMMs

Figure 5-1 provides a photographic illustration of these hardware components.










03-0302 rev 0.12 05/14

Additional architectural elements include the following:

- Redundant power supplies
- Passive Memory Plane – Provides interconnection between modules and controllers
- Redundant Power Controllers (PCs) – Power management controllers, modules and fans
- Redundant Array Controllers (ACs) – Provide PCIe interfaces, switching & management services
- Network Interfaces (NIs) – The Network Interfaces connect via PCIe to the Memory Gateways and provide the physical connectivity to the SAN or LAN

All active modules listed above are hot swappable.

## 2.1. Why Store SAP HANA Data on Violin Arrays?

The SAP HANA database resides in memory in a columnar architecture. All read/write activity is occurring in memory. But the system also writes to a traditional hard-drive based database, which resides in fact inside the Violin 6000 storage array.

The real purpose of this traditional database is for system restarts. By using the Violin Memory 6000 Series Array, SAP HANA is now able to recover back from Tier 1 storage into the in-memory database much faster compared to a traditional disk based storage system. The main goal is to reduce network latency when serving up data from a storage array. With flash memory, Violin is also getting rid of disk latency.

## 2.2. How Applications Benefit from Violin Technology

The Violin 6000 Series is ideal for applications with large amounts of data that require active processing or rapid access. These applications include:

- Transaction processing (e.g. Oracle, DB2, SQL Server, Sybase)
- Data Warehousing and Business Intelligence
- Extract, Transform, Load (ETL) for Data Warehouses
- Real-time Analytics
- System Integration and Test
- Active file storage and metadata for file systems (e.g. GPFS, Lustre)
- Tier 1 storage for High Volume Messaging systems (e.g. Email)
- Software Configuration Management
- Multi-user software development environments
- Multi-user CAD/CAM environments

Each of these applications scales more easily with storage that has both high IOPS and low latency. The increase in storage performance leads to both improved user response times and more efficient use of server, CPU, and software resources.

## 2.3. Network and Cabling Overview for Violin based HANA Appliance

Figure 4-2 shows the network and fiber channel diagram of the configuration

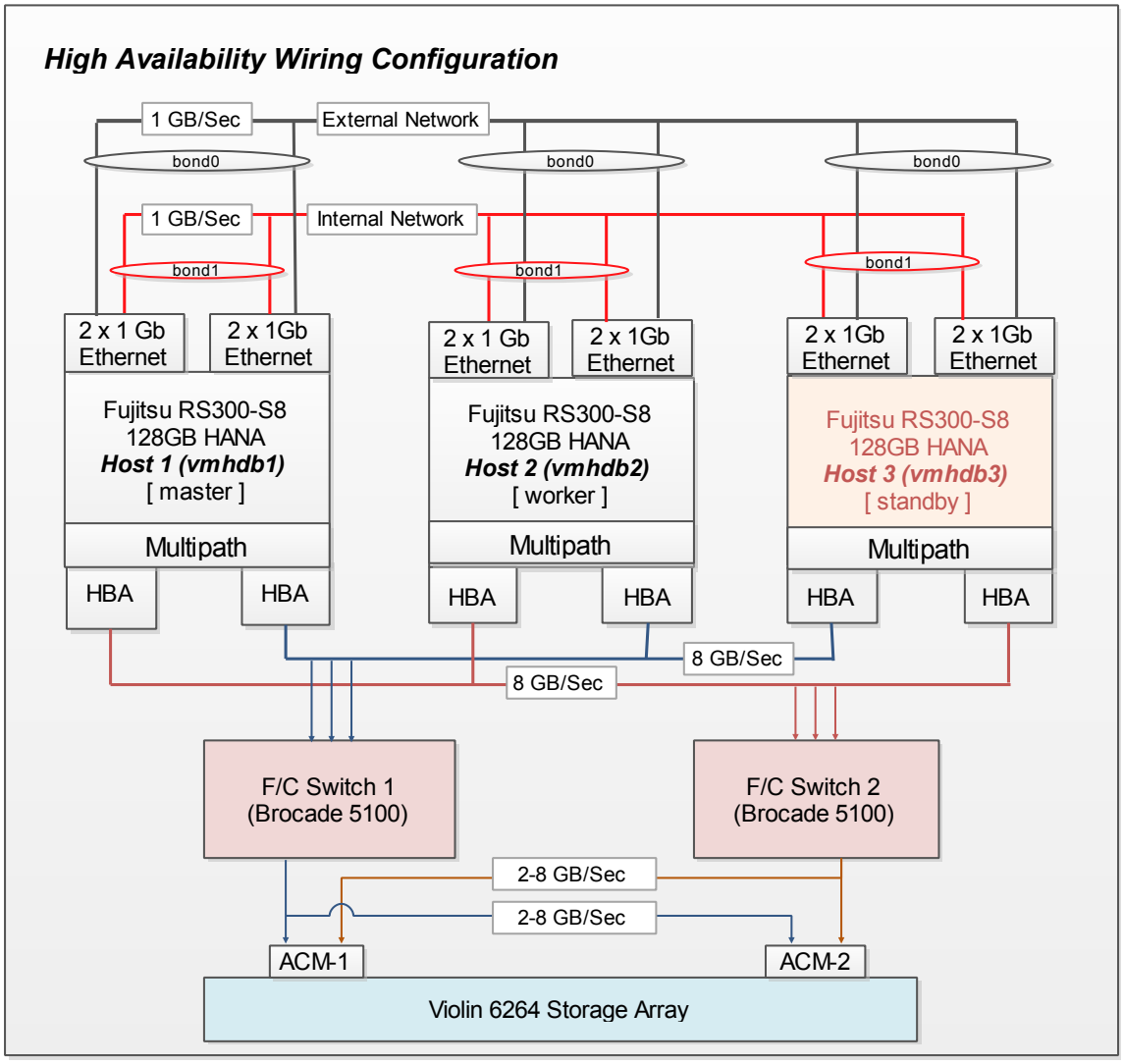



*Figure 4-2. Network and Cabling for Rack 1*

## 2.4. DNS naming and IP addresses for IPMI , public and heartbeat LAN

DNS Naming for the nodes was configured as shown below

| HANA Node | role | Server IPMI | public | Public-HA eth0+eth4 | internal heartbeat eth2+eth5 |
|---|---|---|---|---|---|
| vmhdb1 | master | vmhdb1.ipmi | vmhdb1 | bond0 | bond1 |
| vmhdb2 | worker | vmhdb2.ipmi | vmdhb2 | bond0 | bond1 |
| vmhdb3 | standby | vmhdb2.ipmi | vmhdb3 | bond0 | bond1 |

The Violin Array requires 6 dns-resolvable hostnames/IP addresses for management as below

| Host-name | Function |
|---|---|
| 6264-1-mg-vip | Virtual IP for Memory Gateway |
| 6264-1-mg-a | IP address for Memory Gateway A |
| 6264-1-mg-b | IP address for Memory Gateway B |
| 6264-1-acm-vip | Virtual IP for Array Controller Module |
| 6264-1-acm-a | IP address for Array Controller Module A |
| 6264-1-acm-b | IP address for Array Controller Module B |

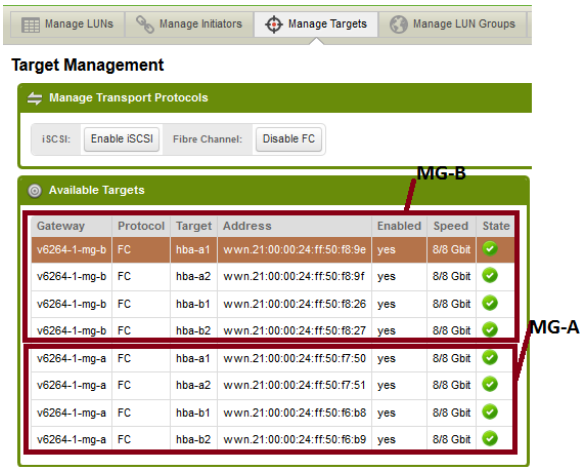
## 2.5. LUN Configuration & FC Connectivity of Violin 6000 to HANA nodes

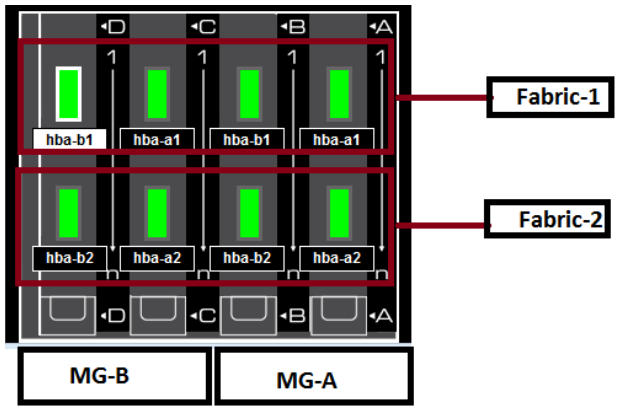
LUNs were created as per HANA sizing requirements and exported to all the nodes using Initiator group configuration. All LUNs were created as Active/Active .The below screenshot shows the details of the LUN names and LUN sizes


All 8 array targets were connected to a pair of brocade switches using 50 micron fiber cable.

The below figure shows the array management showing the FC target ports and cabling of the targets ports to the redundant fabrics.

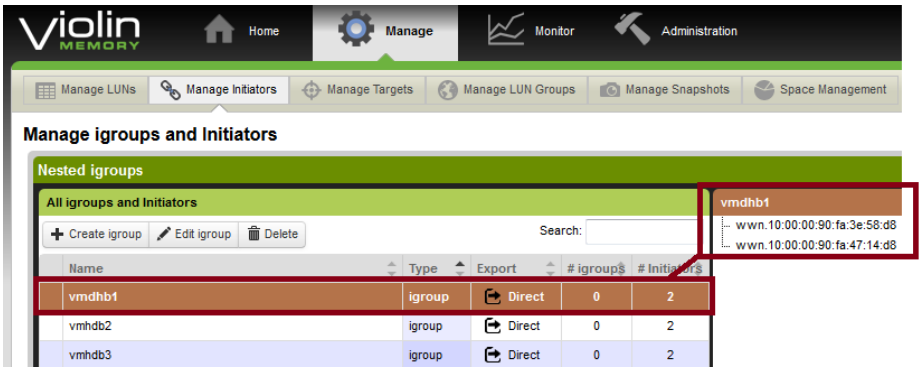
This topology ensures full redundancy for MG and Fabric failures.




Initiator groups were configured for each HANA node as shown below.

The below screenshot shows the initiator group for vmhdb1 which has 2 HBA ports configured in it.










03-0302 rev 0.12 05/14

The SAN aliasing/Zoning is done as follows for the two fabric Switches: Fabric 1 and Fabric 2.

| SAN ALIAS | Fabric | WWPNs |
|---|---|---|
| violin_tgt_top_ports | Fabric1 | Gateway / Protocol / Target / Address / Enabled<br>v6264-1-mg-b / FC / hba-a1 / wwn.21:00:00:24:ff:50:f8:9e / yes<br>v6264-1-mg-b / FC / hba-a2 / wwn.21:00:00:24:ff:50:f8:9f / yes<br>v6264-1-mg-b / FC / hba-b1 / wwn.21:00:00:24:ff:50:f8:26 / yes<br>v6264-1-mg-b / FC / hba-b2 / wwn.21:00:00:24:ff:50:f8:27 / yes |
| violin_tgt_bottomport | Fabric2 | v6264-1-mg-a / FC / hba-a1 / wwn.21:00:00:24:ff:50:f7:50 / yes<br>v6264-1-mg-a / FC / hba-a2 / wwn.21:00:00:24:ff:50:f7:51 / yes<br>v6264-1-mg-a / FC / hba-b1 / wwn.21:00:00:24:ff:50:f6:b8 / yes<br>v6264-1-mg-a / FC / hba-b2 / wwn.21:00:00:24:ff:50:f6:b9 / yes |
| Vmhdb1_initiators_1 | Fabric1 | 10000090fa4714d8 |
| Vmhdb1_initiators_2 | Fabric2 | 10000090fa3e58d8 |
| Vmhdb2_initiators_1 | Fabric1 | 10000090fa4719e2 |
| Vmhdb2_initiators_2 | Fabric2 | 10000090fa3e5892 |
| Vmhdb3_initiators_1 | Fabric1 | 10000090fa471468 |
| Vmhdb3_initiators_2 | Fabric2 | 10000090fa471f36 |

The aliases were grouped into zones as per the below table.

| zone | Fabric | Host Initiators | Array Targets |
|---|---|---|---|
| vmhdb1_zone_f1 | Fabric1 | vmhdb1_initiators_1 | violin_tgt_top_ports |
| vmhdb1_zone_f2 | Fabric2 | vmhdb1_initiators_2 | violin_tgt_bottomport |
| vmhdb2_zone_f1 | Fabric1 | vmhdb2_initiators_1 | violin_tgt_top_ports |
| vmhdb2_zone_f2 | Fabric2 | vmhdb2_initiators_2 | violin_tgt_bottomport |
| vmhdb3_zone_f1 | Fabric1 | vmhdb3_initiators_1 | violin_tgt_top_ports |
| vmhdb3_zone_f2 | Fabric2 | vmhdb3_initiators_2 | violin_tgt_bottomport |

The Effective zoning for both the fabrics is as follows.

| Zone-name | | Vmhdb1_zone | Vmhdb2_zone | Vmhdb3_zone |
|---|---|---|---|---|
| Fabric1-config | Aliases In zone | Violin_tgt_top_ports<br>Vmhdb1_initiators_1 | Violin_tgt_top_ports<br>Vmhdb2_initiators_1 | Violin_tgt_top_ports<br>Vmhdb3_initiators_1 |
| Fabric2-config | | Violin_tgt_bottom_ports<br>Vmhdb1_initiators_2 | Violin_tgt_ bottom _ports<br>Vmhdb2_initiators_2 | Violin_tgt_ bottom _ports<br>Vmhdb3_initiators_2 |

# 3. SLES Linux Setup

The HANA nodes come preinstalled with SLES11SP3 for SAP Applications.

There are a series of host customization steps to be done for HANA.

The higher-level steps are described as follows.

**1. Hostname/IP address configuration for basic connectivity**

- Login via IPMI console
- Set the ip address in /etc/hosts
- Bring up eth0 on public network.
- Configure DNS and default gateway
- Ensure you can ping outside the server
- Disable the firewall and make sure it doesn't start again

**2. Enabling root login via ssh to enable additional configuration**

- Edit /etc/ssh/sshd_config file and set PermitRootLogin=yes
- Restart sshd (service sshd restart)
- Verify you can ssh to the public IP from outside as root user.

**3. Enabling password-less ssh communication between the nodes**

- Enablign passwordless ssh is a requirement for HANA
- All nodes do the following
    - Generate a RSA key ( ssh-keygen -t rsa )
    - Using a blank passphrase Create the .ssh directory on all nodes

```
ssh root@vmhdb1 mkdir -p .ssh ( will ask for root password)
ssh root@vmhdb2 mkdir -p .ssh ( will ask for root password)
ssh root@vmhdb3 mkdir -p .ssh ( will ask for root password)
```

    - Copy the key to all other hosts and enter root password when prompted

```
cat .ssh/id_rsa.pub | ssh root@vmhdb2 'cat >> .ssh/authorized_keys'
cat .ssh/id_rsa.pub | ssh root@vmhdb2 'cat >> .ssh/authorized_keys'
```

    - Repeat above steps on all nodes
- Test passwordless ssh communication

### 4. Configuring and enabling bonding between the public interfaces (eth0/eth4) for high-availability and for eth2/eth5 for HANA heartbeat

- Work off the system console
- Setup the following file for bond0 as master and eth0 and eth4 as slaves.

**file:/etc/sysconfig/network/ifcfg-bond0**

```
BONDING_MASTER='yes'
BONDING_MODULE_OPTS='mode=active-backup miimon=100'
BOOTPROTO='static'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR='10.1.73.140/24'
MTU=''
NAME='bond0'
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='onboot'
USERCONTROL='no'
BONDING_MODULE_OPTS='mode=active-backup miimon=100 use_carrier=0'
BONDING_SLAVE0='eth0'
BONDING_SLAVE1='eth4'
```

**file:/etc/sysconfig/network/ifcfg-eth0**

```
BOOTPROTO='none'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR=''
MTU=''
NAME='Intel Ethernet controller'
NETMASK=''
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='off'
USERCONTROL='no'
```

```
file:/etc/sysconfig/network/ifcfg-eth4
BOOTPROTO='none'
BROADCAST=''
ETHTOOL_OPTIONS=''
IPADDR=''
MTU=''
NAME='Intel Ethernet controller'
NETMASK=''
NETWORK=''
REMOTE_IPADDR=''
STARTMODE='off'
USERCONTROL='no'
```

- Reboot the server and test bonding. Disable network interfaces eth0 and eth4 interchangeably to see if you can survive a ssh connection

**5. Configuring private IP addresses for eth2 and eth5 for HANA heartbeat communication of shared hana file-system**

- Use the same procedure as above to create bond1

## 4. Multipath-IO configuration for Violin Array

Run the below commands to make sure multipath daemon is running at boot time.

```
# chkconfig multipathd on
#service multipathd start
# multipath –F
# multipath –v2
# multipath –ll
# /etc/init.d/irqbalance start
# chkconfig irqbalance on
```

Populate a /etc/multipath.conf file as follows.

---

```
blacklist {
        devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
        devnode "^hd[a-z]"
        devnode "^dcssblk[0-9]*"
        }
blacklist_exceptions {
        property "(ID_SCSI_VPD|ID_WWN)"
}
```

```
defaults {
               user_friendly_names yes
               bindings_file "/etc/multipath/bindings"
                uid_attribute "ID_SERIAL"
}
devices {
    device {
            vendor                  "VIOLIN"
            product                 "SAN ARRAY ALUA"
            path_grouping_policy    group_by_prio
#           getuid_callout          "/sbin/scsi_id --whitelisted --replace-whitespace --
page=0x80 --device=/dev/%n"
            prio                    alua
            path_checker            tur
            path_selector           "round-robin 0"
            hardware_handler        "1 alua"
            failback                immediate
            rr_weight               uniform
            no_path_retry           0
            #no_path_retry           fail
            rr_min_io_rq            200
            features                "0"
   }

   device {
           vendor                  "VIOLIN"
           product                 "SAN ARRAY"
           path_grouping_policy    group_by_serial
#           getuid_callout          "/lib/udev/scsi_id --whitelisted --replace-whitespace --
page=0x80 --device=/dev/%n"
           path_checker            tur
           path_selector           "round-robin 0"
           hardware_handler        "0"
           failback                immediate
           rr_weight               uniform
           no_path_retry           fail
           rr_min_io_rq            200
   }
}
multipaths {
multipath {
 # Container 81349F00340
 wwid 36001b970f24554e1f24554e19c8cae84
 alias vmhdb1_data
 }
```







03-0302 rev 0.12 05/14

```
multipath {
 # Container 81349F00340
 wwid 36001b970f24554e1f24554e14833c510
 alias vmhdb1_log
 }
 multipath {
 # Container 81349F00340
 wwid 36001b970f24554e1f24554e118a9d796
 alias vmhdb2_data
 }
 multipath {
 # Container 81349F00340
 wwid 36001b970f24554e1f24554e1cc16bc02
 alias vmhdb2_log
 }
}
```

# 5. Setting and enabling udev rules for Violin Array performance

Populate udev rules as below .

Udev rule file = /etc/udev/rules.d/12-violin.rules

```
-------------------------------------------------------------------------------------------------
---------------------------------
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo noop >
/sys/$devpath/queue/scheduler'"
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo 0 >
/sys/$devpath/queue/rotational'"
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo 1024 >
/sys/$devpath/queue/nr_requests'"
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo 2 >
/sys/$devpath/queue/rq_affinity'"
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo 1 >
/sys/$devpath/queue/nomerges'"
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo 4096 >
/sys/$devpath/queue/max_sectors_kb'"
KERNEL=="sd*[!0-9]|sg*", BUS=="scsi", SYSFS{vendor}=="VIOLIN", RUN+="/bin/sh -c 'echo 0 >
/sys/$devpath/queue/add_random'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo noop >
/sys/$devpath/queue/scheduler'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo 0 > /sys/$devpath/queue/rotational'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo 1024 >
/sys/$devpath/queue/nr_requests'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo 2 > /sys/$devpath/queue/rq_affinity'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo 1 > /sys/$devpath/queue/nomerges'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo 4096 >
/sys/$devpath/queue/max_sectors_kb'"
ENV{DM_UUID}=="mpath-36001b97?*", RUN+="/bin/sh -c 'echo 0 >  /sys/$devpath/queue/add_random'"
```

Then run the following commands to get the rule to take effect or reboot the server

```
# chmod 0644 /etc/udev/rules.d/60-violin.rules
# udevcontrol --reload_rules
# udevtrigger
```

# 6. HANA Configuration

## 6.1. HANA Configuration File Parameters

| Name | Parameter | Value | Restrictions |
|---|---|---|---|
| SAP System ID | sid | VM1 | Upper case, 3 alphanumeric |
| Instance Number | instanceNumber | 00 | 00 – 99 |
| Mount Path | sapmntPath | /hana/shared | |
| Data File Path | dataPath | /hana/data/VM1 | |
| Log File Path | logPath | /hana/log/VM1 | |
| DB Host Name | hdbHost | vmhbd1 | Fully qualified name vmhdb(n) .Consistent case across hosts. Case sensitive |
| External Host Name | externalHost | vmhdb1.coil.sap.com | |
| Time Zone | timezone | PDT | |
| Storage | storageConfigPath | /hana/shared/VM1 | |
| Additional host (2) | additionalHostsRootUser | root | |
| | host | vmhdb2 | additional host (where HDB server is located) |
| | hostname | vmhdb2.coil.sap.com | Insert the fully qualified host name of the additional HDB server. |
| | hostType | worker | ‘worker’ or ‘standby’ |
| | haGroup | vmhdb | |
| Additional host (3) | additionalHostsRootUser | root | |
| | host | vmhdb2 | additional host (where HDB server is located) |
| | hostname | vmhdb1.coil.sap.com | Insert the fully qualified host name of the additional HDB server. |
| | hostType | standby | ‘worker’ or ‘standby’ |
| | haGroup | vmhdb | |

## 6.2. HANA Global.Ini

```
[communication]
listeninterface = .global
[internal_hostname_resolution]
192.168.2.1 = vmhdb1-int
192.168.2.2 = vmhdb2-int
192.168.2.3 = vmhdb3-int
[persistence]
basepath_datavolumes = /hana/data/VM1
basepath_logvolumes = /hana/log/VM1
use_mountpoint = yes
[storage]
ha_provider = hdb_ha.fcClient
partition_*_*__mountoptions = -t xfs
partition_2_log__alias = vmhdb2_log
partition_2_data__alias = vmhdb2_data
partition_1_log__alias = vmhdb1_log
partition_1_data__alias = vmhdb1_data
[trace]
ha_fcclient = info
```

## 6.3. HANA name server.ini

```
[landscape]
id = 53634ef0-fbfe-3f77-e100-00000a01498c
master = vmhdb1:30001 vmhdb2:30001 vmhdb3:30001
worker = vmhdb1 vmhdb2
active_master = vmhdb1:30001
standby = vmhdb3
```

## 6.4. HANA sapprofile.ini

```
# Basis
OS_UNICODE=nuc
SAPSYSTEM=00
SAPSYSTEMNAME=VM1
INSTANCE_NAME=HDB$(SAPSYSTEM)
SAPLOCALHOST=vmhdb1
ipc/shm_permission_01=600
```

# 7. Conclusion

SAP HANA is an amazingly fast database platform and combined with Violin flash storage, it can do many tasks thousands of times faster than other RDBMS platforms.

SAP HANA has opened up new opportunities for companies to compete smarter using the latest advanced business applications and analytics required by employees to make better, timelier decisions.

## About Violin Memory

*Business in a Flash*. Violin Memory transforms the speed of business with high performance, always available, low cost management of critical business information and applications. Violin's All Flash optimized solutions accelerate breakthrough CAPEX and OPEX savings for building the next generation data center. Violin's Flash Fabric Architecture™ (FFA) speeds data delivery with chip-to-chassis performance optimization that achieves lower consistent latency and cost per transaction for Cloud, Enterprise, and Virtualized mission-critical applications. Violin's All Flash Arrays and Appliances, and enterprise data management software solutions enhance agility and mobility while revolutionizing data center economics. Founded in 2005, Violin Memory is headquartered in Santa Clara, California.

For more information about Violin Memory products, visit www.violin-memory.com.







03-0302 rev 0.12 05/14