D1.1 Server Scalibility

Ronald van der Pol and Freek Dijkstra

SARA Computing & Networking Services, Science Park 121, 1098 XG Amsterdam, The Netherlands
March 2010
ronald.vanderpol@sara.nl,freek.dijkstra@sara.nl
http://nrg.sara.nl/

1 Introduction

SARA's contribution to the GigaPort3 network research is in the area of demanding applications. There is still a large gap between the bandwidth that local and wide area networks can provide and the speed at which a single server can send and receive data on the network. SARA is working on identifying the bottlenecks in the the overall end-to-end performance and trying to remove those bottlenecks in order to get a higher throughput between applications running on different servers.

This document describes the scalability analysis and tests that SARA has done in order to decide on the specifications for a high end server. The aim for this high end server is to reach a throughput of 40 Gb/s for a single application. There are two obvious bottlenecks in present day high end servers: the disk I/O and the memory bandwidth. SARA has done extensive testing with solid state disks. The results are described in section 3. Section 2 gives an overview of the system architecture. PCI Express data rates are described in section 4. Section 5 describes the network I/O tests and we end with conclusions and future work.

2 System Architecture Overview

Appendix A shows the specifications of the server that was used for the tests described in this document. It is a Dell PowerEdge T610 [1]. This is an off the shelf server with Intel Xeon 5550 processor and Intel 5520 chipset. Figure 1 shows the block diagram of the architecture of this combination.

The Xeon 5550 has four cores supporting 8 threads and runs at 2.66 GHz. It has three memory channels to DDR3 memory running at 1333 MHz. This has a peak transfer rate of 10667 MB/s for each channel.

The I/O controller is the Intel 5520 I/O Hub, which is connected to the CPU with a 6.4 GT/s Quick Path Interconnect (QPI). The Intel 5520 supports 36 PCI Express 2.0 lanes of 500 MB/s each, for a peak transfer rate of 18 GB/s.

The Dell T610 server used by SARA has two PCI Express 2.0 $\times 8$ slots, three PCI Express 2.0 $\times 4$ slots and a PCI Express 1.0 $\times 4$ slot for the PERC RAID controller.

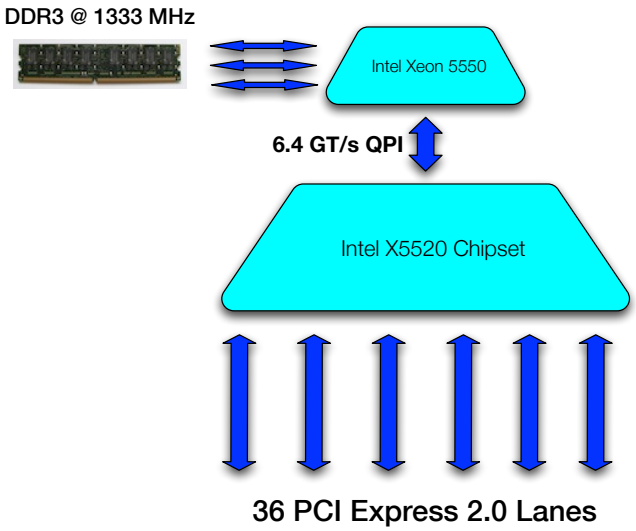

Fig. 1. Typical architecture of an Intel Xeon 5550 and 5520 chipset.

3 Filesystem and Disk I/O

In order to determine the scalability of multiple solid state disks several tests were done. In this section the major results and conclusions of the full report [4] are described. The filesystems with the highest throughput were BTRFS on Linux and ZFS on FreeBSD.

3.1 Solid State Disks

Solid state disks (SSDs) use non-volatile NAND (flash) memory instead of the rotating magnetic platters of a mechanical hard disk drive (HDD). The major advantages of solid state disks compared to regular hard disks are low energy consumption, high reliability (due to the lack of moving parts), and near-zero access times.

For our purpose, a streaming server, we are only interested in the sustained bulk read times from disk. Despite that file systems have gone to extensive length to tune the sequential read performance for mechanical hard disk drives, earlier reports show that solid state disks still outperform mechanical hard disk drives [5, 6].

There are currently two types of SSDs available on the market, single-level cell (SLC) and multi-level cell (MLC) SSDs. SLC SSDs achieve higher write performance, while MLC SSDs have lower cost and more capacity [5]. The file read performance is roughly the same for MLC and SLC SSDs [6]. For this reason, we conducted our experiments with MLC SSDs, using Intel's X25-M SSDs.

The Intel specification for the X25-M SSDs claims a maximum throughput of 250 MB/s sustained read speed. External parties have achieved up to 154 MB/s in practice [6].

3.2 Experimental Setup

Read performance was measured for up to six solid state or hard disk drives in various RAID configurations. All experiments were performed using the sequential read speed in the IOzone benchmark. While 'sequential' read has no meaning for solid state disks, this allowed a good comparison to mechanical hard disk drives. The Phoronix test suite was used to automate the tests. Each test was repeated three times, while verifying that the CPU usage stayed well bellow 100% (making sure that the CPU was not the bottleneck). The file size of each test was 8 or 32 GByte, well above the 6 GByte internal memory of the machine.

3.3 Results

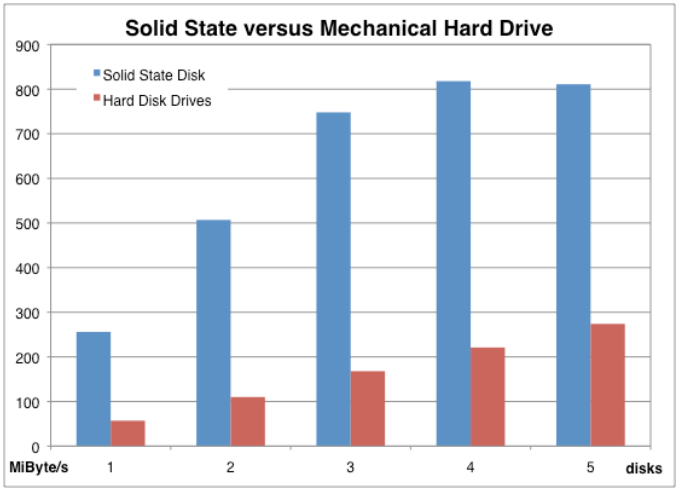

Fig. 2. Disk array read performance for up to five solid state disks.

We got very good sustained read performance for a single disk, easily exceeding 200 MB/s and sometimes even over 250 MB/s. However, the read speed does not grow linearly with the number of disks, levelling off around five solid state disks.

In order to eliminate the bottleneck of a single RAID controller, two RAID controllers were used. However, this did not result in a better performance. The

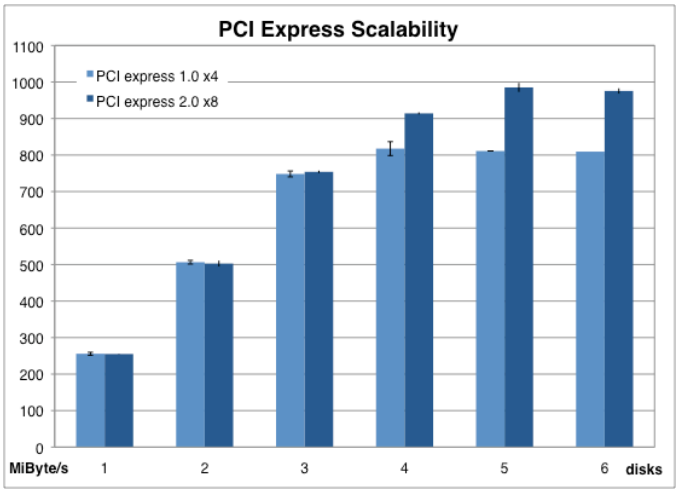

Fig. 3. Measurements for PCI Express 1.0 and 2.0

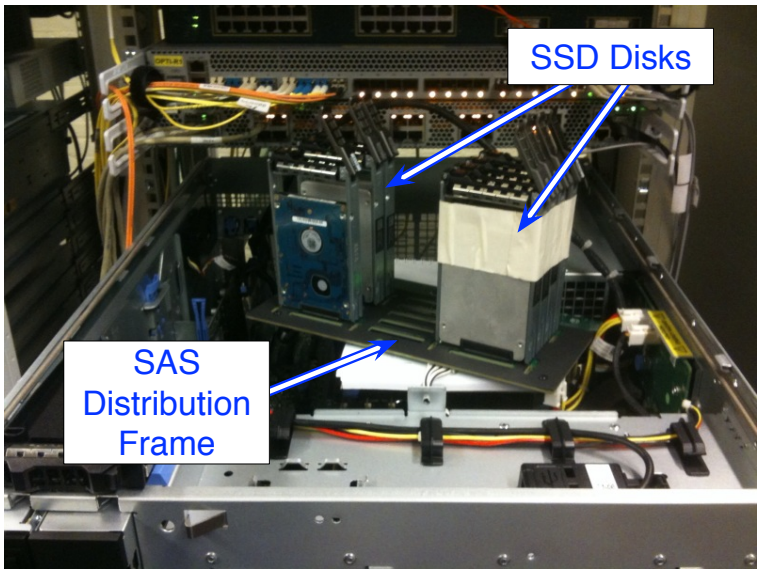

Fig. 4. Experimental setup with the RAID controller in a PCI Express 2.0 slot.

performance for Btrfs remained the same at 815 MiByte/s, and the performance of ZFS even dropped a little.

The levelling off in figure 2 occurs around 813 MiByte/s, or 6.82 Gb/s. This is close to the theoretical maximum of the PCI Express 1.0 ×4 bus, so we repeated the measurement with a PCI Express 2.0 ×8 slot. We used the PERC RAID

card only. Because the cable between the PERC RAID controller card and the SAS distribution frame with 8 SAS connectors was too short, we had to partly disassemble the server to do the tests (see figure 4). The result of the tests was a levelling off at 980 MiByte/s instead of 813 MiByte/s, as shown in figure 3.

3.4 Filesystems

Both operating system and file system are involved in the disk I/O operations, and are thus in part responsible for the disk I/O performance. All current I/O queues and file systems are designed with the limitations and particularities of mechanical hard disk drives in mind. For example, lots of effort has gone into tuning the physical location of blocks on a disk (to enhance sequential read performance) and in the queueing order for disk operations (minimising the movement of arms and platters). Neither of these enhancements is of any meaning to solid state disks.

Instead, solid state disks require enhancements geared towards wear levelling and reducing the number of times a page has to be updated. The minimum write size on solid state disks is a page, which has a size of 4 kB. However, the minimum delete size is a block, which has a size of 512 kB or 128 pages. This means that in order to update a single page, the other 127 pages in the block also need to be rewritten to disk.

The most significant enhancement for solid state disks is the introduction of the TRIM function in the DATA SET MANAGEMENT command of the ATA specification. The Operating System can use the TRIM function to signal to the SSD firmware which data blocks have been deleted by the file system. The helps the SSD firmware to manage the data blocks on the disk more efficiently.

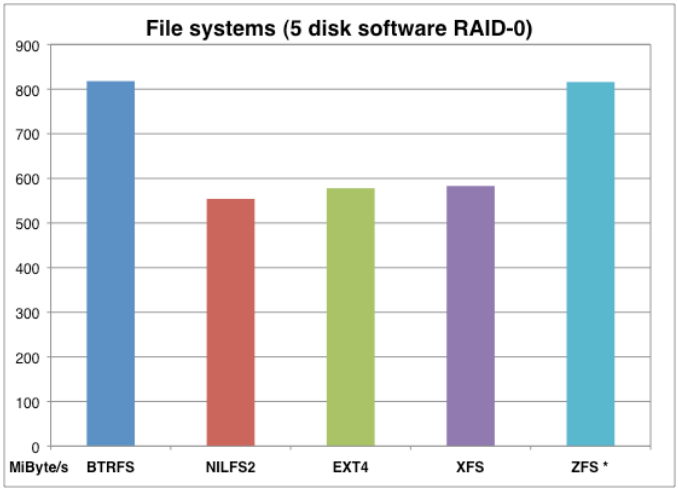

Fig. 5. Read performance with different file systems. ZFS runs under FreeBSD, the other file systems under Ubuntu Linux.

The performed tests examined the following file systems:

- Ext4
- Btrfs
- Nilfs2
- XFS
- ZFS

File systems specifically designed for flash memory, such as JFFS2, UBIFS or LogFS, can not be used with SATA or ATA-based solid state disks; they can only be used on raw flash memory. All file systems were tested under Linux (Ubuntu 8.10, Linux 2.6.31), with the exception of ZFS, which was run under FreeBSD 8.0[1]. The filesystems with the highest throughput were BTRFS on Linux and ZFS on FreeBSD.

3.5 RAID Controllers

Our ultimate aim is to achieve streaming rates of well over 10 Gb/s, up to 40 Gb/s. For this reason, multiple disks are required. The chosen RAID level greatly affects the I/O performance of a disk array. Since the goal is raw speed as opposed to reliability, we choose RAID 0 for most experiments. The tests examined both hardware RAID and software RAID. For software RAID and RAIDz (as used by ZFS), the on-board RAID controller was configured as "just a bunch of disks" (JBOD), letting the software handle the RAID.

Our tests were performed using two different RAID controllers:

[1] ZFS is not supported in the Linux kernel due to a licensing incompatibility.

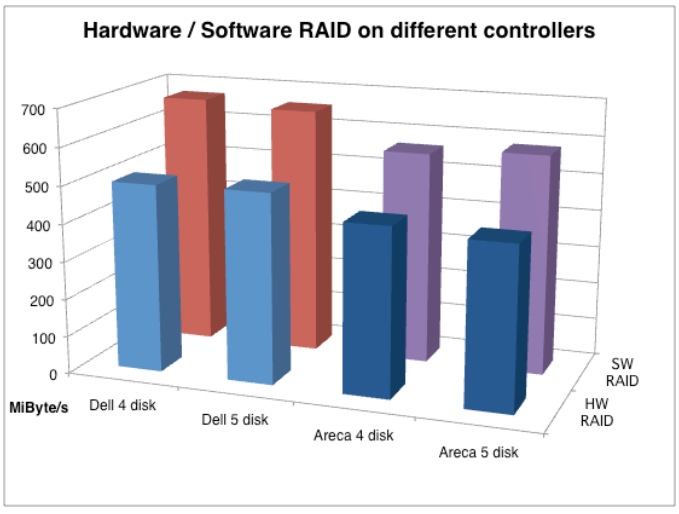

Fig. 6. Comparison for hardware and software RAID.

– Dell PERC 6/i SAS RAID controller
– Areca ARC-1680 PCI Express SAS RAID controller

Unfortunately, the Areca ARC-1880, which is said to be better tuned for solid state disks, was not available at the time of testing.

Contrary to our expectations, the software RAID outperformed the hardware RAID. Also, the Areca ARC-1680 did not outperform the Dell PERC 6/i controller. Simultaneous use of multiple RAID cards could have indicated whether the bottleneck would be in either the RAID controller itself, or the bus in which the controller is installed. However, the results were inconclusive, with Btrfs yielding higher performance (suggesting that the RAID controller is the bottleneck), while ZFS yielding equal or even lower performance (suggesting that the PCI Express bus is the bottleneck).

4 PCI Express Data Rates

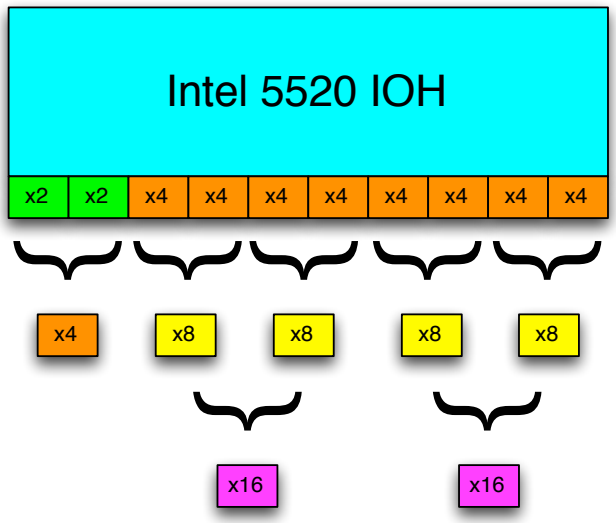

Fig. 7. Intel 5520 chipset PCI Express combinations

Figure 1 shows that all I/O connections to and from the CPU pass through a 6.4 GT/s Quick Path Interconnect (QPI). The Intel 5520 chipset handles the I/O and supports 36 PCI Express 2.0 lanes. Figure 7 shows how these lanes can be grouped to form the various ×2, ×4, ×8 and ×16 slots (see also figure 2-2 of [2]).

In the T610 server 32 of these 36 lanes are configured as two ×8 PCI Express 2.0 slots for full-height, full length cards, three ×4 PCI Express 2.0 slots for full-height, half-length cards and one ×4 PCI Express 2.0 slot for the PERC controller.

PCI Express 1.0 has a clock of 2.5 GHz and operates at 2.5 GT/s. For data encoding the 10b/8b algorithm is used, which results in an effective data rate of 250 MB/s. For PCI Express 2.0 the clock rate was doubled to 5.0 GHz, resulting in 5.0 GT/s and 500 MB/s effective data rate.

This gives the following effective data rates for the various PCI Express slots, as shown in table 1. The two ×8 PCI Express 2.0 slots have an effective data rate of 32 Gb/s and the three ×4 PCI Express 2.0 slots have an effective data rate of 16 Gb/s. This is only true when PCI Express 2.0 cards are used. When using PCI Express 1.0 cards the effective data rate is cut by half.

	×4	×8
PCI Express 1.0	8 Gb/s	16 Gb/s
PCI Express 2.0	16 Gb/s	32 Gb/s

Table 1. Effective data rates of PCI Express slots

5 Network I/O Tests

A series of tests using multiple 10 Gb/s network cards was performed in order to investigate possible bottlenecks when using multiple NICs simultaniously. For these tests SARA had access to seven 10GE NICs of three different types [3]:

- one 10GE SR NIC (Dell E15729)
- four 10GE UTP NICs (Intel E10G41AT2)
- two dual port 10GE SFP+ NICs (Intel E10G42AFDA)

All seven cards clock at 2.5 GHz[2]. This means that we can reach 10 Gb/s only in the two PCI Express ×8 slots, which have a maximum data rate of 16 Gb/s. The three PCI Express ×4 slots are limited to a maximum data rate of 8 Gb/s when clocking at 2.5 GHz. Tests were done with two 10GE UTP NICs in the two PCI Express ×8 slots (slot 2 and slot 3) and a 10GE UTP NIC in a PCI Express ×4 slot (slot 1).

The tests were done with Ubuntu 9.10 with a Linux 2.6.31 kernel. We used the following kernel tuning parameters in /etc/sysctl.conf:

```
net.core.rmem_max = 102400000
net.core.wmem_max = 102400000
net.core.rmem_default = 524287
```

[2] despite the branding as a "PCI Express 2.0 card"

```
net.core.wmem_default = 524287
net.core.optmem_max = 524287
net.core.netdev_max_backlog = 300000
```

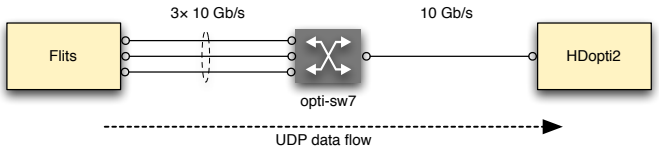

Fig. 8. Triple network link configuration between two hosts. Since UDP is used, the data is dropped at the switch.

All tests were done using UDP traffic and an MTU of 9000 bytes. Three iperf streams were used, one for each of the three interfaces. The following arguments were used:

```
sender$ iperf -u -i 1 -t 300 -c 10.20.20.133 -l 63K -w 200K -b 10000M
receiver$ iperf -u -s -B 10.20.20.133 -l 63K -w 200K
```

The results are presented in table 2. This shows that SARA was able to reach

	slot 1	slot 2	slot 3	total
no CPU pinning	6.74 Gb/s	9.24 Gb/s	9.23 Gb/s	25.21 Gb/s
with CPU pinning	6.91 Gb/s	9.93 Gb/s	9.93 Gb/s	26.77 Gb/s

Table 2. UDP Throughput

full 10G throughput on both PCI Express ×8 slots and slightly less than 7 Gb/s on the PCI Express ×4 slot. This slot has a theoretical maximum data rate of 8 Gb/s. The effect on CPU usage was also investigated. The server has a single Xeon 5550 processor, which has 4 cores (and 8 threads). When the iperf processes were started without pinning them to a particular core, the CPU usage was as shown in table 3. When we used CPU pinning with *taskset*, this resulted in a CPU usage as shown in table 4. The idea of pinning is to force the distribution of the processes over the cores in the most efficient way with respect to CPU load and possible I/O bottlenecks [7]. Linux assigns CPU 0 and CPU 4 to the two threads of core 0, CPU 1 and CPU 5 to core 1, CPU 2 and CPU 6 to core 2 and CPU 3 and CPU 7 to core 3. The following commands were used to pin each of the three iperf processes to a particular core:

```
$ taskset 0x00000002 iperf ....
$ taskset 0x00000004 iperf ....
$ taskset 0x00000008 iperf ....
```

id	user	system	nice	idle	iowait	hardware IRQ	software interrupt	steal time
CPU 0	2.0	60.8	0.0	12.0	0.0	0.7	24.6	0.0
CPU 1	2.0	4.6	0.0	91.4	0.0	0.0	2.0	0.0
CPU 2	0.0	3.9	0.0	96.1	0.0	0.0	0.0	0.0
CPU 3	0.0	2.6	0.0	97.4	0.0	0.0	0.0	0.0
CPU 4	0.0	3.1	0.0	96.2	0.0	0.0	0.6	0.0
CPU 5	0.7	0.3	0.0	99.0	0.0	0.0	0.0	0.0
CPU 6	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
CPU 7	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0

Table 3. CPU Usage Without Pinning

id	user	system	nice	idle	iowait	hardware IRQ	software interrupt	steal time
CPU 0	0.3	0.0	0.0	99.0	0.0	0.0	0.7	0.0
CPU 1	0.6	34.6	0.0	9.8	0.0	0.0	5.1	0.0
CPU 2	2.9	31.0	0.0	59.6	0.0	0.0	6.6	0.0
CPU 3	0.0	19.2	0.0	76.3	0.0	0.0	4.5	0.0
CPU 4	2.0	0.0	0.0	98.0	0.0	0.0	0.0	0.0
CPU 5	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
CPU 6	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
CPU 7	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0

Table 4. Pinning each iperf to a specific core

This pins the first iperf to CPU 1 (core 1), the second iperf to CPU 2 (core 2) and the third iperf to CPU 3 (core 3).

6 Conclusions

The purpose of the work described in this document was to investigate bottlenecks in an off the shelf server when trying to transfer data from disk to network with higher than 10 Gb/s speeds. For this work SARA used a single processor Intel Xeon 5550 based server with Intel 5520 chipset and various 10GE NICs. For the disk I/O tests six Intel X25-M G2 Solid State Disks were used.

SARA was able to reach a network I/O throughput of 26.77 Gb/s with three 10GE NICs and a disk I/O throughput of 8.22 Gb/s with 6 SSD disks. Based on these results, SARA expects to be able to reach 40 Gb/s of network I/O with two dual port PCI Express 2.0 ×8 NICs. As expected, getting the data from disk with this speed is probably going to be more of a challenge. This probably requires 5 to 6 PCI Express 2.0 ×8 RAID controller cards and 40 to 50 SSD disks. Finding a motherboard with 7 to 8 PCI Express 2.0 ×8 slots may be difficult.

The PCI Express I/O turned out the major cause of lower than expected thoughputs. Although the various 10GE NICs were advertised as PCI Express 2.0 cards, it turned out they were all using 2.5 GHz clocking, which resulted in

PCI Express 1.0 data rates. Therefore, it was important to insert these cards in one of the two ×8 slots in order to reach 10 Gb/s throughput. The ×4 slots are limited to 8 Gb/s data rate when using 2.5 GHz clocking.

With the SSD disk I/O tests we also experienced the effect of PCI Express slots with limited data rates. With the PERC RAID controller in a ×4 slot the throughput leveled off at around 6.8 Gb/s. With the PERC RAID controller in a ×4 slot the leveling off accured at around 8.2 Gb/s. At this moment, we are not sure why there is still leveling off in a ×8 slot. The data rate of a PCI Express 2.0 ×8 slot is 16 Gb/s.

In the tests that SARA has done so far, there was no evidence that the processor or memory was a bottleneck, although pinning processes to particular cores might be needed to take full advantage of the multi-core architecture.

The results of this work are used to buy a new high end server. With this server SARA plans to stream scientific visualisation data at speeds of up to 40 Gb/s. In order to decide on the specifications of this server, several additional tests need to be done. SARA will continue the network tests with a true PCI Express 2.0 dual port 10GE adapter. Such a card should be able to reach 20 Gb/s in a PCI Express 2.0 ×8 slot. SARA will also do addtional testing with a high end RAID controller that is capable of using the full capacity of both the PCI Express slot and the SATA/SAS speed.

7 Acknowledgments

This work was funded by the GigaPort3 project. Pieter de Boer helped with the architecture analysis. Igor Idziejczak and Mark Meijerink helped with network tests. Daan Muller and Sebastian Carlier performed the original SSD tests described in section 3. The Areca 1680 RAID controller used for some tests was lend to us by WebConnexxion.

8 Appendix A

Specifications of the Dell T610 server that was used for the solid state disk throughput tests.

- Intel Xeon ×5550 2.66GHZ
- 6 GB (3 x 2 GB) DDR3 @ 1333 MHz
- 2x 2.5” 7200 RPM 250 GB SATA Hard Disks
- 6x 2.5” 160 GB Intel X25-M G2 Solid State Disks
- 1x PERC 6/i SAS RAID Controller Card 256MB PCIe
- 1x 10GE SR NIC
- 4x 10GE UTP NICs
- 2x dual port 10GE SFP+ NICs

References

1. Dell PowerEdge T610 Technical Guidebook
   https://noc.sara.nl/wiki/images/6/62/Server-poweredge-t610-tech-guidebook.pdf
2. Intel 5520 Chipset and Intel 5500 Chipset Datasheet http://www.intel.com/Assets/PDF/datasheet/321328.pdf
3. Intel Ethernet Server and Desktop PCI Express Adapters
   http://download.intel.com/network/connectivity/products/prodbrf/252454.pdf
4. Sebastian Carlier and Daan Muller:
   SSD Performance
   UvA SNE Master project 2010 http://staff.science.uva.nl/~delaat/sne-2009-2010/p30/report.pdf
5. Terry Yoshii, Christian Black, and Sudip Chahal:
   Solid-State Drives in the Enterprise: A Proof of Concept
   Intel white paper http://download.intel.com/it/pdf/Solid_state_drives_in_Enterprise.pdf
6. Geoff Gasior:
   Intel's X25-E Extreme solid-state drive
   http://techreport.com/articles.x/15931
7. Vishwanath, V., Leigh, J., Shimizu, T., Nam, S., Renambot, L., Takahashi, H., Takizawa, M., Kamatani, O.:
   The Rails Toolkit (RTK) - Towards End-System Topology-Aware High End Computing
   Proceedings of the 4th IEEE International Conference on e-Science
   12/07/2008 - 12/12/2008 http://www.evl.uic.edu/files/pdf/RTK_e-Science08_Vishwanath.pdf