CHARISMA: A Smart Look at Parallel File-Access Patterns

Michael L. Best*
mikeb@media.mit.edu

MIT Media Laboratory

1 Introduction

The so-called "I/O bottleneck" in high-performance scientific parallel computing is, at this point, axiomatic. While multiprocessors have been able to ride the wave of Moore's law (which states that processors will double their performance every 18 months) their supporting I/O subsytems have been often left in its wake. Thus, the gap between scalable computing systems' delivered computational performance and their I/O bandwidth has widened.

Parallel filesystem software and disk array hardware has been the primary path towards closing this gap. The design of such filesystems has been guided, at least initially, by our experience with workloads and access patterns on traditional serial I/O systems. A collection of research programs have studied serial filesystems in a variety of environments, produced models of access patterns, generated synthetic loads, and so forth. The match between these research results (and their corresponding real-life workloads) and the I/O usage patterns of scientific applications on parallel platforms is uneven at best. Some researchers, however, have begun to study parallel I/O usage by examining selected applications or using anecdotal or informal evidence [CHK+93], [GGL93], [RC94].

The CHARISMA (CHARacterizing I/O in Scientific Multiprocessor Applications) project was started with the goal of studying real-life workloads of production scientific applications running on parallel systems. Initial results were collected from the iPSC/860 at the NASA Ames Research Center and were reported in [KN94]. Traces were also made of I/O usage on a CM-5 at the National Center for Supercomputing Applications (NCSA). These results are reported in [PEK+95]. In this abstract we will present the general methodology used to collect workload data, summarize the main results from our two studies, and present some goals for future research.

2 Methodology

We instrumented the Concurrent File System (CFS) [Pier89] on an iPSC/860, the CMMD message passing library [BGST93] on the CM-5, and CMF the data parallel Fortran compiler on the CM-5. The instrumented filesystem, communication library, and compiler were then installed on their respective parallel platforms at NASA and NCSA. Usage data was accumulated and studied over the span of a few weeks. Across all three systems a standard trace format was used (with minor variation) and a single set of analysis tools were implemented. Trace data logged the salient features of system calls such as open, close, read, write, truncate, and so forth. I/O to tty's (and in the case of CFS all serial I/O) was not traced. Parallel extensions to traditional I/O operations (such as the setting and use of I/O modes in CMMD and CFS) were also logged and studied. In the case of CFS and CMF all programs linked and ran during the test period were traced; for CMMD users were required to link against an experimental library which resulted in some users alluding the study.

By tightly coupling trace data production with the systems I/O libraries we were able to carry on our studies without overly perturbing program behavior. Anecdotal evidence supports this claim.

3 Main Results

Across all three instrumented systems we have noticed a number of characteristics we believe are important and general to parallel workloads.

We observed a significant mix of small and large I/O operations. For both read and write most I/O's were small in size, yet most data was transferred during large I/O's (Figure 1). This is an important observation; parallel filesystems will need to optimize both the latency of small I/O's and the bandwidth of large ones. It is interesting that these results were true across platforms. CFS provides lower latency but does not enjoy the high bandwidth of the CM-5's I/O subsystem. The CM-5 can provide very high bandwidth I/O but at a high start-up cost (on the order of 100 milliseconds). Thus users are not simply coding their I/O to meet the performance strengths of their particular platform. Further, these conclusions were true in the cases where single nodes independently performed an I/O as well as when all nodes of a machine ganged together in a co-operative operation.

* Michael L. Best is advised by Prof. Ken Haase and is supported by sponsers of the MIT Media Lab. This research was supported in part by the National Center for Supercomputer Applications, NASA Ames Research Center under agreement number NCC 2-849, and Thinking Machines Corporation. Address: E15-468B, 20 Ames Street, Cambridge MA 02139. Web: http://mikeb.www.media.mit.edu/people/mikeb/

As is the case with serial I/O, we found parallel access patterns to be mostly sequential and consecutive. A sequential request is one which is at a higher file offset then the previous request, a consecutive request is sequential and begins where the previous request left off. However, we also noted new interleaved access patterns. Here, sequential access amongst nodes included a (generally uniform) stride of unaccessed bytes between nodes. Some recent parallel filesystems (e.g. [CFPB93]) have included built-in support for strided I/O.

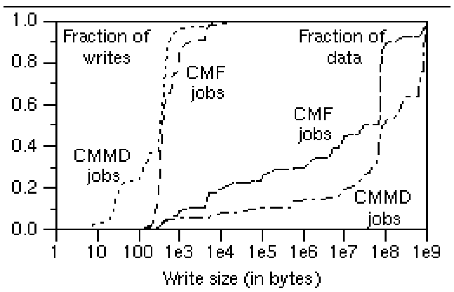

Figure 1: Cumulative Distribution Function (CDF) of number of writes by request size and amount of data by each request under CMMD and CMF on a CM-5.

It is rare in serial environments to observe inter-program file sharing. For parallel programs, however, we expect frequent intra-program file sharing (across nodes in a single job). Our trace data showed this to be almost exclusively read-only sharing with little write-sharing (Figure 2). The large amount of block sharing (in contrast to byte sharing), even in the case of write operations, demonstrates strong inter-node spatial locality and suggests that caching would be effective.

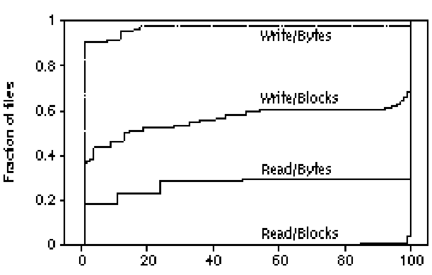

Figure 2: Cumulative Distribution Function (CDF) of file sharing under CFS on the iPSC/860.

4 Conclusions and Future Work

We have studied I/O workload characteristics across multiple platforms and programming models. We have observed usage patterns common with traditional serial filesystems (large file size, sequential access), as well as unique to parallel filesystems (small request sizes, intra-job file sharing, strided access).

We believe our results shed light on the workload characteristics of parallel filesystems. This data can help system engineers and researchers design effective I/O subsytems. Towards such a goal, we would like to use these results in the creation of synthetic workloads and access models.

Our tracing format and methodologies worked across a range of programing models and systems (data parallel, message passing, SPMD, MIMD). Thus we have an existence proof (of sorts) that our methods are useful across a variety of parallel platforms and models. We encourage other researchers to make use of our tracing formats and invite similar studies on other systems.

5 References

[BGST93] M.L. Best, A Greenberg, C. Stanfill, and L.W. Tucker. CMMD I/O: A parallel Unix I/O. In *Proceedings of the Seventh IEEE International Parallel Processing Symposium,* pages 489-495, 1993.

[CFPB93] P.F. Corbet, D.G. Feitelson, J.P. Prost, and S.J. Baylor. Parallel access to files in the Vesta File system. In *Proceedings of Supercomputing '93*, pages 472-481, 1993.

[CHK+93] R. Cypher, A. Ho, S. Konstantinidou, and P. Messina. Architectural requirements of parallel scientific applications with explicit communication. In *Proceedings of the 20th Annual International Symposium on Computer Architecture,* pages 2-13, 1993.

[GGL93] N. Galbreath, W. Gropp, and D. Levine. Applications-driven parallel I/O. In *Proceedings of Supercomputing '93,* pages 462-471, 1993.

[KN94] D. Kotz and N. Nieuwejaar. Dynamic file-access characteristics of a production parallel scientific workload. In *Proceedings of Supercomputing '94,* pages 640-649, 1994.

[Pier89] P. Pierce. A concurrent file system for a highly parallel mass storage system. In *Fourth Conference on Hypercube Concurrent Computer and Applications,* pages 155-160, 1989.

[PEK+95] A. Purakayastha, C.S. Ellis, D. Kotz, N. Nieuwejaar, and M.L. Best. Characterizing parallel file-access patterns on a large-scale multiprocessor. To appear *Proceedings of the Ninth IEEE International Parallel Processing Symposium,* 1995.

[RC94] J.M. del Rosario and A. Choudhary. High performance I/O for parallel computers: Problems and prospects. *IEEE Computer,* 27(3):59-68, March 1994.