Throughput of Existing Multiprocessor File Systems
(An Informal Study)

David Kotz

Technical Report PCS-TR93-190
Department of Math and Computer Science
Dartmouth College
Hanover, NH 03755-3551
David.Kotz@Dartmouth.edu

May 14, 1993

Fast file systems are critical for high-performance scientific computing, since many scientific applications have tremendous I/O requirements [MK91]. Many parallel supercomputers have only recently obtained fully parallel I/O architectures and file systems, which are necessary for scalable I/O performance. Scalability aside, I show here that many systems lack sufficient absolute performance.

I examined several the papers in the literature that report actual performance measurements on existing parallel file systems. For each paper, I chose the highest reported throughput. A list of these throughputs, normalized by the number and speed of disks, is given in the table on page 2.

First, note that the numbers come from different experimental arrangements and thus cannot be directly compared with each other. Some timings include computation, and some do not; some are for reading, some for writing; some are sequential, while some are not. Some of the parameters were estimated. (A commonly-used benchmark would help, and at least one has been proposed [CCFN92, Fin93]).

Second, my normalization is extremely crude: dividing by the number of disks and then by the raw disk's peak bandwidth is an over-simplification. Few systems can approach 100% of the available bandwidth, and then only in special cases such as a long sequential read. Still, the results are instructive in pointing out the difficulty of obtaining good I/O performance in a parallel system.

Third, note the wide range of throughputs. In only a few papers, and only a few cases in those papers, were the authors able to extract a significant percentage of the disk bandwidth. The performance for the 64-disk Touchstone Delta is particularly disappointing (although a new Intel file system is forthcoming). The best cases occurred when the I/O was coordinated (as in the CM-2, CM-5, or Nitzberg's limited concurrency [KN93]), or sequential reading with prefetching and a buffer size to match the block size [FPD93].

These results show us that more work is needed on improving the raw performance of multiprocessor file systems. I suspect that much of the problem is excess software overhead. Throughput is lost to extraneous copying and message-passing overhead, inadequate I/O node power, cache thrashing and prefetch mistakes [KN93, Nit92], architectural bottlenecks [KN93, Kry92], and lack of coordination [dBC93].

Corrections and additions to this table are welcome.

Throughputs of existing multiprocessor file systems: For each paper reporting real performance numbers, the best throughput was chosen for the table. (All rates are in MB/s.) Comparing results from different papers is difficult due to different experimental configurations. The point of this table is the disappointingly low performance obtained by many researchers, even in the best case, and with careful tuning. In comparison, in a sequential write test (also a best case) both a Unix LFS and a Unix extent-based file system attained nearly 100% of the disk bandwidth, while Unix FFS was limited to about 25% [SBMS93].

	Intel CFS							
	Model	Disks	I/O nodes	Total	Per disk	Raw Disk	% of raw	Reference
1	iPSC/860	10	10	8.0	0.80	1	80%	[KN93]
2	iPSC/2	4	4	3	0.75	1?	75%	[FPD93]
3	iPSC/2	16?	8	5.54	0.69	1	69%	[AS89]
4	iPSC/?	10	10	5.5	0.55	1?	55%	[Dun91]
5	iPSC/2	16	8	6.0	0.38	1	38%	[Pie89]
6	iPSC/2	4	2	2.3	0.58	1.875	31%	[Are91]
7	iPSC/2	4	4	0.62	0.15	1	15%	[PFDJ89]
8	iPSC/860	10	10	0.64	0.064	1	6.4%	[Fin93]
9	Delta	64	32	10	0.16	2.5	0.6%	[BCR92]
10	Delta	64	32	7.65	0.12	2.5	0.4%	[dBC93]
	nCUBE							
	Model	Disks	I/O nodes	Total	Per disk	Raw Disk	% of raw	Reference
11	nCUBE/2	4	4?	4.96	1.24			[DdR92]
12	nCUBE/2	6	6?	6.3	1.05			[dR92]
13	nCUBE/2	8	8	3.91	0.49			[dBC93]
14	nCUBE/10	8	8	2.5	0.31			[BN90]
15	nCUBE/10	8	8	0.27	0.03			[PFDJ89]
	TMC CM-2 DataVault							
	Model	Disks	I/O nodes	Total	Per disk	Raw Disk	% of raw	Reference
16	CM-2	32	(1)	25	0.78	1	78%	[KN93]
17	CM-2	32	(1)	4.3	0.13	1	13%	[Fin93]
	TMC CM-5 Scalable File System							
	Model	Disks	I/O nodes	Total	Per disk	Raw Disk	% of raw	Reference
18	CM-5	32	4	54.4	1.7	2	85%	[LIN+93]
19	CM-5	118	15	185	1.57	2	79%	[LIN+93]
	Cray							
	Model	Disks	I/O nodes	Total	Per disk	Raw Disk	% of raw	Reference
20	Y-MP/8	1	(1)	2.44	2.44	9.6	25%	[Fin93]

Detailed notes:

1. A higher rate was possible with interprocess cache locality. Data are estimated from Figures 6 and 7. See [Nit92] for more details.
2. A higher rate of 5.5MB/s for writes was not included because I think the time did not include waiting for the cache to flush at the end. Data are estimated from Figure 1.
3. This was done using beta-test software. The number of disks is unclear, but I expect that it is the same as in [Pie89]. Data are taken from table on page 132.
4. Raw disk performance unknown. Not clear whether data are for iPSC/2 or iPSC/860. Data are estimated from Figure 11.
5. Presumably using beta-test software. Data are estimated from Figures 1 and 2.
6. Data are taken from page 39 and Figure 4.7.
7. Data estimated from Figure 7. A higher rate of nearly 1 MB/s for writes was not included because I think the time did not include waiting for the cache to flush at the end.
8. Time includes unoverlapped computation. Data taken from Table 6, line $N = 102$.
9. Figures 8, 9, and 11 exhibit approximately 10 MB/s. Higher rates are possible with interprocess locality in Mode 0. The Maxtor P1-17S disk drives support 2.5–3.6 MB/s (personal communication with Intel support engineers). I chose 2.5 MB/s conservatively.
10. Data taken from Table 2. The Maxtor P1-17S disk drives support 2.5–3.6 MB/s (personal communication with Intel support engineers). I chose 2.5 MB/s conservatively.
11. Raw disk performance unknown. Data taken from Table 3.
12. Raw disk performance unknown. Data taken from Table 3.
13. Raw disk performance unknown. Data are taken from Table 5.
14. Raw disk performance unknown. Data taken from Summary.
15. Raw disk performance unknown. Data taken from text accompanying Figure 8.
16. This is for CM-FORTRAN I/O. Performance under PARIS is better. Data are estimated from Figure 3. See [Kry92] for more details.
17. Time includes unoverlapped computation. This using "serial mode" format, which is much slower than the DataVault's native format. Data are taken from Table 3.
18. The numbers 1.7 and 32 are estimated from Figure 9, and 54.4 is calculated from that.
19. The numbers are taken from the text describing Figure 7.
20. Time includes unoverlapped computation. Data are taken from Table 2. Used only one disk (personal communication).

References

[Are91] James W. Arendt. Parallel genome sequence comparison using a concurrent file system. Technical Report UIUCDCS-R-91-1674, University of Illinois at Urbana-Champaign, 1991.

[AS89] Raymond K. Asbury and David S. Scott. FORTRAN I/O on the iPSC/2: Is there read after write? In *Fourth Conference on Hypercube Concurrent Computers and Applications*, pages 129–132, 1989.

[BCR92] Rajesh Bordawekar, Alok Choudhary, and Juan Miguel Del Rosario. An experimental performance evaluation of Touchstone Delta Concurrent File System. Technical Report SCCS-420, NPAC, Syracuse University, 1992. To appear, 1993 International Conference on Supercomputing.

[BN90] C. H. Baldwin and W. C. Nestlerode. A large scale file processing application on a hypercube. In *Fifth Annual Distributed-Memory Computer Conference*, pages 1400–1404, 1990.

[CCFN92] Russell Carter, Bob Ciotti, Sam Fineberg, and Bill Nitzberg. NHT-1 I/O benchmarks. Technical Report RND-92-016, NAS Systems Division, NASA Ames, November 1992.

[dBC93] Juan Miguel del Rosario, Rajesh Borawekar, and Alok Choudhary. Improved parallel I/O via a two-phase run-time access strategy. In *IPPS '93 Workshop on Input/Output in Parallel Computer Systems*, pages 56–70, 1993.

[DdR92] Erik DeBenedictis and Juan Miguel del Rosario. nCUBE parallel I/O software. In *Eleventh Annual IEEE International Phoenix Conference on Computers and Communications (IPCCC)*, pages 0117–0124, April 1992.

[dR92] Juan Miguel del Rosario. High performance parallel I/O on the nCUBE 2. *Institute of Electronics, Information and Communications Engineers (Transactions)*, August 1992. To appear.

[Dun91] T. H. Dunigan. Performance of the Intel iPSC/860 and Ncube 6400 hypercubes. *Parallel Computing*, 17:1285–1302, 1991.

[Fin93] Samuel A. Fineberg. Implementing the NHT-1 application I/O benchmark. In *IPPS '93 Workshop on Input/Output in Parallel Computer Systems*, pages 37–55, 1993.

[FPD93] James C. French, Terrence W. Pratt, and Mriganka Das. Performance measurement of the Concurrent File System of the Intel iPSC/2 hypercube. *Journal of Parallel and Distributed Computing*, 17(1–2):115–121, January and February 1993.

[KN93] John Krystynak and Bill Nitzberg. Performance characteristics of the iPSC/860 and CM-2 I/O systems. In *Proceedings of the Seventh International Parallel Processing Symposium*, pages 837–841, 1993.

[Kry92] John Krystynak. I/O performance on the Connection Machine DataVault system. Technical Report RND-92-011, NAS Systems Division, NASA Ames, May 1992.

[LIN+93] Susan J. LoVerso, Marshall Isman, Andy Nanopoulos, William Nesheim, Ewan D. Milne, and Richard Wheeler. *sfs*: A parallel file system for the CM-5. In *Proceedings of the 1993 Summer Usenix Conference*, 1993. To appear.

[MK91] Ethan L. Miller and Randy H. Katz. Input/output behavior of supercomputer applications. In *Proceedings of Supercomputing '91*, pages 567–576, November 1991.

[Nit92] Bill Nitzberg. Performance of the iPSC/860 concurrent file system. Technical Report RND-92-020, NAS Systems Division, NASA Ames, December 1992.

[PFDJ89] Terrence W. Pratt, James C. French, Phillip M. Dickens, and Stanley A. Janet, Jr. A comparison of the architecture and performance of two parallel file systems. In *Fourth Conference on Hypercube Concurrent Computers and Applications*, pages 161–166, 1989.

[Pie89] Paul Pierce. A concurrent file system for a highly parallel mass storage system. In *Fourth Conference on Hypercube Concurrent Computers and Applications*, pages 155–160, 1989.

[SBMS93] Margo Seltzer, Keith Bostic, Marshall Kirk McKusick, and Carl Staelin. An implementation of a log-structured file system for Unix. In *Proceedings of the 1993 Winter Usenix Conference*, pages 295–306, January 1993.