Detailed Performance Analysis of Solid State Disks

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

Abstract – We evaluated two different PCIe attached SSD devices with various methods for analyzing bandwidth and IOPs. The performance over SATA drives is huge as expected, by a factor of 20 in bandwidth and a factor of 1500 for IOPs. The results indicate, that SSDs are a valid alternative for SATA or SAS drives in terms of reliability, performance and field replaceability. The observed IOPS and bandwidth results encourage us to continue the evaluation of these devices such as the deployment in GPFS metadata servers.

1. Introduction

High sustained application performance on next-generation platforms will not be dependent only on performance of processors, future platforms must also provide access to reasonable I/O capabilities. Applications such as computational fluid dynamics, combustion, astrophysics, and climate research, will need fast access to extremely fast file systems with low latency and high bandwidth.

In existing HPC architectures, the storage hierarchy has a several order of magnitudes latency gap between the main memory and spinning disk and evolves as a major obstacle for data-intensive computing (Figure 1). Adding more and more spindles to existing parallel file systems for HPC architectures helps in increasing the aggregated I/O bandwidth, however the latency problem remains unsolved.

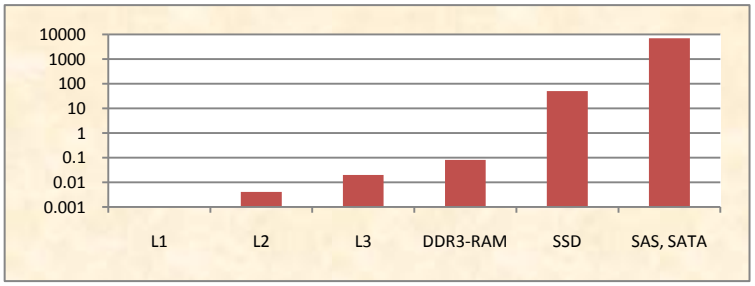

Figure 1: Typical latency in µs for several storage devices

Flash-based Solid State Disks (SSD) are a promising technology [1] to solve the performance gap in the next-generation systems [7], and recent advances in solid-state memory technologies seem to emerge as an interesting alternative to this problem. A large number of companies, ranging from non-volatile memory manufacturers to server OEMs are offering NAND Flash-based SSDs in various form factors. However, the typical variation under realistic workloads is sometimes disappointing as shown in figure 2. To address this problem, we decided to select two types of enterprise class NAND Flash devices to study the differences in the performance characteristics [3] depending on the specific conditions and workload.

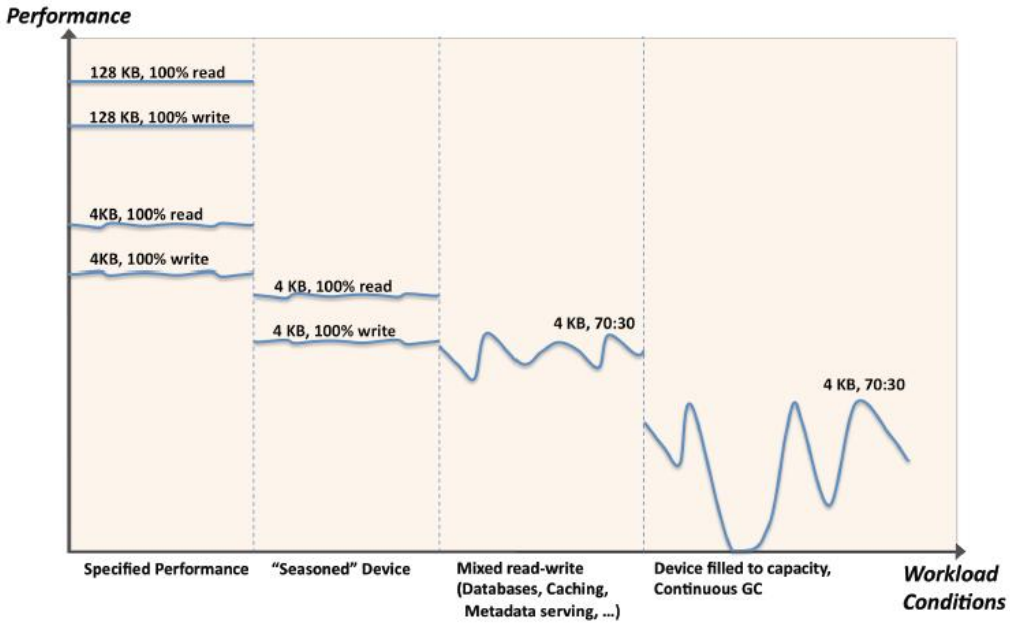

Figure 2: Performance variations in current generation of NAND Flash---based SSDs [taken from 1]

This research is motivated by mainly two reasons. In the current concept of parallel file systems, the metadata servers are a potential bottleneck for the aggregated I/O bandwidth of a cluster caused by latency and bandwidth limitations of the attached disks.

CSCS uses two file systems, GPFS as global parallel file system and Lustre for the HPC systems as local scratch such as the Cray XT5. Lustre still has only one metadata instance and all system operations are concentrated on this instance. GPFS metadata server can be scaled, however the current concept of using FC drives in those servers will cause limitations too. Therefore it is in the main interest of CSCS to optimize the metadata operations in the parallel file systems. The integration of SSD´s into GPFS as well as the combination of rotating and solid state drives is not investigated so far.

Second, for certain application workflows it will be useful to replace local SATA disks by SSD as direct attach storage. If an application doesn´t need more than the capacity of a few SSD cards, this could be a very efficient way to speed up the elapsed time of applications, especially if low latency is required. Even the bandwidth per price could be less compared to a parallel file system. An example of such a configuration is explained in detail in [4].

Related to application performance, it is also important to invest in high-performance parallel I/O libraries or domain-specific IO Libraries to be included in the application. CSCS is doing research in this area together with domain experts at several institutes in Switzerland and US. However, from an I/O perspective, the underlying storage hardware either used as local scratch or as global parallel file system, will remain as a bottleneck for I/O bandwidth.

2. Methods

We used a dual-socket Server from Supermicro with AMD Opteron 6128 processor (Magny-Cours) with 8 cores and 2 GHz frequency and 24 GB Main Memory. The server has got one x8 and two x16 PCIe Gen2 slots.

a. Tested devices

Two enterprise class flash storage devices were evaluated. The first device was the Virident TachIOn SSD card with a usable capacity of 400 GB of SLC NAND Flash. It uses 16 replaceable Flash modules allowing on-site module replacement. It supports various file systems like EXT4, XFS, Lustre and GPFS and needs a single-slot PCIe 8x Gen 1 or PCIe 4x Gen 2 , 25 watt card in a low-profile, half-height and half-length form factor.

The datasheet promises a maximum read performance of 1.44 GB/s and a maximum write performance of 1.20 GB/s. The peak IOPs is advertised around 300,000 IOPS at 4K block size. For the data reliability it uses an advanced, end-to-end error correction. The TachIOn software supports software RAID 1 /-1 and protects against single card failures. It also provides Flash-aware RAID (3+1/7+1) and protects against complete NAND failures, if one uses one NAND module per RAID group. It uses 8-bit BCH ECC against NAND errors. The card was delivered with a driver version 1.1 for AMD motherboards. We tested the device with RAID5 (3+1).

Figure 3: Virident tachIOn N300 device

The second device was a FusionIO ioDrive Duo with 320GB of SLC NAND flash and a high-profile PCI-Express Gen2 x4 connection. It uses flash Raid and it has one SSD as reserved in case of SSD failure. This SSD device should deliver according to the brochure [2] up to 1.5 GB/s read/write bandwidth and about 260,000 IOPS in read and write. The FusionIO card offers a RAID 5-like spare parity chip. In the event of a chip failure, Flashback detects the error, and re-routes the data-path to a parity chip held in reserve. Then, it evaluates whether the failed chip can be brought back online (soft failure), or whether it must be taken out of service (wear-out). If there was only a soft failure, the chip is healed, and brought back into service, with no loss of capacity or processing capabilities. When a

permanent failure occurs, the chip is taken out of service. The card was delivered with a driver version 2.3.0 for AMD motherboards.

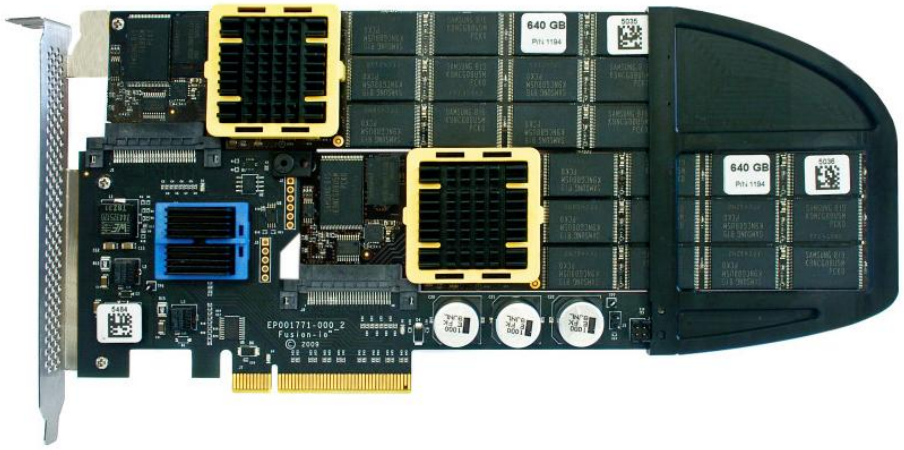

Figure 4: FusionIO ioDrive Duo

b. Experiments

To evaluate the characteristics of the two devices, we considered bandwidth and IOPS of the device with two open-source tools. We used a dual-socket server running a XFS file system.

FIO

fio was created to allow benchmarking specific disk IO workloads. It can issue its IO requests using one of many synchronous and asynchronous IO APIs, and can also use various APIs which allow many IO requests to be issued with a single API call. You can also tune how large the files fio uses are, at what offsets in those files IO is to happen at, how much delay if any there is between issuing IO requests, and what if any file system sync calls are issued between each IO request. A sync call tells the operating system to make sure that any information that is cached in memory has been saved to disk and can thus introduce a significant delay. The options to fio allow you to issue very precisely defined IO patterns and see how long it takes your disk subsystem to complete these tasks. For iodepth we used 256 because that provided the best results, especially for IOPs.

```
Random read Throughput
----------------------------------
fio --name=job --size=4G --directory=/filesystem --numjobs=8 --filesize=4G  --bs=1M --iodepth=256 --
iodepth_batch_complete=8 --iodepth_batch_submit=8 --ioengine=libaio --direct=1 --norandommap --rw=randread --
group_reporting

Random read IOPs
------------------------
fio --name=job --size=4G --directory=/filesystem --numjobs=8 --filesize=4G  --bs=4k --iodepth=256 --
iodepth_batch_complete=8 --iodepth_batch_submit=8 --ioengine=libaio --direct=1 --norandommap --rw=randread --
group_reporting
```

Commands used for the FIO test

IOR is a powerful open source benchmark tool was made to benchmark parallel file systems (GPFS, Lustre etc.). IOR offers several interfaces such as MPI-IO, HDF5 and POSIX.

In this study we used POSIX. Some of the important parameters used: bypass any I/O buffers, size of each file written by a task, block size, number of segments and how many files are written for each task.

For the experiment we used the commands below by adding mpirun –np xx, where xx is the number of threads. We measured 4, 8, 16, 32, 64 and 96. The best results were achieved with 64 threads and are discussed in the results section.

```
Read & Write IOPS
------------------------
IOR -a POSIX -B -b 4G -t 4K -k -r -w -e -g -s 1 -i 1 -F -C -o /directory

Read & Write Throughput
-----------------------------------
IOR -a POSIX -B -b 4G -t 1M -k -r -w -e -g -s 1 -i 1 -F -C -o /directory
```

Commands used for the IOR test

3. Results

a. Bandwidth Measurements

We first discuss the bandwidth measurements received from FIO and IOR for both devices. Both types of devices are capable to deliver 1GB/s bandwidth for a large enough block sizes (>16K).

The Virident TachIOn device performed better using FIO benchmark compared to the IOR as shown in figure 5. We measured a peak bandwidth of 1.2 GB/s for random-write and a lower random read performance of nearly 1 GB/s at block sizes of 64K. At a block size of 4K the performance degrades by up to 50%. For larger block sizes we observed a degradation of 10%. The peak transfer rate for IOR throughput (figure 6) is quite similar at large block sizes, but we have measured a strong degradation for the random write rate at small IO-blocks to 20% of the peak rate. This is clearly below the promised peak performance in the datasheet. We discussed the results with Virident. The problem could be reproduced by them and it was concluded to update the AMD driver for the TachIOn card. The new revision will appear in June 2011.

Figures 7 and 8 show the results for the same experiments with the FusionIO cards. The results obtained for small blocks are similar, except the poor IOR random write we have observed on Virident. Typically the values vary between 400 and 600 MB/s when writing 4K block size. For larger blocks the FusionIO ioDrive behaves different. The performance increases much faster at higher block sizes. With 16K the FusionIO card outperforms the peak performance measured on the Virident card. The maximum bandwidth is obtained at 1MB block size and the write bandwidth is close to 1.4 GB/s. The read bandwidth is 10% better than what is promised in the product brief. In contrast to the Virident card, we observed for block sizes >16K a write performance which is 15% less than the read performance.

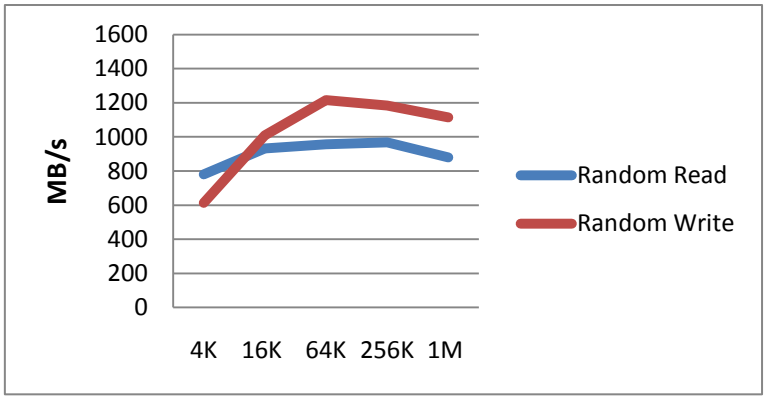

Figure 5: FIO Random Throughput on Virident TachIOn Card

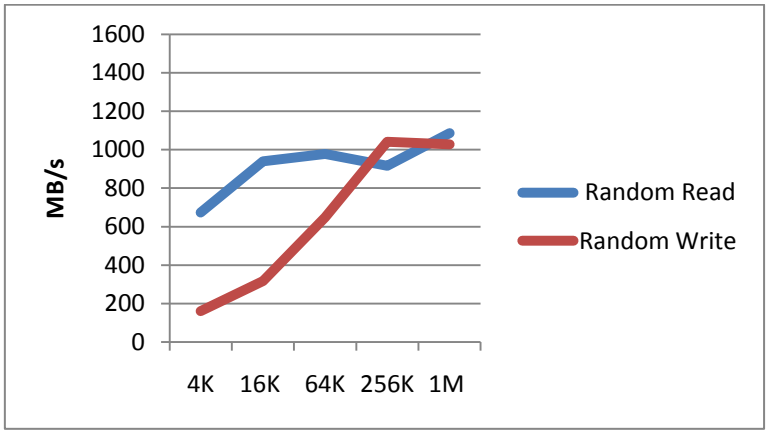

Figure 6: IOR Random Throughput on Virident TachIOn Card

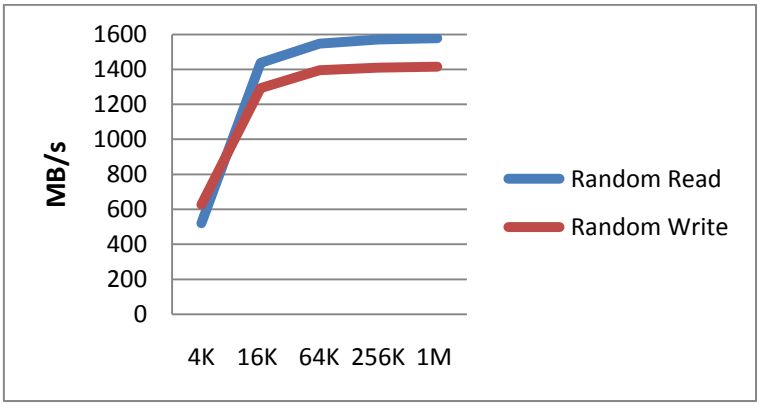

Figure 7: FIO Random Throughput on FusionIO Card

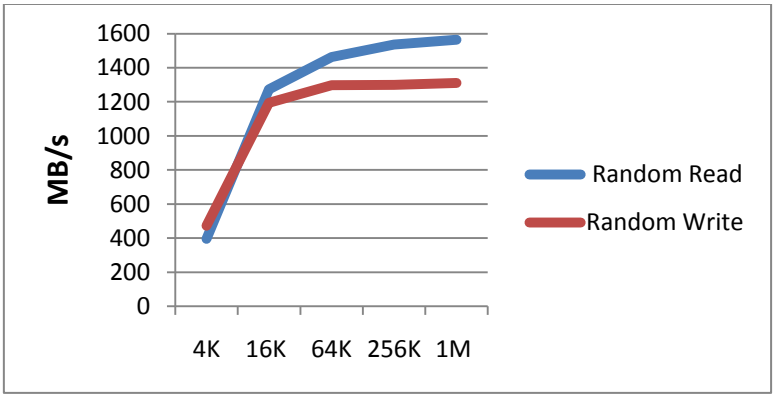

Figure 8: IOR Random Throughput on FusionIO Card

b. IOPS measurements

Both devices are advertised to reach more than 250K IOPS for small block sizes.

We could not reproduce this value during our tests with any of the devices. The highest IOPS number we observed for the FIO benchmark (Figure 9) at 200K random read and only 160K random write for the FIO Random on the Virident TachIOn card. As expected, the IOPs rate drops to a few hundred IOPs at 1M block size.

The IOR random write IOPs peak value was slightly higher with 172K IOPS, however for the same benchmark the random reads reached only 41K (Figure 10). This behavior was discussed with Virident and Virident engineers could reproduce the poor read performance at their lab. Note, the problem is only observed on AMD-based servers.

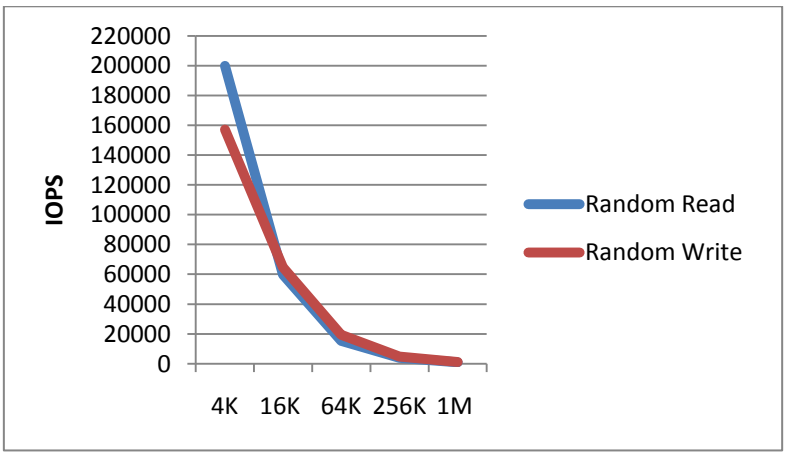

Figure 9: FIO Random IOPS on Virident TachIOn Card

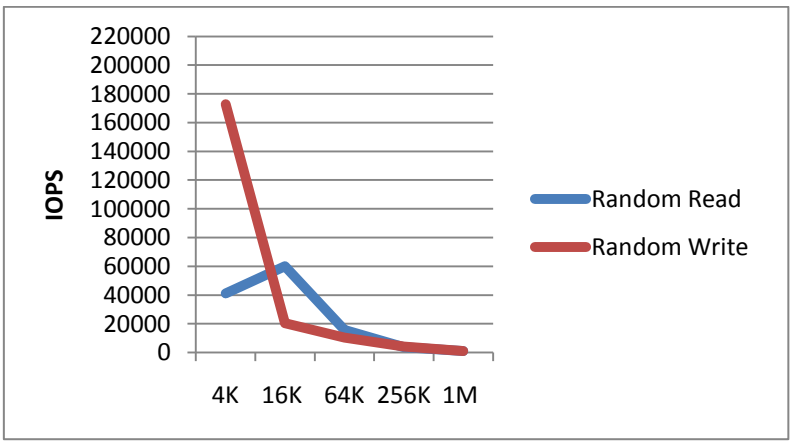

Figure 10: IOR Random IOPS on Virident TachIOn Card

The FusionIO card showed in both experiments (figures 11 and 12) a very consistent behavior. The peak values of the FusionIO card at 4K were below the results of the Virident card, however the performance degradation with higher block sizes is significantly lower than for the Virident card.

In contrast to the Virident card, the FusionIO device produced for both benchmarks almost similar random-reads and random-writes.

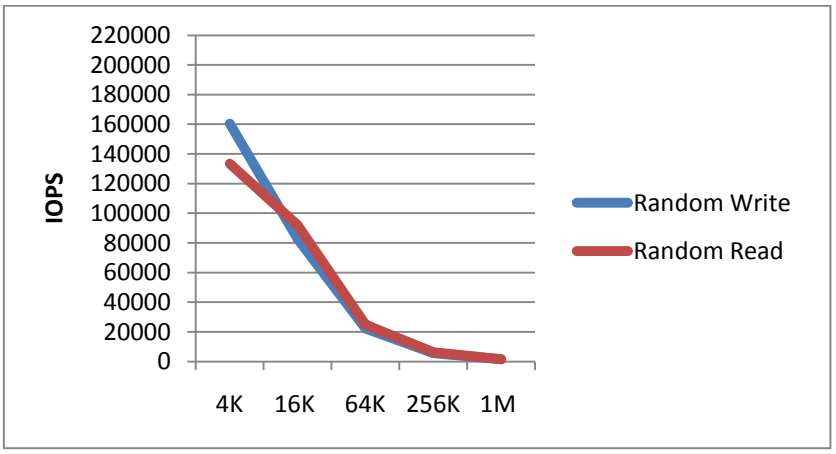

Figure 11: FIO Random IOPS on FusionIO Card

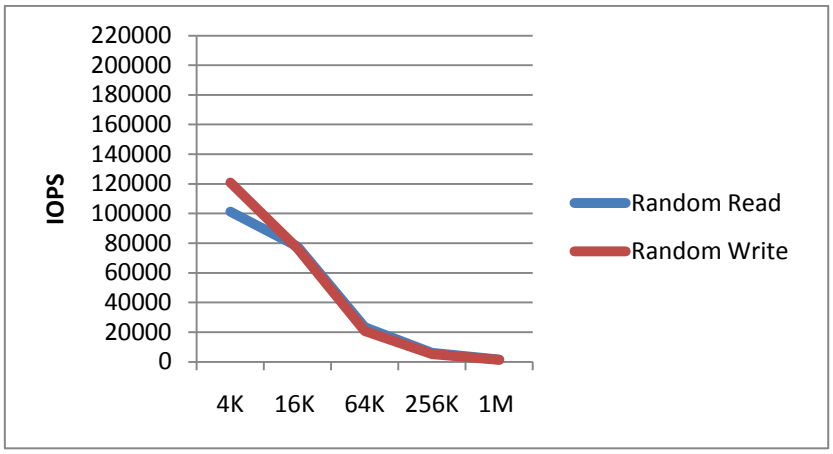

Figure 12: IOR Random IOPS on FusionIO Card

4. Conclusion

We evaluated two enterprise class SSD devices based on SLC NAND using a PCIe connection. As test method we used FIO and IOR, two well-known open source in order to perform bandwidth and IOPS measurements with both devices.

The bandwidth and IOPS performance – compared to SATA drives – is outstanding. Especially for the FusionIO card we could not observe large difference between read or write performance. Also we did not notice any performance degradation when filling the SSDs with data close to the maximum capacity.

However both cards are advertised with significantly higher specs than observed in our environment. We believe that our analysis tools give a more realistic view of what one can expect in a real production environment.

The performance and the overall behavior of the FusionIO card outperformed the Virident card. However, it is known from other studies elsewhere, the performance of the PCI cards is significantly influenced by the drivers.

Virident reproduced some of our results and they filed immediately a bug report with Engineering it is planned to provide a new driver for AMD based servers. We will repeat our experiments once this new driver is available to us.

The advantage of the Virident card is the replacement of single flash modules, which is not possible with the FusionIO card. Also it is possible to upgrade the capacity of the card by adding on additional modules up to 16 modules per card.

In the future we plan to test these devices in our GPFS experimental environment, in order to enhance the performance of metadata servers which have FC-drives right now.

5. Literature

[1] Vijay Karamcheti, Virident: Delivering High Sustained Performance with Enterprise Class Reliability, White Paper, 2011.

[2] Fusion I/O Drive Data Sheet: http://community.fusionio.com/media/p/852.aspx, 2011.

[3] Neal M. Master et al.: Performance Analysis of Commodity and Enterprise Class Flash Devices. Report on a project funded by DOE.

[4] FIO, http://freshmeat.net/projects/fio.

[4] J. He, J. Bennett, and A. Snavely: DASH-IO: an Empirical Study of Flash-based IO for HPC, in IEEE and ACM SC 2010, 2011.

[5] S. Racherla et al.: IBM Midrange System Storage Hardware Guide, IBM Redbooks, ISBN 0738434159 , 2010.

[6] A. Caulfield, L. Grupp, S. Swanson: Gordon, Using Flash Memory to Build Fast, Power-efficient Clusters for Data-intensive Applications. ASPLOS´09, March 7-11, Washington, DC, USA, 2009.

[7] Modeling and Simulation at the Exascale for Energy and the Environment, Report on the Advanced Scientific Computing Research, US Department of Energy, Office of Science, 2007.