

Special Issue on Parallel I/O Systems

David Kotz

Department of Computer Science

Dartmouth College

Hanover, NH 03755-3510

dfk@cs.dartmouth.edu

Many important applications have tremendously large data sets, gigabytes, terabytes, or even petabytes in size. The applications range from transaction-processing databases, to multimedia systems, to the huge scientific datasets used in seismic modeling, whole-earth climate models, aerospace engineering simulations, and the like. These large data stress the limits of I/O hardware and software systems, leading system and application designers to the use of parallelism: parallel processors, multiple disks and busses, and multiple layers in the memory hierarchy.

The five invited papers in this special issue cover a wide range of topics. In the first paper, Cormen and Nicol evaluate techniques for computing Fast Fourier Transforms (FFTs) of millions or billions of points, which are so large that they cannot fit in main memory. The algorithms use both parallel computation and parallel I/O, and their performance measurements demonstrate the importance of a carefully designed out-of-core algorithm over a solutions that depend on traditional demand-paged virtual-memory support.

The second paper considers an entirely different application, a video-on-demand service running on a dedicated multi-disk server. Papadopoulos and Golubchik evaluate techniques for the server to meet quality-of-service guarantees while adapting to both predictable and unpredictable workload fluctuations. Their techniques use a clever technique that stores multiple resolutions of the video stream on different disks, allowing them to adapt by changing resolution or by redistributing the load on the disks.

In the third paper, Bordawekar, Landherr, Capps, and Davis evaluate the architecture and systems software of a relatively new multiprocessor, the Hewlett-Packard (formerly Convex) Exemplar. The team includes Exemplar experts from Hewlett-Packard as well as an experienced parallel-I/O researcher at Caltech. They carefully explain the architecture of the Exemplar and the characteristics of its file system, then examine the system's perfor-

mance in a series of microbenchmarks.

The fourth paper discusses distributed file systems that use striped disks. Rochberg and Gibson extend the TIP prefetching system, originally developed to improve file-system performance within a uniprocessor, to the NFS distributed file system. CTIP, as the new version is called, aggressively prefetches data from the server's parallel-disk system, through the server's cache, to the client cache. In a set of application benchmarks, CTIP reduced total execution times by 17-69%, and there is room for more improvement.

Finally, at the lowest level, Menon and Treiber describe their prototype virtual-disk hierarchical storage manager. In this system, they use the disk (or disk array) as a cache for a much larger tertiary-storage device, such as a robotic tape archive. *Virtual tracks* are migrated from tape to disk as needed, and back to tape when not accessed for a long time. They resolve many subtle issues, but in the end they have a subsystem that appears to be a disk drive, from the perspective of the file system, with the nearly the performance of a disk and the capacity of a tape system. They evaluate the performance of their prototype with the TPC-C benchmark.

If you're interested in learning more about parallel I/O systems, applications, theory, tools, and special events, check out the parallel-I/O web site at <http://www.cs.dartmouth.edu/pario/>.

Many thanks to the invited authors that contributed their papers to this special issue of PER. I hope that you enjoy the papers as much as I did!