Evaluation of the New Cray Sonexion 1300

Hussein N. El-Harake and Thomas Schoenemeyer
Swiss National Supercomputing Centre (CSCS), Lugano, Switzerland
hussein@cscs.ch, schoenemeyer@cscs.ch

Abstract: During SC11 Cray announced a new innovative HPC data storage product named Cray Sonexion. CSCS installed an early Sonexion system in December 2011; this system is connected to a Cray XE6 test and development machine. The purpose of this study is to evaluate the mentioned product, covering installation, configuration and tuning including the Lustre file-system and integrating it with the CRAY XE6. We describe the hardware, infrastructure, software stack, Lustre filesystem and benchmarks. IOR, IOzone, mdtest and obdfilter_survey were used for benchmarking. By way of comparison the Sonexion storage was also connected to a small X86-64 cluster. The Cray XE6 (Gele) has four compute blades (AMD Interlagos) and four service blades while the X86 cluster has four nodes, two are based on Intel Westmere and two on AMD Interlagos.

1 Introduction

Scaling of storage to sustain the performance of HPC applications on the next-generation platforms will be a significant challenge. Such platforms are capable of delivering immense amounts of data; computing should have access to adequate I/O that is capable of delivering fast access to HPC file-systems with low latency and high bandwidth.

The Sonexion 1300 Data Storage is the first generation of the Cray Sonexion family. It is a complete high performance, reliable and scalable HPC solution. Sonexion benefits from Lustre integration to scale at any level. Cray delivers different layouts for scaling requirements; it could be as small as one SSU (Scalable Storage Unit) all the way to a total of 180 SSU units. The Metadata Management Unit (MMU) consists of two I/O servers and a 2U24 (2 Units 24 2.5" Drives) JBOD. The JBOD has 22 drives for MDS and MGS raids and 2 100GB SSDs for the metadata journal. Aggregating MMUs is possible; up to three MMUs could be aggregated as one system.

Cray Sonexion 1300 SSU

From the Cray Sonexion datasheet [1], a standalone 1300 should be able to deliver 3GB/s using IOR, in read and write throughput. The system should scale linearly, so a small configuration mode with 3SSUs should deliver 9GB/s [1]. CSCS installed a single SSU system and went through different benchmark scenarios showing results and comparing it to what Cray announced.

2 Benchmark Tools

Obdfilter_survey comes with Lustre [2]; it measures the performance of one or more OSTs directly on the OSS node or alternately over the network from Lustre clients.

IOR [3] is a powerful open source benchmark, specifically designed to benchmark parallel file systems (GPFS, Lustre etc.). IOR offers several interfaces such as MPI-IO, HDF5 and POSIX. We used it from the Cray compute nodes only.

Mdtest [4] is an MPI-coordinated metadata benchmark test that performs open, stat and close operations on files and directories and reports the performance.

IOzone [5] is a trusted filesystem benchmark tool; it measures different file operations and has been ported to different platforms. We used it on the Cray service nodes (LNET routers) and the X86 cluster.

3 Methods

All experiments described below were performed using two different versions of Lustre clients, version 1.8.1 on the XE6 and version 2.1.1 on the X86 cluster. Lustre 2.1.1 is used on the Sonexion system; this version came with the Sonexion software stack 1.0.1 release. We used Obdfilter_survey to run the first test from the OSSs and results showed the peak performance. Running Obdfilter_survey is somehow similar to running a raw test on standard RAIDS or disks, and hence such tests will help understanding the performance capacity of any system. The results are summarized in Table 1.

Lustre results from the OSSs using obdfilter-survey					
Results are in MB					
Tool	Nodes	size MB	Threads	write	read
obdfilter-survey	OSS 1	16384	64	1899	2300
	OSS 2			1858	2301
Total				3757	4601

Table 1: Obdfilter delivered 3.66GB/s in write and 4.49GB/s in read

4 Results

4.1 Bandwidth measurements

From the results of the Obdfilter test the system showed interesting numbers compared to what Cray announced. More comprehensive testing with IOzone produced equally interesting results presented in Figures 1 and 2

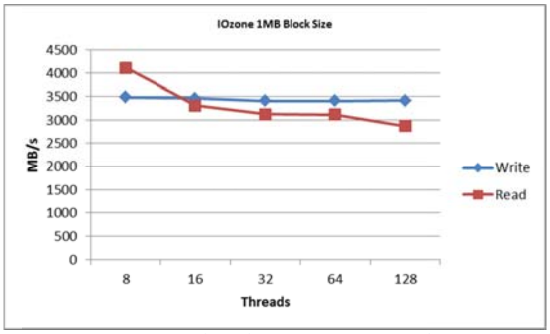

Figure 1: IOzone throughput on Cray Service nodes

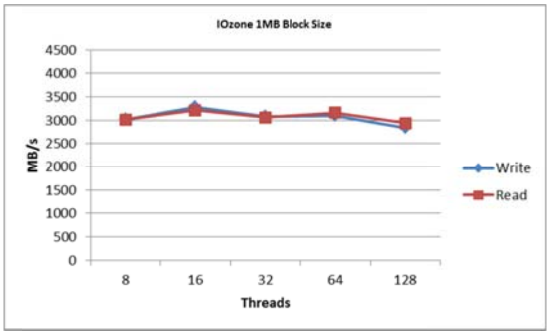

Figure 2: IOzone throughput on X86 Cluster

On the XE6 service nodes IOzone showed slightly better performance than the X86 cluster; read performance dropped in scaling while write reminded constant. On the X86 cluster performances showed a decrease in both read and write performance by ~10% in scaling up to 128 threads.

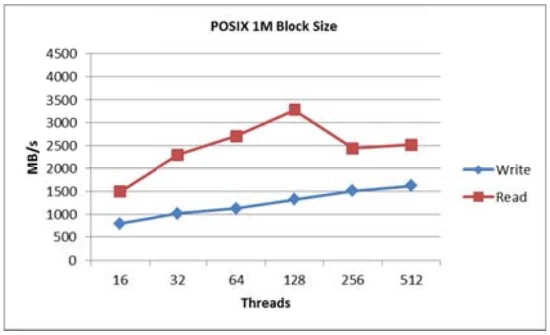

Figure 3: IOR throughput on Cray compute nodes POSIX 1MB

Using the POSIX interface of IOR with a 1MB block size on the XE6 compute nodes had a significant difference when compare to MPIIO and POSIX with 4MB block size. As shown in Figure 3 write performance started at 900MB with 16 threads and increased to 1.6GB with 512 threads, while read reaches 3.3GB with

128 threads, while at 512 threads read performance decreased to 2.5GB/s.

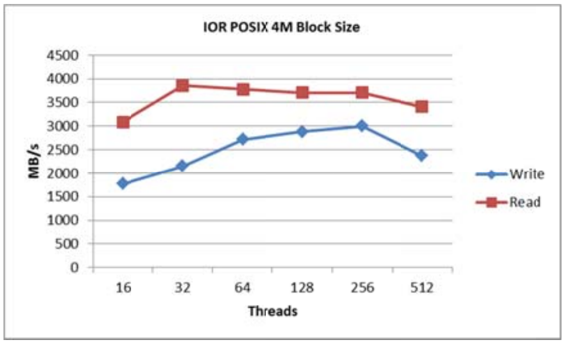

Figure 4: IOR throughput on Cray compute nodes POSIX 4MB

Figure 4 shows that a 4MB block size with the IOR POSIX interface had significant improvements in read and write comparing to 1MB block size. Read numbers reached 3.8GB/s and kept good performance at 3.4GB/s on 512 threads, while write had an average of 2.5GB/s.

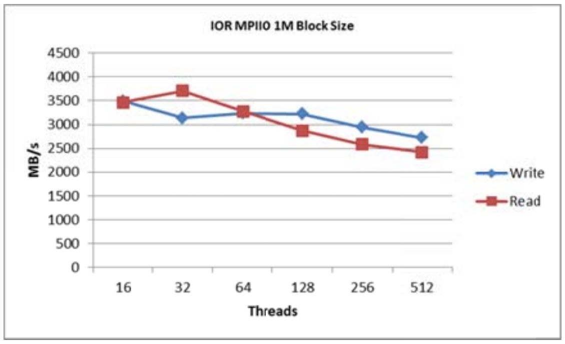

Figure 5: IOR throughput on Cray compute nodes using MPIIO 1MB

In Figure 5 we see that Running IOR with MPIIO showed somewhat different results than the POSIX interface. In this test Lustre stripe was set to 1 so that every thread will create a file on every OST. The best numbers in read were at 32 threads, 3.6GB/s while 16 threads in write reached 3.5GB/s. A gradual drop appeared while scaling until we reached 2.6GB/s at 512 threads.

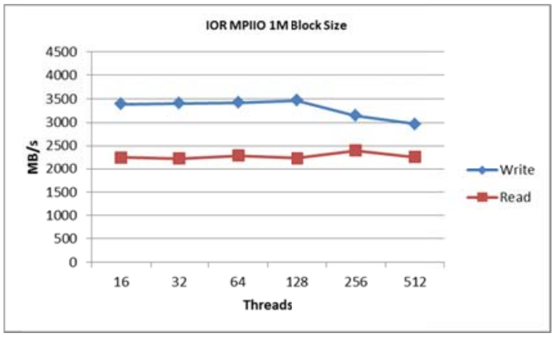

Figure 6: IOR throughput on Cray compute nodes MPIIO 1MB

Figure 6 summarizes the second IOR test. We changed the Lustre stripe to 8 (count equal to the total number of OSTs). In this test every run creates a single file which will be striped across all OSTs. Write results showed ~3.5GB/s which performed better than read by 30 %, in total we had an average of 2.4GB/s in read.

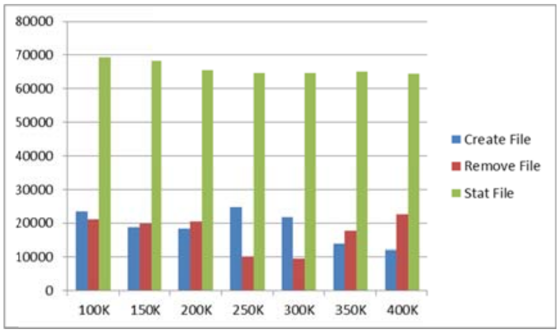

Figure 7: mdtest results on the XE6 compute nodes

Figure 7 shows the mdtest results, where we measured performance of 512 tasks creating, removing and stating files. Every task creates its own files in one directory. From the results we see that number of files has an impact on results, an average of 20 KOP/s in file creation and 17KOP/s in file deletion, while state was not affected and performance was around 65KOP/s.

5 Parameters and Lustre tuning

5.1 Lustre tuning:

Maximum the number of concurrent RPCs in flight, this will improve performance of data and metadata, clients only; **echo 32 > /proc/fs/lustre /osc/*/max_rpcs_in_flight**

Check the status of checksums: **lctl get_param osc.*.checksums** to disable checksums: **echo 0 > /proc/fs/lustre /osc/*/checksums**

By default lustre has 32MB of memory cache for every OST; this number could be increased to 256 MB to set the new memory parameters: **lctl set_param osc.*.max_dirty_mb= 256**

By default debug is ON that could affect performance. It's possible to reduce the debug level or simply turn it off completely. To turn debug OFF: **sysctl -w lnet.debug=0**

5.2 IOR and IOzone parameters:

```
aprun -n $N –N $N  IOR -a MPIIO -F -E -k -b $N G -i 3 -t 1M -B -v -C –o $D
```

```
aprun -n $N –N $N  IOR -a POSIX -F -E -k -b 20G -i 3 -t 1M/4M -B -v -C –o $D
```

IOzone parameters: iozone -i 0 -i 1 –r 1M -s $N g -+u -c -C -+m file -t $N

5.3 mdtest parameters:

```
aprun -n $N –N $N mdtest -n $N -u -i 3 -N 1 -d $DIRECTORY
```

6 Conclusion

We evaluated the Sonexion system with all its components, controlling the basic infrastructure, software stack and running some benchmark using different tools.

Our observations are comparable with the specifications provided by the Cray datasheet; even in some cases we got better sustained performance figures

We believe that there are several steps that could be taken to improve the system functionality and performance. An example is replacing SAS disks with SSDs for Metadata, as reported above SSD drives are currently only used for metadata journaling.

The system arrived with experimental beta software that was unusable. The second release of the experimental software arrived in early 2012, which significantly improved the usability of the system.

With the new software stack we also noted significant improvements in performance, the functionally of the available tools, new features and new documentation. The new software stack also seated a new Lustre release.

7 References

[1] Cray Sonexion: 1300 storage Data sheet http://www.cray.com/Assets/PDF/products/sonexion/Sonexion1300Datasheet.pdf

[2] Lustre HPC Parallel File System http://wiki.lustre.org/index.php/Main_Page http://www.whamcloud.com/

[3] IOR HPC File-System Benchmark tool. http://sourceforge.net/projects/ior-sio/

[4] Mdtest MPI metadata benchmark. http://sourceforge.net/projects/mdtest/

[5] IOzone file-system performance benchmark utility, www.iozone.org