Performance of Direct Attached Disk Subsystems

Roger D. Chamberlain
Berkley Shands

Exegy, Inc.

and

Dept. of Computer Science and Engineering
Washington University

Performance of Direct Attached Disk Subsystems

Roger D. Chamberlain*[†] and Berkley Shands[†]
*Exegy, Inc., St. Louis, Missouri
[†]Dept. of Computer Science and Engineering, Washington University in St. Louis
{roger,berkley}@cse.wustl.edu

Introduction

For a number of years, our group has been constructing appliance-class systems that include significant local disk storage as an integral component of the system [1,2]. As a part of this activity, we have previously reported empirical performance results for the disk subsystem [3]. In this previous study, we investigated primarily read performance across a variety of disk subsystems (drive types, controllers, RAID configurations, etc.) and file size distributions.

Here, we describe a new set of experiments that both updates the previous investigation to currently available components and also widens the scope of the study, adding multiple file systems and usage patterns.

Experimental Systems

We use two hardware platforms to carry out our experiments. The first platform occupies 3U of rack space in a single enclosure and contains 16 drives (totaling 8 TB). The second platform occupies 9U of rack space in three enclosures and contains 32 drives (totaling 13 TB). Details of each hardware platform are listed below:

Platform 1:

1	TSTCOM ESR316 enclosure (3U)
1	Tyan 3992 motherboard
2	AMD Opteron 2222 dual-core 3 GHz procs
2	LSI 8888elp SAS disk controllers
16	Seagate 500 GB NS series SATA-2 disks

Platform 2:

1	Uniwide 3546ES enclosure (3U) and motherboard
4	AMD Opteron 8222 dual-core 3 GHz procs
1	LSI 8888elp SAS disk controller
2	Xtore XJ1100 SAS JBOD expansion enclosures (3U each)
16	Seagate 500 GB NS series SATA-2 disks
16	Seagate 320 GB AS series SATA-2 disks

Each platform has 16 GB of RAM, uses CentOS 5 (Linux kernel 2.6.20), and the disk controllers are each installed in 8-lane PCI-e slots. Also present (installed in a PCI-X slot) is a board that includes two Xilinx Virtex-4 LX100 platform FPGAs. Each system has a dedicated boot disk (using a native controller on the motherboard), independent of the disk subsystems used for experimentation.

Disk Subsystems

Since each disk controller supports 8 SAS channels and there are two controllers present in platform 1, each disk is individually connected to one of the controllers via a dedicated channel. Platform 2 takes advantage of the 24 lane SAS switches present in the expansion enclosures to support 32 disks. The logical topology is shown in Figure 1.
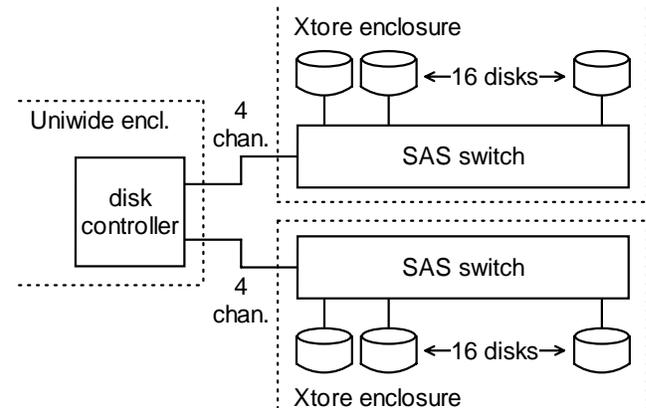

Figure 1: Disk subsystem for platform 2.

Several distinct configurations are investigated. The set of disks are partitioned into 2, 4, and 8 logical drives (RAIDs). For example, platform 1 configured for 2 logical drives would have 8 physical disks per logical drive (denoted below as a 2x8 configuration), and platform 2 configured for 8 logical drives would have 4 physical disks per logical drive (denoted as an 8x4 configuration). Finally, two distinct file systems are loaded, ext3 or SGI's xfs.

Data Movement

The software organization for reading data from the disk subsystem and delivering it to the FPGA (the normal path that is exploited in our systems) is illustrated in Figure 2. A number, k, of independent threads are allocated to associated silos. The silos are FIFO buffers in system memory that are used to stage data from the disk subsystem prior to delivery to the FPGA. The number of silos, k, is normally equal to the number of independent RAIDs in the disk subsystem configuration.

It is important to point out that the data movement indicated in the figure is under the control of an execution thread on one of the general-purpose processors, but the actual data transfers are all DMA transfers commanded either by the disk controller (for data moving from disk to silo) or by the firmware socket within the FPGA (for data moving from silo to FPGA).
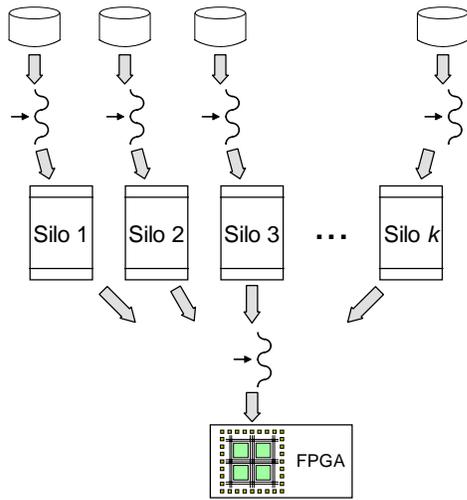
Figure 2: Read silos.

Experiments

The data set used in the majority of the experiments comprises a total of 1.9 million individual files drawn from a variety of sources, including: the complete Medline PubMed abstract database, the multilingual Reuter's corpus, one full day of CNN news articles, the TREC-5 Confusion Track corpus, and 4 captured Windows system images. Figure 3 plots a histogram of the number of files for each file size. The total data set is 129 GB in size.
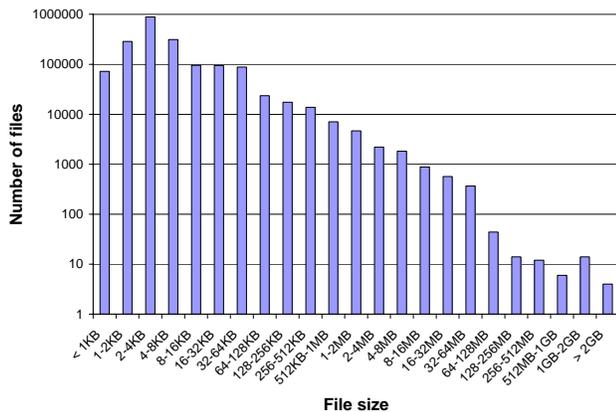

Figure 3: File size distribution.

In addition to the above data set, a collection of 8 GB synthetically generated files totaling an additional 128 GB is used to exercise the disk subsystem with sequential accesses and minimal meta-data processing required.

The following parameters are fixed for each experiment:

- platform {1 or 2}
- number of logical drives {2, 4, or 8}
- file system {ext3 or xfs}

Prior to each experiment, both of the data sets described above are loaded onto freshly formatted file systems (one file system on each logical drive). Additional data sets are derived from the original data set of Figure 3 by concatenating files smaller than a given threshold into aggregate files (maintaining appropriate indices to retain

the original file semantics) using the techniques described in [4]. This mechanism establishes a minimum file size to be managed by the underlying file system and is effective for infrequently written but frequently read file systems.

Once the data sets have been loaded, read tests are performed on each data set, separately measuring directory lookup times (i.e., meta-data processing) and file read times (i.e., data movement). These are followed by write tests using the same data sets.

Preliminary Results

Due to the limitations of space, only preliminary results are included here. More comprehensive results will be available in [5]. Figure 4 shows the read throughput for two experiments as a function of the minimum file size in the data set. The performance is clearly impacted fairly dramatically by the large numbers of small files present in the original data set of Figure 3. Examining the raw data, over half of the total time is spent doing directory lookups for the unbounded minimum file size case. In these two experiments, the xfs file system outperforms ext3 primarily on the basis of faster directory lookup times. Both file systems achieve 1 GB/s read rate on the synthetic data set.
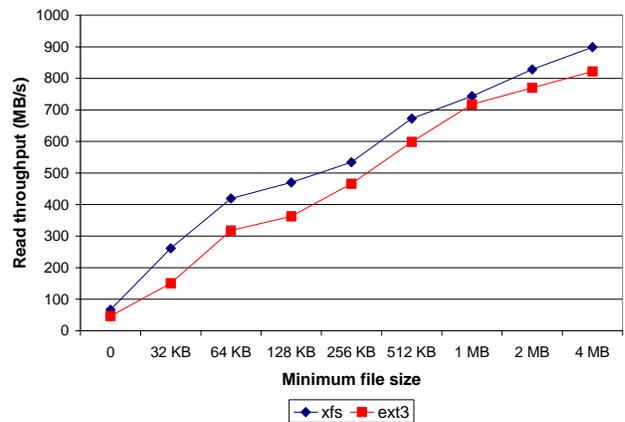

Figure 4: Read throughput, platform 2, 8x4 RAID0.

References

[1] M.A. Franklin, R.D. Chamberlain, M. Henrichs, B. Shands, and J. White, "An Architecture for Fast Processing of Large Unstructured Data Sets," In *Proc. of 22nd Int'l Conf. on Computer Design*, October 2004, pp. 280-287.

[2] R.D. Chamberlain, B. Shands, and J. White, "Achieving Real Data Throughput for an FPGA Co-Processor on Commodity Server Platforms," in *Proc. of 1st Workshop on Building Block Engine Architectures for Computers and Networks*, October 2004.

[3] R.D. Chamberlain and B. Shands, "Streaming Data from Disk Store to Application," in *Proc. of 3rd Int'l Workshop on Storage Network Architecture and Parallel I/Os*, September 2005, pp. 17-23.

[4] R.D. Chamberlain and R.K. Cytron, "Novel Techniques for Processing Unstructured Data Sets," in *Proc. of IEEE Aerospace Conference*, March 2005.

[5] R.D. Chamberlain and B. Shands, "Direct-Attached Disk Subsystem Performance Assessment," to appear in *Proc. of 4th Int'l Workshop on Storage Network Architecture and Parallel I/Os*, September 2007.