TECHNICAL BRIEF:
SOFTWARE-DEFINED STORAGE AT THE SPEED
OF FLASH

Intel® SSD Data Center P3700 Series and Symantec™ Storage Foundation with Flexible Storage Sharing

Software-defined Storage at the Speed of Flash

Who should read this paper

Architects, application, database and storage owners who are designing mission-critical systems for performance without compromising on flexibility or data availability

Carlos Carrero
Sr. Technical Product Manager

Content

Introduction

Solid State drives are changing the shape of the data center. Myths around lifespan and reliability are a thing of the past and as $/GB continues to decline, the value of SSDs is now unquestionable. While a few years ago a few GB was the norm, today's Intel® Solid-State Drive Data Center P3700 Series drives possess capacities up to 2TB with incredible performance and resiliency. At the same time, servers are maximizing the number of drive bays. Combining denser, more powerful CPUs, and several terabytes of high performance storage is driving a shift away from SAN to software-defined infrastructures.

Intel R1208WTTGS 1U Server

This white paper describes how to use Symantec software to consume and protect Intel storage, resulting in a small, fast, low-cost architecture ideal for running multiple high-performance Oracle databases.

The result is a pair of fast, flexible application 1U platforms that achieve better performance **(up to 184%)** than alternative architectures using All Flash Arrays at a fraction of the cost **(5-10%)**

Architecture Setup

Hardware

2 x Intel® R1208WTTGS 1U servers, each having the following configuration:

- 2 x Intel® Xeon® Processor E5-2697 v3 @ 2.60GHz (14 cores / 28 threads per processor)
- 4 x Intel® Solid-State Drive Data Center P3700 Series (800 GB 2.5" SSD)
- 1 dual-port Intel® Ethernet Converged Network Adapter XL710-QDA2 (interconnects)
- 128 GB DDR4 Memory

Software

- Symantec™ Storage Foundation Cluster File System 6.2
- Oracle 11gR2 Single Instance
- Red Hat Enterprise Linux 6.5

Architecture

The two servers are connected using a high speed interconnect. Flexible Storage Sharing in Cluster File System is used to create volumes that are mirrored across the two servers in the cluster, protecting against data loss should any server downtime occur.

In our testing, single instance Oracle 11gR2 Enterprise instances are used. While Cluster File System also offers a RAC option for parallel applications, in this architecture we are going to examine how using Cluster File System's Fast Failover capabilities, a single instance can be made available in the other server very quickly, avoiding costs and complexities of RAC. In any case, RAC is another option but it is not discussed in this architecture.

Architecture

The instances will be made highly available using Symantec™ Cluster Server, powered by Veritas. Data will be always accessible from any node within the cluster.

Intel® Solid-State Drive Data Center P3700 Series

The Intel® Solid-State Drive (SSD) Data Center Family for PCIe* brings extreme data throughput directly to Intel® Xeon® processors with up to six times faster data transfer speed than 6 Gbps SAS/SATA SSDs[1]. Intel SSD DC P3700 Series is capable of reading data up to 2.8GB/s and 460k Input/Output Operations Per Second (IOPS) and writing up to 2.0GB/s and 175k IOPS. Taking advantage of the direct path from the storage to the CPU by means of NVMe, P3700 Series exhibits low latency of less than 20 µs for sequential access to the SSD. The P3700 Series also includes High Endurance Technology (HET), which combines NAND silicon enhancements and SSD NAND management techniques to extend SSD write endurance up to 10 drive writes per day for 5 years[2].

Intel® Solid-State Drive Data Center P3700 Series 2.5" Drive

Flexible Storage Sharing in Cluster File System

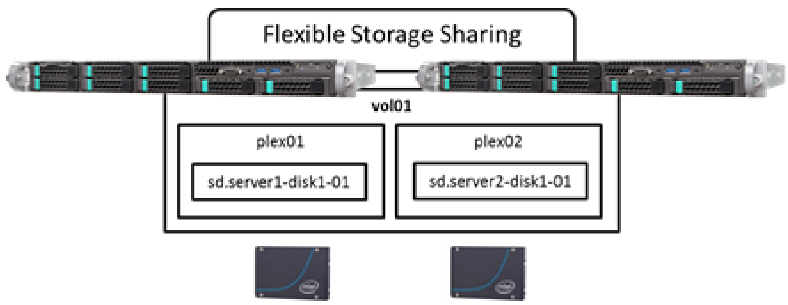
Flexible Storage Sharing is a feature of Cluster File System that allows any local device to be shared with other members of the cluster. This allows a highly resilience configuration, extending traditional volume manager and file system capabilities to share nothing environments. Traditionally, a SAN is needed in order to provide high availability for data. When a node fails, data needs to be accessible to other nodes. With a powerful software-defined storage platform like Cluster File System, Direct Attached Storage is always accessible.

Flexible Storage Sharing is fully integrated with Cluster File System and exposed through the Cluster Volume Manager component. With a single step, internal devices are shared across the cluster and those devices will be used as if they were local. Cluster Volume Manager hides all the complexities and uses internal components to provide visibility of internal storage. All the IO is shipped across the internal cluster interconnects. As an addition to Flexible Storage Sharing, InfiniBand with RDMA is supported in order to provide a high performance and low latency interconnects.

1. http://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/intel-ssd-dc-family-for-pcie-brief.pdf
2. http://www.intel.com/content/dam/www/public/us/en/documents/product-specifications/ssd-dc-p3700-spec.pdf

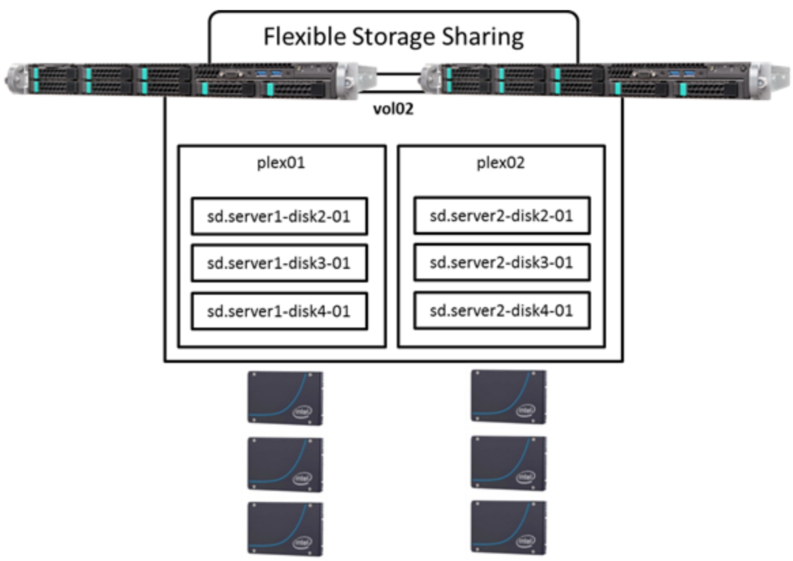
Flexile Storage Sharing has expanded traditional Volume Manager capabilities to use any exported device as if it were local. A traditional high availability configuration is composed by two copies of data. In case of a disk failure, the other copy will maintain the volume and file system available, and ideally, across different hardware and data center racks. The following pictures shows the how Flexible Storage Sharing can use DAS and apply concepts of volumes, plexes and sub-disks for data protection.

In our configuration we have a RAID1 configuration, or in other terms, a volume containing two copies with one copy in each server using one sub-disk per server.



Concat-mirror configuration

Flexible Storage Sharing allows any traditional storage layout configuration supported with Cluster Volume Manager. For the Oracle data files used in this white paper architecture, three internal disks in a RAID0+1, or stripe-mirror, configuration have been used.



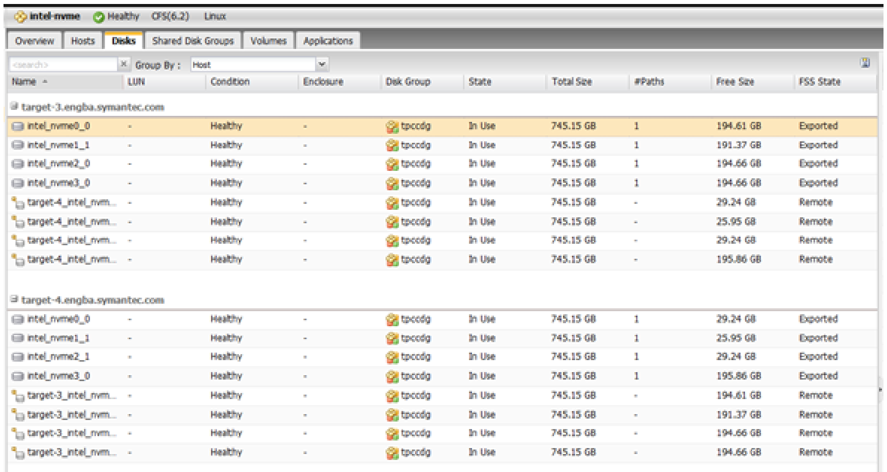
Stripe-mirror configuration

Flexible Storage Sharing is a key enabler to unlock the capacities of the internal storage while providing flexibility around protection and layout schemes controlled by the software. Only internal Intel SSDs cards are used and copies of data are available in both nodes, providing a high availability configuration without the need of a SAN providing much higher performance.

The same systems will also run and protect our Oracle instances, to provide a converged infrastructure. There is no need for external servers or storage arrays with a minimal 2U footprint.

File System Layout

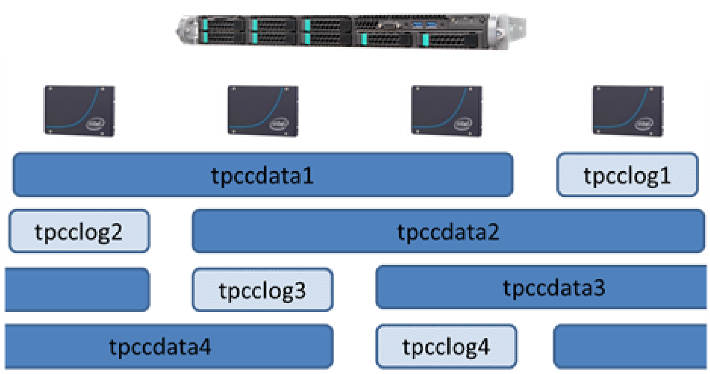
As we only have four internal Intel SSD cards per server, these will be added to the same Disk Group called *tpccdg*. Each of the cards is exported so it can be seen from the other server. In total, each server will see 8 SSD cards and will treat them as local. The following picture shows the storage seen by each server. Note the disk icon and the *FSS State* column that denotes whether it is a local or remote disk.



Name	LUN	Condition	Enclosure	Disk Group	State	Total Size	#Paths	Free Size	FSS State
target-3.engba.symantec.com									
intel_nvme0_0	-	Healthy	-	tpccdg	In Use	745.15 GB	1	194.61 GB	Exported
intel_nvme1_1	-	Healthy	-	tpccdg	In Use	745.15 GB	1	191.37 GB	Exported
intel_nvme2_0	-	Healthy	-	tpccdg	In Use	745.15 GB	1	194.66 GB	Exported
intel_nvme3_0	-	Healthy	-	tpccdg	In Use	745.15 GB	1	194.66 GB	Exported
target-4_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	29.24 GB	Remote
target-4_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	25.95 GB	Remote
target-4_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	29.24 GB	Remote
target-4_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	195.86 GB	Remote
target-4.engba.symantec.com									
intel_nvme0_0	-	Healthy	-	tpccdg	In Use	745.15 GB	1	29.24 GB	Exported
intel_nvme1_1	-	Healthy	-	tpccdg	In Use	745.15 GB	1	25.95 GB	Exported
intel_nvme2_1	-	Healthy	-	tpccdg	In Use	745.15 GB	1	29.24 GB	Exported
intel_nvme3_0	-	Healthy	-	tpccdg	In Use	745.15 GB	1	195.86 GB	Exported
target-3_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	194.61 GB	Remote
target-3_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	191.37 GB	Remote
target-3_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	194.66 GB	Remote
target-3_intel_nvm...	-	Healthy	-	tpccdg	In Use	745.15 GB	-	194.66 GB	Remote

Storage

The different volumes for data and redo logs will be carved from this disk group. We are going to keep the configuration quite simple, having one volume for data that will be striped across three cards, and another volume for redo log that will be carved from a single card. We will have four concurrent Oracle instances running, so to optimize for the different IO patterns, the device used to store the redo logs for each of the instances will be rotated, so there will not be two redo logs volumes on the same physical disks. This will be the logical volume distribution for one server.

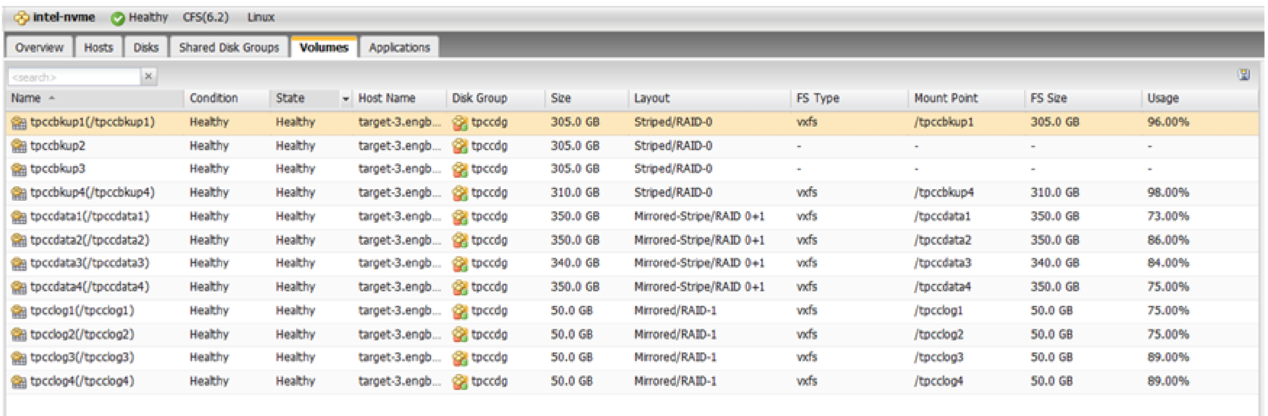


Logical Volume Configuration

This configuration will completely be mirrored across the other server, so there will be two logical copies in a stripe-mirror configuration (concat-mirror for the redo log volumes as only one drive per sever will be used)

Volumes Mirrored Across nodes

Using Veritas Operation Manager we can inspect the storage configuration via the Storage Clusters tab. In addition of the previous volumes used, another four backup volumes have also being created. The following picture shows all the file systems and volumes used.



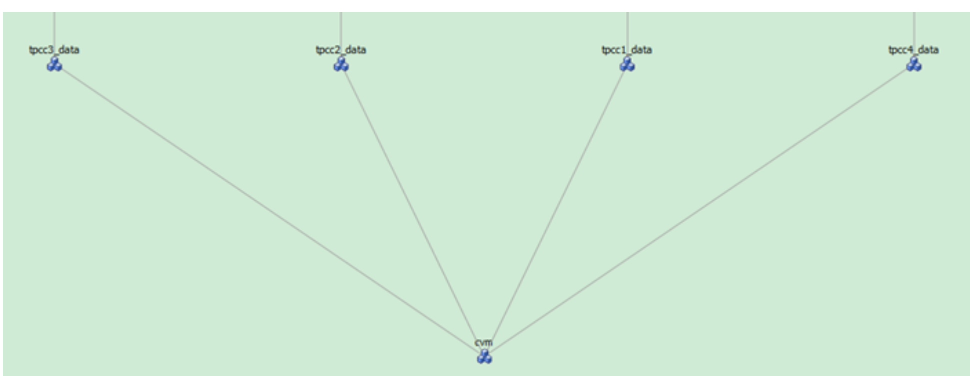
intel-nvme Healthy CFS(6.2) Linux

Overview | Hosts | Disks | Shared Disk Groups | Volumes | Applications

Name	Condition	State	Host Name	Disk Group	Size	Layout	FS Type	Mount Point	FS Size	Usage
tpccbkup1(/tpccbkup1)	Healthy	Healthy	target-3.engb...	tpccdg	305.0 GB	Striped/RAID-0	vxfs	/tpccbkup1	305.0 GB	96.00%
tpccbkup2	Healthy	Healthy	target-3.engb...	tpccdg	305.0 GB	Striped/RAID-0	-	-	-	-
tpccbkup3	Healthy	Healthy	target-3.engb...	tpccdg	305.0 GB	Striped/RAID-0	-	-	-	-
tpccbkup4(/tpccbkup4)	Healthy	Healthy	target-3.engb...	tpccdg	310.0 GB	Striped/RAID-0	vxfs	/tpccbkup4	310.0 GB	98.00%
tpccdata1(/tpccdata1)	Healthy	Healthy	target-3.engb...	tpccdg	350.0 GB	Mirrored-Stripe/RAID 0+1	vxfs	/tpccdata1	350.0 GB	73.00%
tpccdata2(/tpccdata2)	Healthy	Healthy	target-3.engb...	tpccdg	350.0 GB	Mirrored-Stripe/RAID 0+1	vxfs	/tpccdata2	350.0 GB	86.00%
tpccdata3(/tpccdata3)	Healthy	Healthy	target-3.engb...	tpccdg	340.0 GB	Mirrored-Stripe/RAID 0+1	vxfs	/tpccdata3	340.0 GB	84.00%
tpccdata4(/tpccdata4)	Healthy	Healthy	target-3.engb...	tpccdg	350.0 GB	Mirrored-Stripe/RAID 0+1	vxfs	/tpccdata4	350.0 GB	75.00%
tpcclog1(/tpcclog1)	Healthy	Healthy	target-3.engb...	tpccdg	50.0 GB	Mirrored/RAID-1	vxfs	/tpcclog1	50.0 GB	75.00%
tpcclog2(/tpcclog2)	Healthy	Healthy	target-3.engb...	tpccdg	50.0 GB	Mirrored/RAID-1	vxfs	/tpcclog2	50.0 GB	75.00%
tpcclog3(/tpcclog3)	Healthy	Healthy	target-3.engb...	tpccdg	50.0 GB	Mirrored/RAID-1	vxfs	/tpcclog3	50.0 GB	89.00%
tpcclog4(/tpcclog4)	Healthy	Healthy	target-3.engb...	tpccdg	50.0 GB	Mirrored/RAID-1	vxfs	/tpcclog4	50.0 GB	89.00%

File Systems and Volumes

There will be one parallel service group to manage the storage used by each instance. The service group will make sure that the file systems with the data needed for each instance will be mounted all the time in each of the nodes. Having the data available in each node will reduce the restart time when one instance needs to be failed over from one node to the other. The four service groups used will have a dependency on top of *cvm* service group, which goal is to make sure internal storage is accessible across all the cluster nodes.



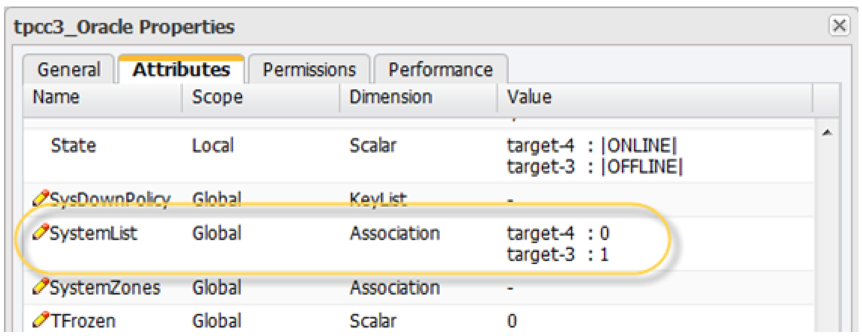
Data service group dependencies with cvm

Oracle Configuration

Four separate Oracle Single Instances have been configured for this architecture. Although Oracle RAC is a highly available and highly performance solution, there are some drawbacks, complexities and higher costs than when using a Single Instance. In this particular architecture, the high performance of Intel SSDs plus Fast Failover technology of Cluster File Systems allows a configuration that enables high performance and reduced recovery times for low-failover times without the cost of Oracle RAC. Single Instance configurations are quite easy to install and manage and have less maintenance overhead.

Cluster File System also offers ODM (Oracle Disk Manager) which enables the flexibility and simplicity of using a file system to store data, but at the speed of raw devices. Every instance in this configuration uses ODM.

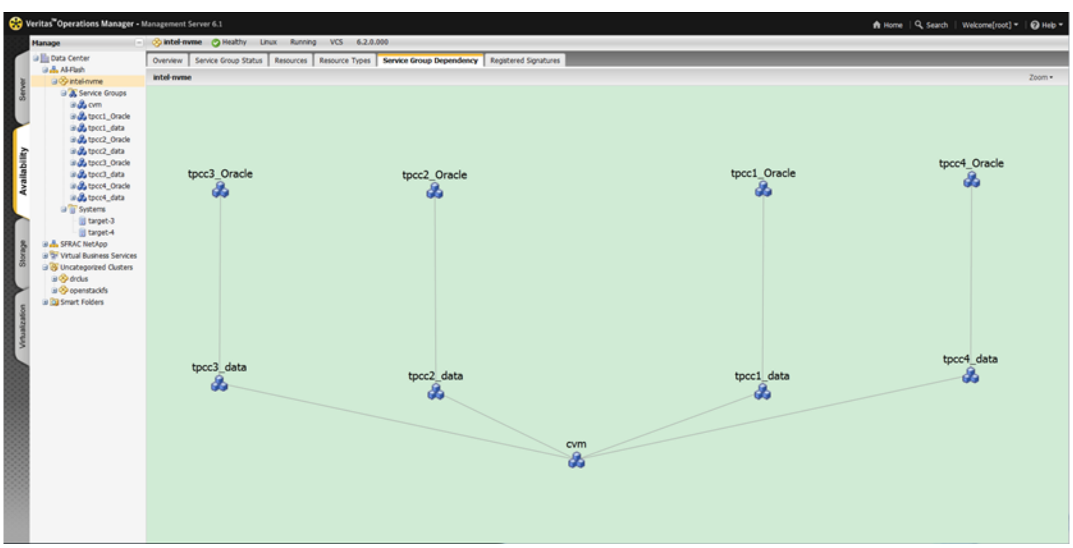
Each Oracle instance was also configured with 23GB of SGA as we will outline in the performance section. Each instance will have a preferred node to run, and this will be specified within the Veritas Cluster Server configuration. In this architecture, upon cluster start, server target-3 will run instances *tpcc1* and *tpcc2*, while server target-4 will run instances *tpcc3* and *tpcc4*. The following figure shows how target4 appears first in the list of *SystemList* for *tpcc3_Oracle* service group.



System List for one instance

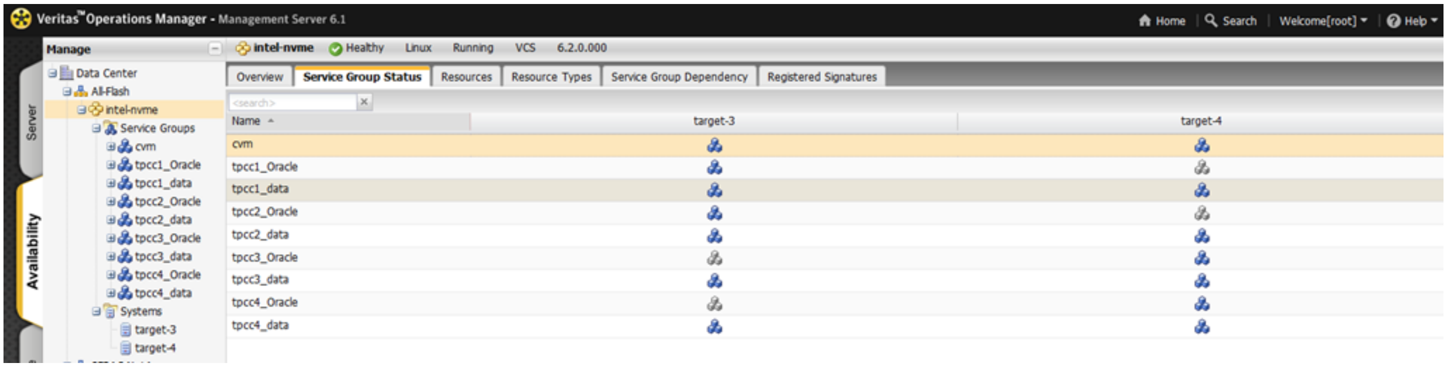
Each instance will be protected by one service group called *tpccX_Oracle*. That will be a failover service group. Each Oracle service group will have a dependency on a parallel service group containing the file systems. Those service groups called *tpccX_data* will keep the file systems needed for each instanced mounted in each node.

The following picture shows the general service group dependencies.



Service groups dependencies

The following view will report what is the status of each of the service groups in each server. We can notice how the file systems are mounted in both nodes, while the Oracle instances only run in one of the nodes.

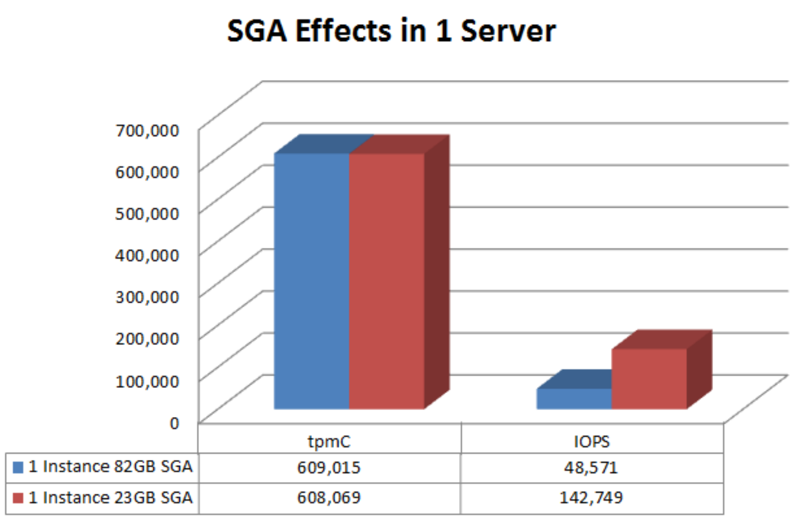


Service group status

Performance Metrics

A TPCC benchmark emulates an OLTP type workload where users are running online transactions. In this section we are going to analyze how many transactions each instance can execute and how we can maximize the global number of transactions per cluster. While the number of transactions is important, we also have to take a look to how the storage subsystem is responding and what are the latencies observed from the application and database (Oracle in this exercise). We are going to compare this at the end with some public benchmarks from all-flash arrays much more expensive than the current solution described in this paper.

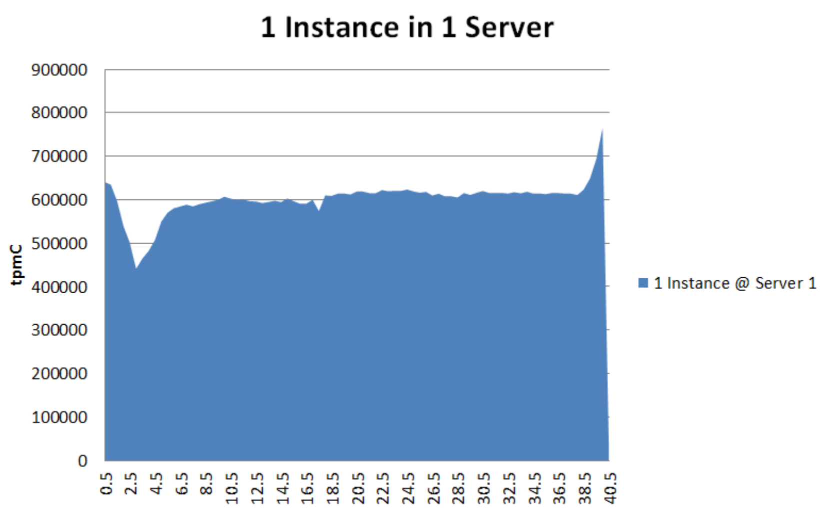
One key thing when running an Oracle benchmark is the size of the SGA. The bigger amount of memory dedicated to Oracle the better performance. It is common to see benchmarks with several hundreds of GB for RAM. In this benchmark, in order to perform as much IO as possible, we are going to limit the SGA per instance to 23GB of RAM. Also, we are going to measure what is the impact of smaller or bigger SGAs in the configuration. The following graph shows the effect of a bigger SGA size for the base test of one instance running in the system:



IOPS depending on SGA size

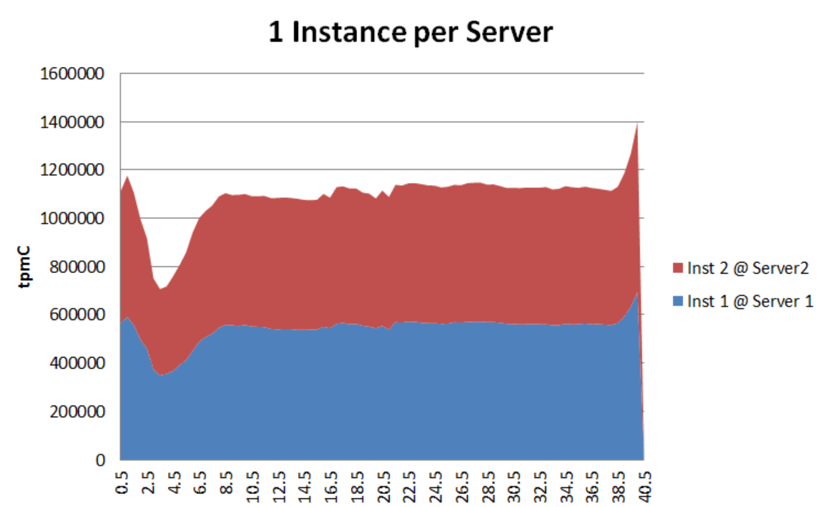
Reducing the SGA from 82GB to 23GB increased the number of IOs issued to the system by almost three times, so the effect of caching can be seen very clearly through the reduced disk I/O. More reads are coming from the SGA, More interestingly is that overall number of transactions remained very similar with a very small loss. While the pressure on the storage did increase, the overall performance remained very stable. This speaks to the high performance capabilities for both reads and writes of the underlying Intel hardware.

In this scenario the average number of transactions per minute achieved was 609,000 for our benchmark.



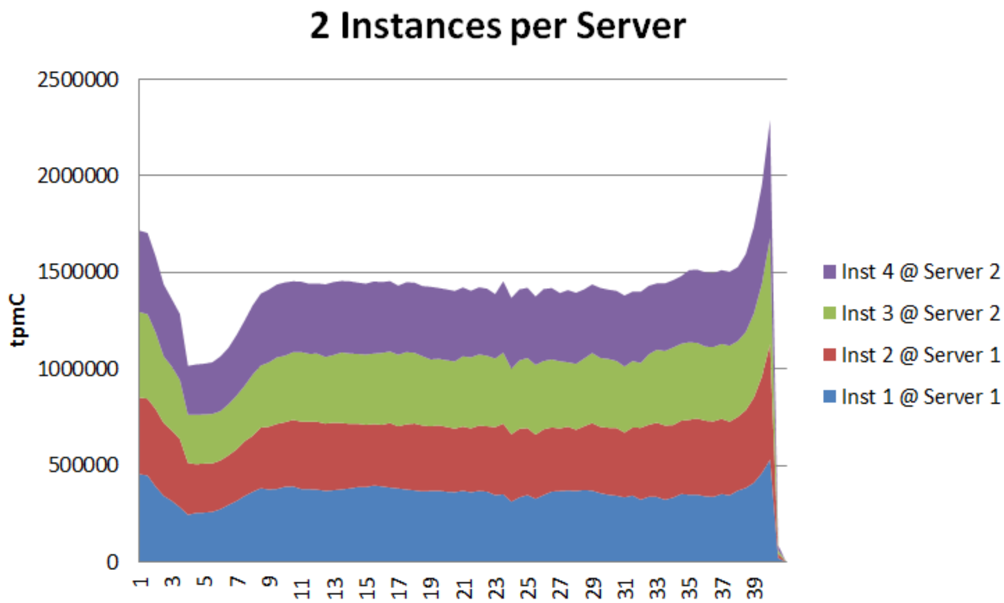
Transactions per minute for one instance

In this case, target-3 is idle outside of receiving the IOs to provide our mirrored copies from target-4. With available CPU, target-3 can also be used to run another instance. This will put double pressure on the storage and we now have two instances sharing the same 4 internal SSD cards. Using Volume Manager (VxVM), we can use software to properly define how the underlying storage to maximize performance. With this scenario, one instance per server, the performance doubles to 1,116,000 transactions per minute.



Transactions per minute when one instance runs in each server

At this point, we still have underutilized CPU, RAM, and IO capabilities across the configuration. This means we can stack two more oracle instances and see the impact on our performance.



Total Transactions per minute when running two instances per node

In Figure 18, we can see that four concurrent instances reached 1,500,000 total transactions per minute, all on the same set-up that we started out with.

These very high numbers in a 2U configuration show how a software solution can maximize the performance of underlying storage structure. The two servers are keeping a high available solution using just four internal SSDs per server and all the data is redundant with zero Recovery Point Objective. Any storage failure will be completely transparent for the database and will be completely unnoticed. A server or application failure will use fast failover technology to recover the service in the surviving node very quickly.

An All Flash Comparison

Today, all Flash Arrays are seen as the optimum in pure performance for high I/O workloads. While the 1.5M transactions per minute figure is impressive, public records show similar numbers have been achieved by all-flash arrays[3]. Digging deeper in these papers show architectures using a dedicated array which contains a bigger number of SSDs and at a much higher cost than the 2U architecture outlined above. Some of those benchmarks even show using up to 10 HDDs dedicated to redo logs[4]. This is due to perceptions and documented best practices that SSD is not optimal for the high sequential writes of redo logs. This is not required in our set-up and, as we achieved the same numbers without it, shows the write capabilities through a combined Symantec and Intel solution. Also, in the scenarios where several instances are run on the same sever, that redo log volume uses the disks that other databases are also using.

Along with TPCC benchmarks, All-flash array vendors make a special note for response times for the I/O. Figure 19 is the histogram output generated by Oracle on the AWR reports enabling a comparison between our configuration and other all flash arrays.

3. https://access.redhat.com/sites/default/files/attachments/red_hat-violin_perf.brief_v5_0.pdf
4. http://c970058.r58.cf2.rackcdn.com/fdr/tpcc/Cisco-Oracle%20C240%20TPC-C%20FDR.pdf

Event	Total Waits	% of Waits <1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<=1s	>1s
[...lines deleted...]									
db file parallel write	666K	98.1	1.6	.3	.0	.0	.0		
db file scattered read	6358	64.9	33.3	1.4	.5				
db file sequential read	299.3	60.3	32.6	6.5	.4	.1	.0	.0	
[...lines deleted...]									
direct path read	71	59.2	32.4	7.0		1.4			
direct path write	67	85.1	11.9	1.5	1.5				
[...lines deleted...]									
log file parallel write	2817.	56.7	37.4	5.5	.3	.0	.0		

Wait Event Histogram for leading all-flash array

This leading flash-array vendor[5] reported the above numbers during a benchmark that generated 83,114 physical reads and 15,754 physical writes per second and a transaction log average of 20.8 MB/s throughput.

Just our single instance run, was generating 39% more IO pressure on the eight SSD cards, with 81,355 physical reads and 56,497 physical writes per second, with an average transaction log of 75.8 MB/s. The histogram for latencies was very similar to the one reported by the all-flash array as we can observe in the following picture.

Event	Total Waits	% of Waits <1ms	<2ms	<4ms	<8ms	<16ms	<32ms	<=1s	>1s
(..lines deleted..)									
db file parallel write	1476K	94.8	2.1	1.6	1.0	.4	.0	.0	
db file scattered read	1326K	88.6	6.2	4.9	.1	.1	.0		
db file sequential read	148M	93.2	3.9	2.8	.1	.1	.0	.0	
(..lines deleted..)									
direct path read	1376	95.8	2.3	1.7	.1	.1	.1		
direct path write	1353	96.2	1.8	1.0	.5	.4	.1		
(..lines deleted..)									
log file parallel write	762K	22.6	23.4	32.7	20.4	.9	.1	.0	

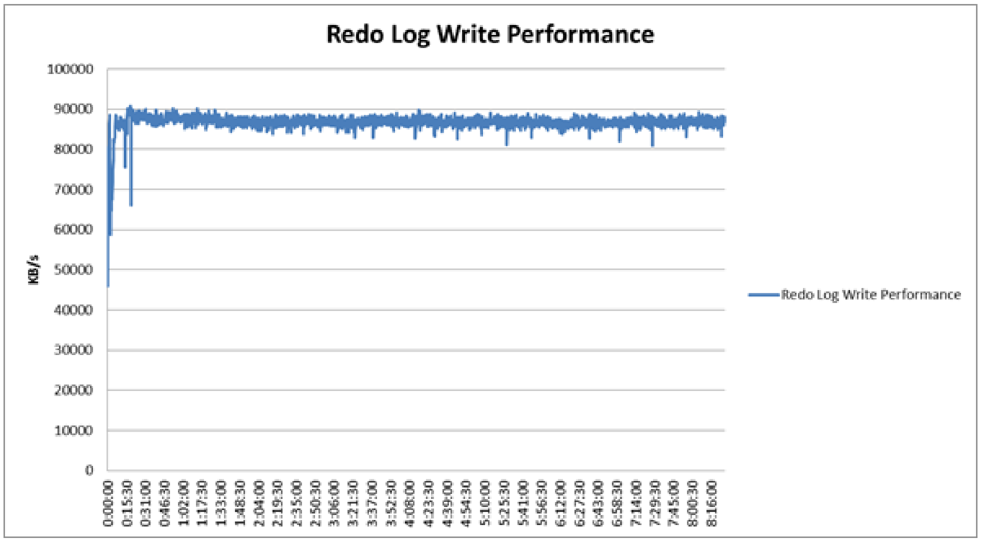
Wait Event Histogram for this White Paper architecture

The log writer latencies are a little bit higher than the all-flash array, but considering there are almost four times I/O on our volume, the normalized latencies are much lower. Every other number within our AWR histograms point to similar or better performance across the board. Again, here we have just four SSDs in one server that are mirrored across the other server four SSDs and the log writer is using only one SSD.

Steady write performance is another key metric when using flash technology. Longer runs illustrate that the performance for both data and redo log volumes can be consistent during larger periods of time.

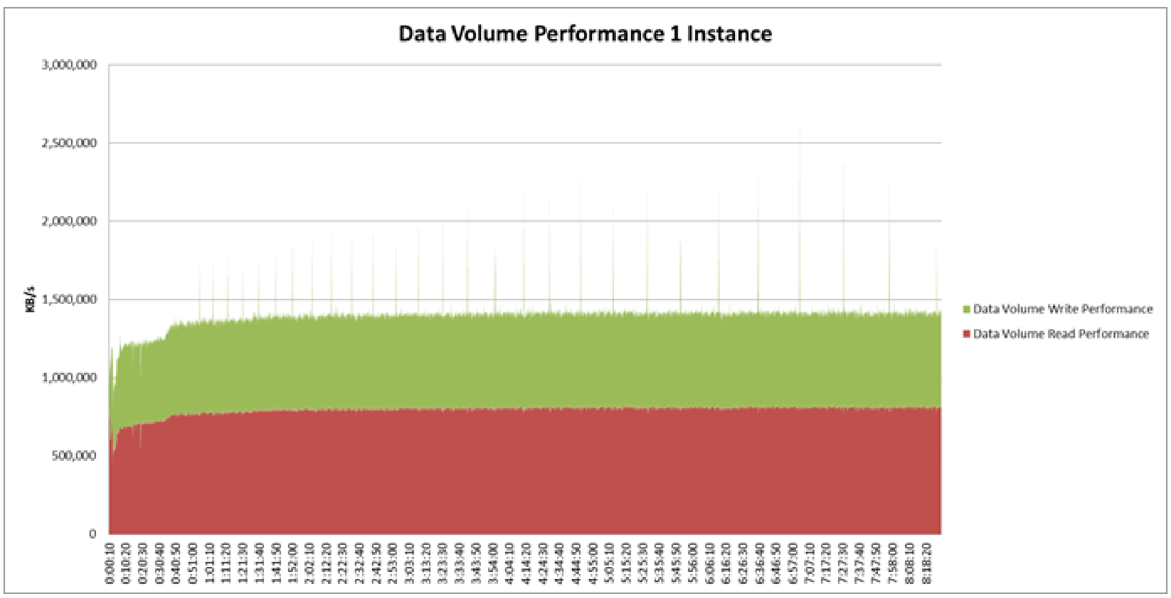
The following picture shows the redo log write performance during 8 hours.

5- http://xtremio.com/wp-content/uploads/2014/07/h13174-wp-optimized-flash-storage-for-oracle-databases.pdf



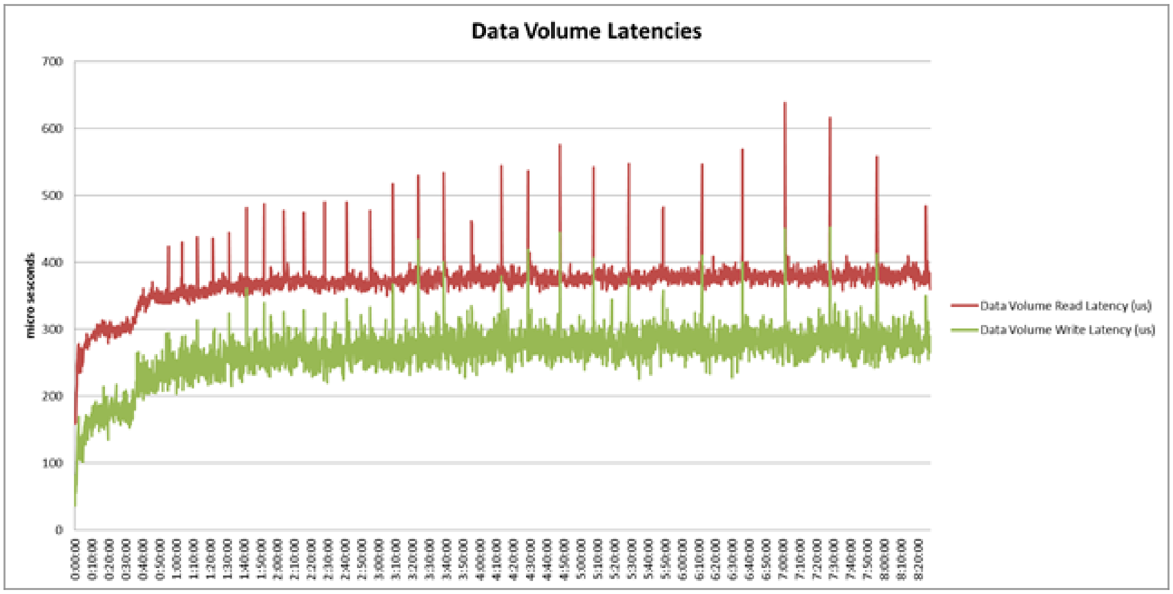
Redo log volume writes performance

The data volume with just one single instance is providing close to 1.5GB/s steady throughput as shown in the following picture.



Response times for data volume

And the response time for the volumes is always in micro seconds units during the 8 hours interval.



Data volume latencies

Fast Failover

Symantec Cluster Server's Oracle agent uses Intelligent Monitoring Framework to detect any instance failure, triggering the instance recovery very quickly, without having to wait for a traditional monitor interval. In case that the instance needs to be restarted in the other node, because Flexible Storage Sharing uses Cluster Volume Manager underneath, the volumes and file systems are accessible by all the nodes at the same time. Therefore, the time needed to recover the database is limited to the time needed to restart and recover the instance plus a minimal detection time.

In order to proof this capability in this high performance demand environment, we have tested how much time is needed for one instance to be recovered in the other node once Oracle fails and the database is running at full transactions capacity. To do that, we have killed one Oracle process and measure the time needed until the database is recovered in the other node, taking also note on how much data had to be recovered from redo log.

	RUN1	RUN2	RUN3	RUN4
Oracle starting in the other node	0:00:18	0:00:17	0:00:16	0:00:21
Database mounted	0:00:07	0:00:01	0:00:06	0:00:06
Crash recovery	0:02:10	0:02:09	0:00:50	0:02:06
Database online	0:00:04	0:00:02	0:00:02	0:00:02
Total Recovery Time	**0:02:39**	**0:02:29**	**0:01:14**	**0:02:35**
Read KB redo	15,970,970	15,984,448	5,654,785	15,854,764
Blocks need recovery	1,031,287	1,015,626	985,103	1,026,460
Completed redo application (MB)	2,657	2,276	1,033	1,726

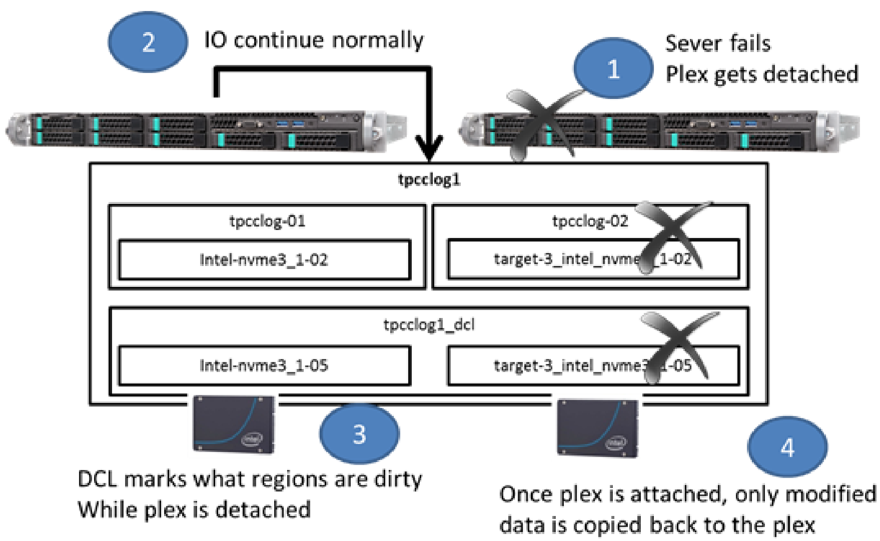
Instance recovery times

Storage Resiliency

As described in the previous sections, the architecture used in this white paper provides high availability for the application and for the storage. Any storage failure is transparent to the application and it does not trigger any IO failure. Volume manager deals with any storage or server failure and make it transparent to the application. In this scenario, each volume will be composed by two plexes, and each plex will reside in a different server. If a server goes down, the plex coming from that server will be disabled, and the volume will continue working normally with the other plex. Any disk failure will only affect the plex where the disk resides. Data will be always synchronized between the two plexes.

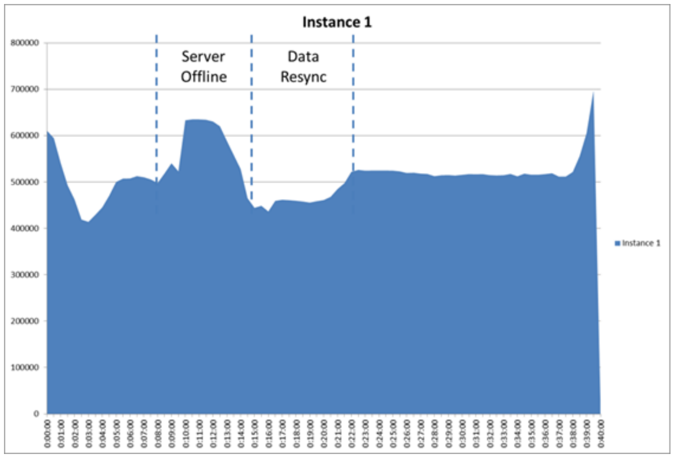
When any of the plexes is disabled, volume manager will use the Fast Mirror Resync feature to enable a fast resynchronization once the plex is back. Each volume will use a DCO (Data Change Object) where volume manager will note what blocks are dirty and need to be copied to the other plex once it is back.

The following picture represents graphically the configuration of the *tpcclog1* volume and outlines the steps when there is a failure in one of the plexes.



Volume Layout and Plex Failure

As one example, when a server is down, the plexes from that server will get disabled and IO will continue working normally using the other plex. In the configuration proposed here, when a server is down, IOs will not be shipped to the other server, and in fact, volume performance will increase because it will only have to make local writes and note what blocks are dirty, not having to ship IO to the other node. The following figure shows the performance for an instance when the other server is rebooted. First it can be observed how the performance increase given the other server is offline. Once the server is back there is a small degradation while dirty blocks are resynchronized and then performance goes back to normal.



Server offline scenario

Cost Analysis

All-flash and engineering solutions can be cost prohibited. The array used on (2) is listed on $540K MSRP price and it will offer same raw capacity that the solution described in this white paper using 2TB SSDs (a total 16TB capacity with 8TB usable). This is just storage and does not include any server or software license. Other vendors have similar prices with even higher costs for engineered solutions.

The converged infrastructure presented in this white paper is estimated in a MSRP price of $169K, including HW and Symantec software. The following table shows some metrics from three different white papers from all-flash vendors. This white paper configuration achieved similar numbers in terms of TPmC that configuration proposed on (1) with a clearly lower costs. Configuration (2) is clearly superior in number of TPmC given that the number is for just one instance, but clearly the HW resources used are much bigger. Comparison with configuration described on (3) is interesting in terms of IOPS issued to the HW and the latencies showed from the storage system.

Configuration	Violin (1)	Cisco (2)	XtremIO (3)	Symantec/Intel	XtremIO (3)	Symantec/Intel	XtremIO (8)	Symantec/Intel
Storage HW	Violin 6616	Violin V6000 VTrack J630sS JBOD	Single X-Brick	**8 x Intel P3700**	Dual X-Brick	**8 x Intel P3700**	Dual X-Brick	**8 x Intel P3700**
Capacity	8TB (6)	32TB All-Flash + 12TB HDD	10TB (7)	**8TB (5)**	20TB (7)	**8TB (5)**	14.94 TB (6)	**8TB (5)**
System RAM	128GB	768 GB	----	**256GB (4)**	----	**256GB (4)**	2TB (9)	**256GB (4)**
Fibre Channel Paths	6	16	4	**N/A**	4	**N/A**	----	**N/A**
Storage Costs (MSRP)	540,000	1,098,460	340,000	**56,656**	680,000	**56,656**	680,000	**56,656**
Oracle Instances	4	1	1	**1**	2 (RAC)	**2**	4 (RAC)	**4**
Highly Available Config	NO	Yes	NO	**Yes**	Yes	**Yes**	Yes	**Yes**
TPmC	1,500,000	1,609,186	----	**608,069**	----	**1,116,069**	----	**1,427,477**
IOPs	----	----	98,868	**137,852**	189,528	**256,052**	182,041	**327,614**
DB sequential reads latency (ms)	----	----	1.07	**0.377**	0.961	**0.626**	1.747	**1.423**
DB parallel read (ms)	----	----	----	**0.862**	----	**1.578**	4.298	**2.876**
Redo Log Thoughput (MB/S)	----	----	20.8	**75.8**	41.4	**142**	28	**187.6**
DB writer response time (ms)	----	----	----	**0.267**	----	**0.305**	0.691	**0.424**

1. https://access.redhat.com/sites/default/files/attachments/red_hat-violin_perf.brief_v5_0.pdf
2. https://c970058.r58.cf2.rackcdn.com/fdr/tpcc/Cisco-Oracle%20C240%20TPC-C%20FDR.pdf
3. http://www.xtremio.com/wp-content/uploads/2014/07/h13174-wp-optimized-flash-storage-for-oracle-databases.pdf
4. 128 GB per system - 23 GB SGA per Instance
5. IOPS using 3.2 Usable TB capacity. Prices for 16 TB raw capacity (8 TB usable capacity in a high resilience configuration)
6. Usable capacity
7. Raw capacity
8. https://www.emc.com/collateral/white-paper/h12117-high-performance-oracle-xtremio-wp.pdf
   (OLTP 75% / 25% Query / Update Table 14 - 128 Sessions)
9. 4 Node configuration with 512 GB each node (128 Gb for each VM - SGA 16 GB)

This represents an important reduction in TCO achieved by a software defined storage solution created using Symantec software on a cluster of standard servers that leverages high performance server-side storage devices, while providing shared storage capabilities.

Conclusion

As enterprises data centers continue their quests to provide a high level of service for their internal customers and applications, there is the constant underlying goal to do so at reduced complexity and cost. However, meeting those goals without sacrificing performance or availability has been elusive. Converged architectures based on flash arrays will provide high performance with simple install, but also at a high cost. Scale-out software defined storage solutions can utilize DAS for reduced cost, but lack built-in data and application management functionality leading to complex point solution.

Symantec Cluster File System provides a full suite of data and application availability tools in a single solution to bring performance, availability, and flexibility to the data center. Combined with unmatched read and write capabilities from Intel solid state devices, enterprises have the ability to achieve those results at a much lower cost than the alternatives.

More Information

For additional details, please reach out to Carlos Carrero at *carlos_carrero@symantec.com*

About Symantec

Symantec Corporation (NASDAQ: SYMC) is an information protection expert that helps people, businesses, and governments seeking the freedom to unlock the opportunities technology brings—anytime, anywhere. Founded in April 1982, Symantec, a Fortune 500 company operating one of the largest global data intelligence networks, has provided leading security, backup, and availability solutions for where vital information is stored, accessed, and shared. The company's more than 20,000 employees reside in more than 50 countries. Ninety-nine percent of Fortune 500 companies are Symantec customers. In fiscal 2014, it recorded revenue of $6.7 billion. To learn more go to www.symantec.com or connect with Symantec at: go.symantec.com/socialmedia.

For specific country offices and contact numbers, please visit our website.

Symantec World Headquarters
350 Ellis St.
Mountain View, CA 94043 USA
+1 (650) 527 8000
1 (800) 721 3934
www.symantec.com