SLAC-PUB-11908
June 2006

PetaCache: A Memory-Based Data-Server System

Chuck Boeheim, Stephen J. Gowdy, Andy Hanushevsky, David Leith, Randy Melen,
Richard Mount, Teela Pulliam, Bill Weeks
Stanford Linear Accelerator Center
(boeheim, gowdy, abh, leith, randym, rmount, teela, wcw@slac.stanford.edu)

Abstract

Scientific advances depend increasingly on agility in the analysis of data, along with access to massive computation. The PetaCache project addresses the data-access issue by recognizing that the future for intense, non-sequential, data access must be based on low latency solid-state storage. The PetaCache architecture aims at a minimum unit cost, highly scalable hardware and software approach that can take advantage of existing and emerging solid-state storage technologies providing data-access latencies in the range 10 – 100 microseconds.

A prototype system has been constructed as a cluster of 64 nodes hosting a total of one terabyte of memory. Client processors retrieve data from the data-server nodes over a switched Ethernet infrastructure using SLAC's xrootd data-server software. The system is in use for performance testing, optimization and trial deployments for scientific data analysis. It also provides an excellent platform for testing new data access paradigms.

1: Introduction

Basic Science has grown more and more dependent on high-end computing, taking full advantage of the country's investment in developing super-computers. A growing fraction of those basic science activities in the US are also becoming more data intensive. This means that the codes executing the scientific exploration cannot access directly, through the close-coupled fast memory, the data under study. Rather the data is accessed from disk or from tertiary storage (potentially across a network). Highly selective subsets of the data can be made to increase the efficiency of this method. Experiment, observation and modeling generate huge volumes of data; progress in scientific understanding is enhanced or enabled by our ability to store, move, and access data.

Our rising ability to store data has not been matched by our ability to access data. In particular, as disk capacities in bytes per dollar continue their "Moore's Law" type of increase, disk latencies in random accesses per second per dollar have been static for a decade. Much of the use of data in science is now crippled by the time it takes to retrieve objects from disk. The raw device level latency for disk is around 6 milliseconds[1] to be compared with 50 nanoseconds for memory: a disk latency penalty factor of more than 100,000. Even in large systems optimized for disk-based access to hundreds of terabytes, the latency penalty imposed by disk is more than a factor 100.

Certain classes of problems have been addressed by traditional supercomputing techniques. These are characterized by sequential access to very large files, using large block sizes, and high-performance disks and channels. While very high I/O rates can be achieved for these sequential problems by streaming I/O, read-ahead and caching, these techniques are actually detrimental to the random access and small block I/O that characterize a very large class of data-intensive science computing.

PetaCache addresses this issue by creating software and hardware technology that can provide data to applications with 100 times less latency than that of disk-based storage systems, and with scalability to petabytes and beyond. The immediate approach to low-latency storage is to use commodity DRAM, but emerging "storage-class memory" technologies, typified today by consumer-driven flash memory, are a promising future direction.

The PetaCache approach is cluster based, allowing the use of commodity components interconnected by mainstream network technologies. This approach is targeted at applications that are data-intensive, which could see revolutionary benefits from data-access latency 100 times below that of disk storage. The application-level assumption is that data can normally be treated as immutable, so that hardware support for locking and cache coherence is not needed. In addition, we have focused initially on a file-access paradigm rather than the creation of a large directly addressable memory.

[1] We measure a device-level latency of 6.2 to 7.0 milliseconds for retrievals in the range 1byte to 5kilobytes from Seagate 10,000 rpm "Cheetah" disks.

Contributed to The 15th IEEE International Symposium on High Performance Distributed Computing, 6/19/2006--6/23/2006, Paris, France

Work supported in part by Department of Energy contract DE-AC02-76SF00515

Such a new architecture system potentially will provide much higher throughput of analysis in a data intensive environment (maybe by factors between 10 and 1000). Furthermore, it also opens the door for a revolution in how we ask questions of the data in heavily data-intensive studies.

2: PetaCache Prototype System

We have implemented a cluster with one terabyte of memory, installing SLAC's Scalla data server software[1], and making detailed measurements of performance and scalability, as described in the following sections.

2.1: Cluster Implementation

The cluster is composed of 64 dual-CPU systems consisting of commodity V20z's[2] from Sun. Each system contains dual Opteron 244 (1.8GHz) processors, a 36GB system disk, and a 73GB persistent-data disk. We populated the eight memory slots with 2GB DIMMs, for a total of 16GB, as larger 4GB DIMMs were not cost-effective. Each system has two gigabit Ethernet ports, of which only one was used.

We installed both Solaris 10 x64 and Red Hat Enterprise Linux 3 on these systems in a dual-boot configuration.

The cluster is connected to a Cisco 6509 switch with copper gigabit Ethernet. The switch has four bonded gigabit Ethernet connections to the main SLAC batch farm. The batch farm consists of 2,200 nodes (3850 processors) ranging from 440MHz Ultrasparc II to 2.0GHz Opterons. Each of these nodes has a 100 megabit connection to one of ten Cisco 6509 switches, which are interconnected with multiple gigabit Ethernet backbones.

2.2: Software Implementation

The file server software employed by this cluster is a generalization of several generations of previous experience[3] in databases for particle physics. The Scalla system consists of two components: xrootd[4][5] and olbd.

Xrootd provides byte-level access to files at device speeds, and is capable of serving thousands of simultaneous requests with low overhead. To date, we have demonstrated that the xrootd server delivers data at levels only limited by hardware speed (i.e., disk, network, and cpu). Server load scales linearly with the number of simultaneous requests

Olbd organizes clusters of xrootd servers serving a uniform namespace into a fault-tolerant and scalable distributed file system. Clustering provides multiple access point to increase the total amount of available data as well as to increase the overall throughput of the system. Configuring large clusters is simplified by allowing the cluster to self-organize. To date, we have successfully clustered over 1,000 servers and see no impediment in clustering larger numbers.

3: Results and Future Direction

Our current tests show that the system delivers latencies on the order of 100µs and scales well. The BaBar experiment is preparing a "sparse iteration" data access test for the system. Other experiments have expressed considerable interest in developing novel data access approaches for the system

The next stage of PetaCache development is to move beyond the "prototype system" to a "development system", just large enough to hold a significant amount of analyzable data. This is estimated to require 30TB of memory. We also plan to investigate two different technologies for scaling to the next level. The first is a continuation of the current DRAM-based model, and the second would use flash solid state storage.

4: Acknowledgement

Work supported by the U.S. Department of Energy under contract number DE-AC02-76-SF00515.

References

[1] The BaBar Database: Challenges, Trends and Projections, I. Gaponenko, A. Mokhtarani, S. Patton, D. Quarrie, A. Adesanya, J. Becla, A. Hanushevsky, A. Hasan, A. Trunov, CHEP Conference, Beijing, China, September 2001

[2] http://www.sun.com/servers/entry/v20z

[3] Lessons Learned from Managing a Petabyte, J. Becla, D. Wang, CIDR 2005 Conference, Asilomar, CA, USA, January 2005

[4] On the Verge of One Petabyte - the Story Behind the BaBar Database System, A. Adesanya, T. Azemoon, J. Becla, A. Hanushevsky, A. Hasan, W. Kroeger, A. Trunov, D. Wang, I. Gaponenko, S. Patton, D. Quarrie, CHEP Conference, La Jolla, CA, USA, March 2003

[5] http://xrootd.slac.stanford.edu/