File System Workload Analysis For Large Scale Scientific Computing Applications

Feng Wang, Qin Xin, Bo Hong, Scott A. Brandt, Ethan L. Miller, Darrell D.E. Long

Storage System Research Center
University of California, Santa Cruz

Tyce T. McLarty

Lawrence Livermore National Laboratory

NASA/IEEE MSST 2004
12th NASA Goddard/21st IEEE Conference on
Mass Storage Systems & Technologies
The Inn and Conference Center
University of Maryland University College
Adelphi MD USA
April 13-16, 2004

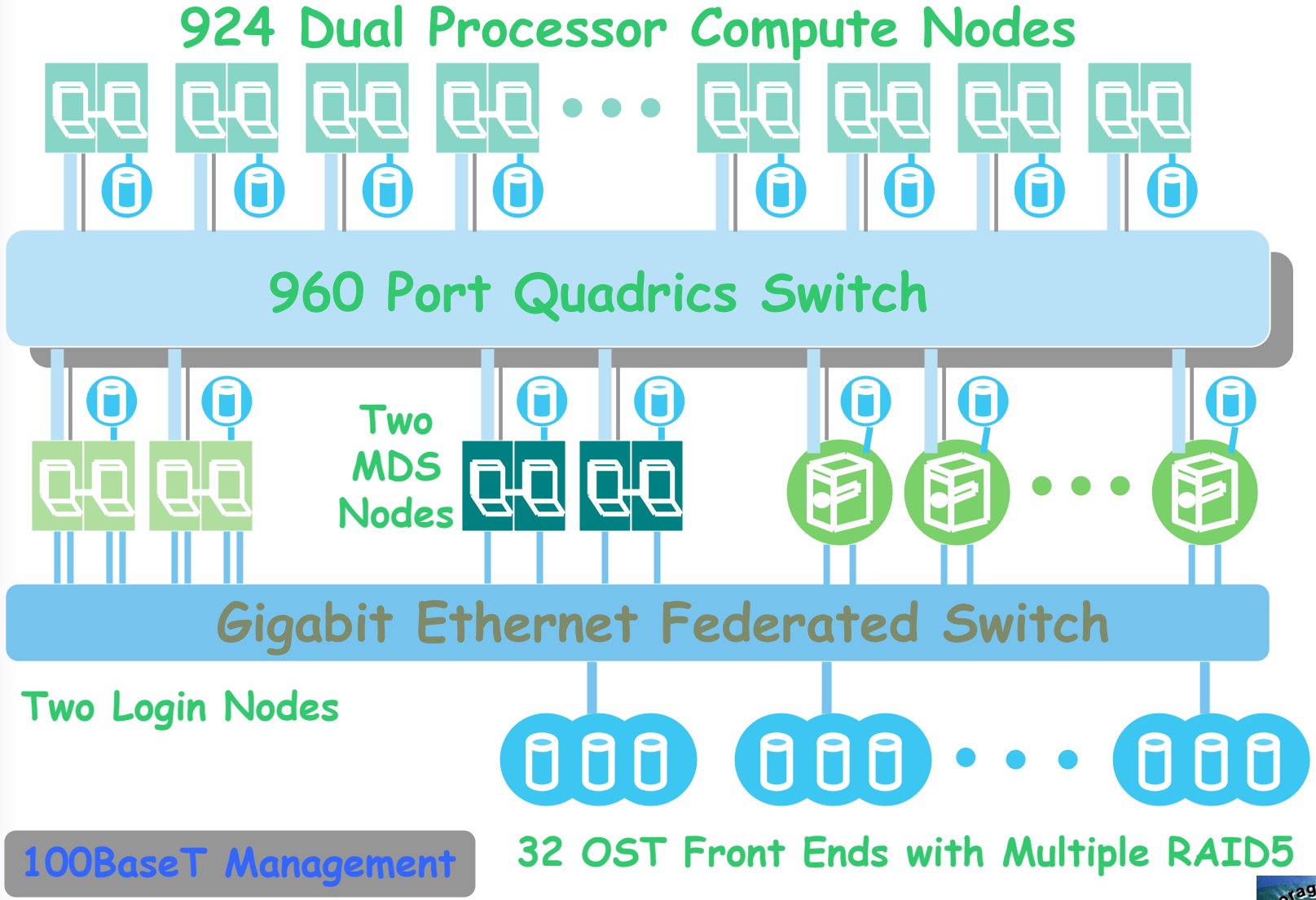
Motivations

- Modern parallel scientific applications require high-performance I/O support.
- The I/O access patterns of the scientific applications keep changing in light of the technologies advancement.
- Understanding the expected workloads from typical applications is essential for designing a parallel file system.

Questions

- What is the file size distribution? What is the average file life time?
- What is the I/O request size distribution and how does it change over time?
- How bursty are the I/O requests?
- How are the files opened? What are the typical file access patterns?

System Under Study



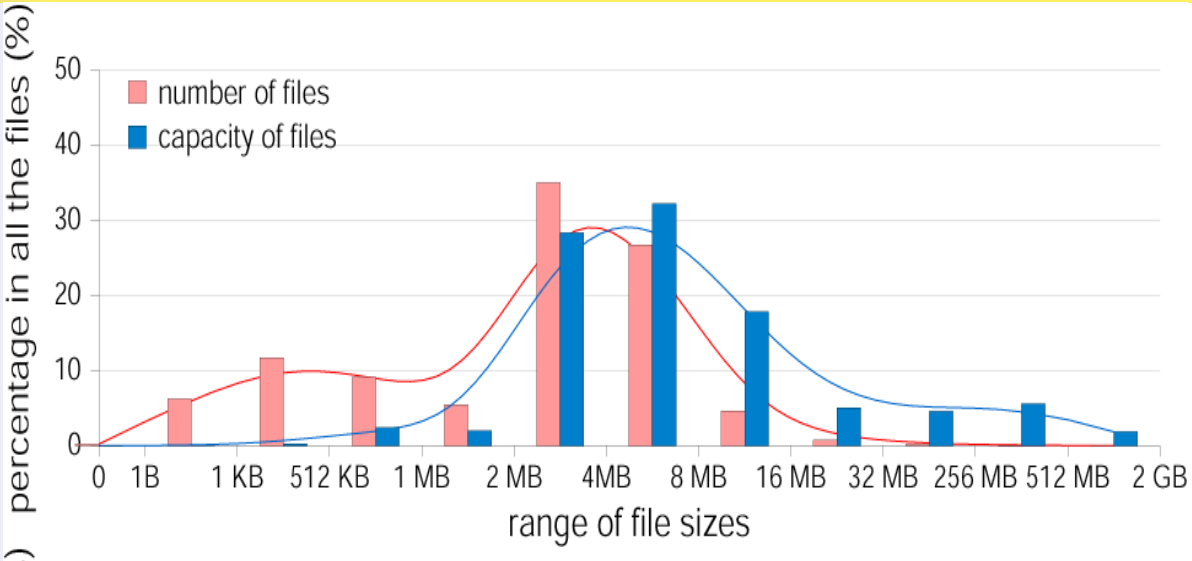
Data Collection

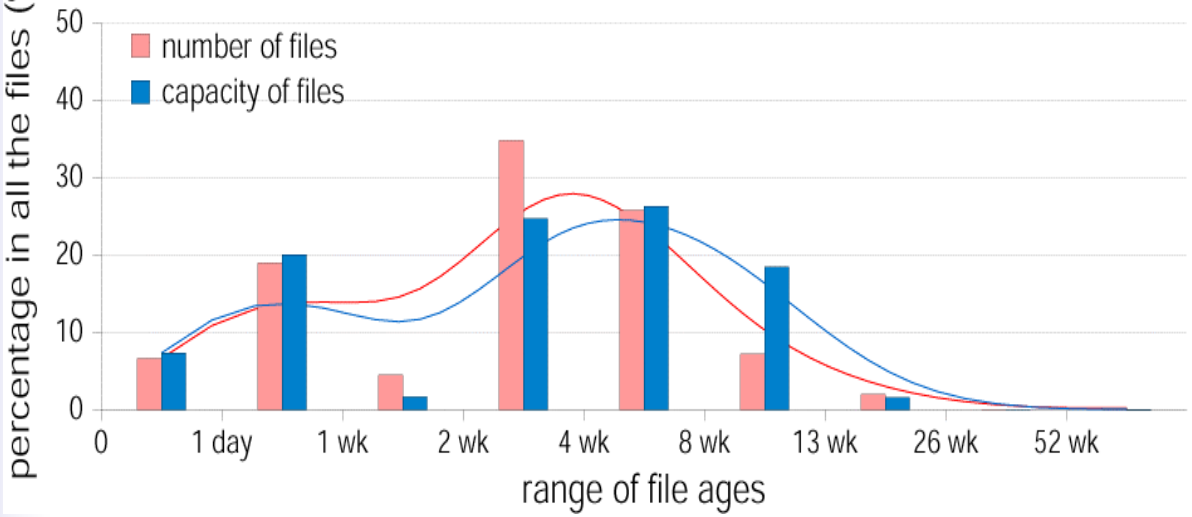
- Use ***strace*** utility with parameters tuned for tracing file system related activities
- Shortcut computation phases to minimize the tracing time
- Dump traces to local disks in individual files for each node
- The time of trace records are globally synchronized
  - Quadrics switch has a common clock

Applications and Traces

Applications	# of Nodes	Traces	Properties
File System Benchmark ior2	512 (single)	ior2-fileProc	Individual output per node
		ior2-shared	Shared file; Contiguous region
		ior2-stride	Shared file; Stride blocks
Physical Application f1	343 (single)	f1-write	Results dump phase; Master Node collects writes
		f1-restart	Restart phase; Read dominates
Physical Application m1	810 (dual)	m1-write	Results dump phase; Large sequential writes
		m1-restart	Restart phase; Very large reads; Large sequential writes

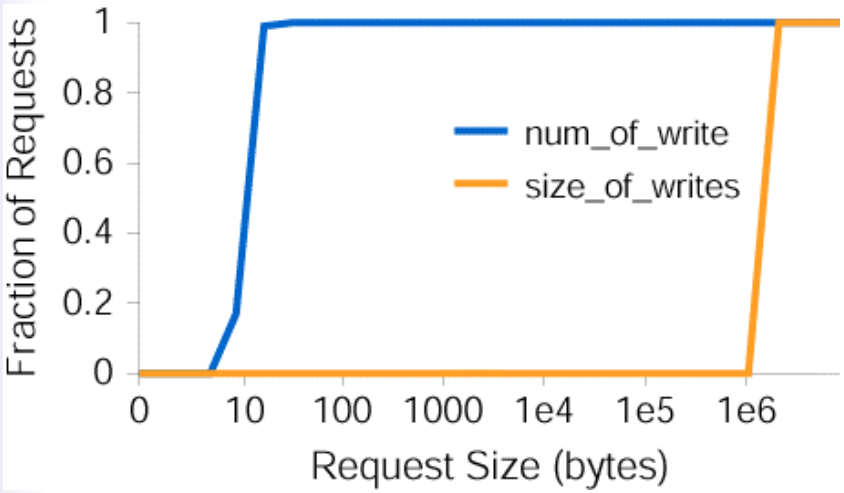
File Distribution



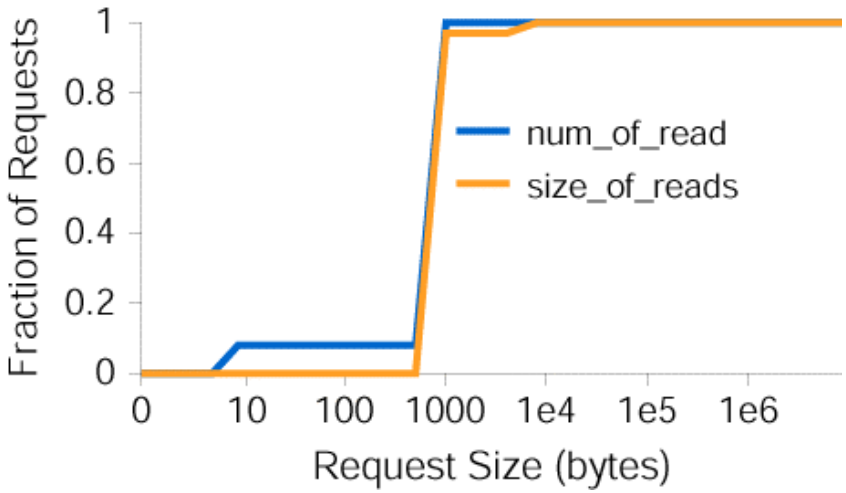


- Total number of files ~ 9.7 million
- Total capacity of files ~ 33 TB

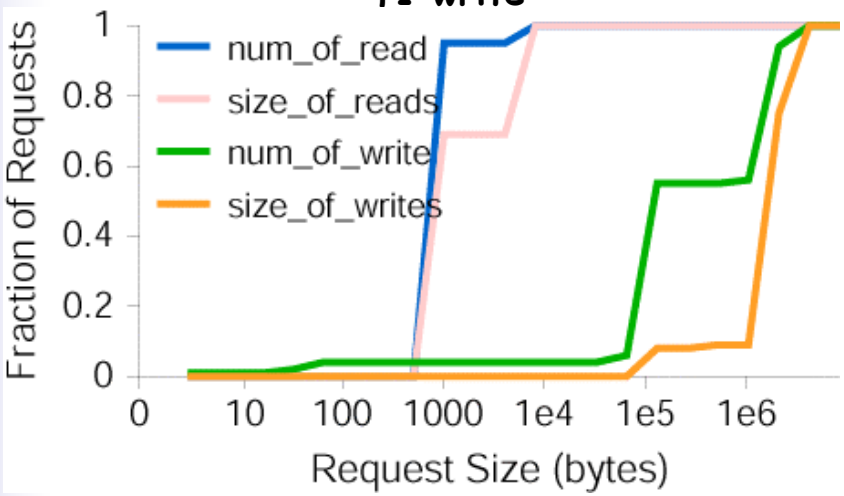
I/O Request Size



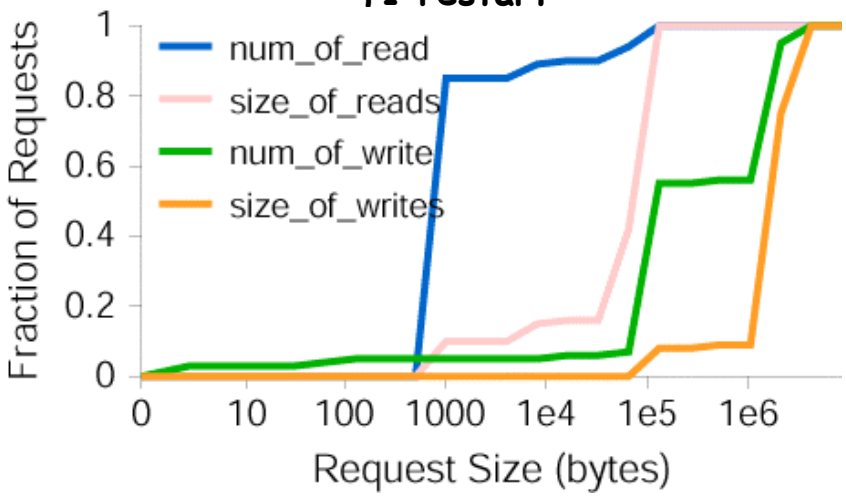
f1 write



f1 restart

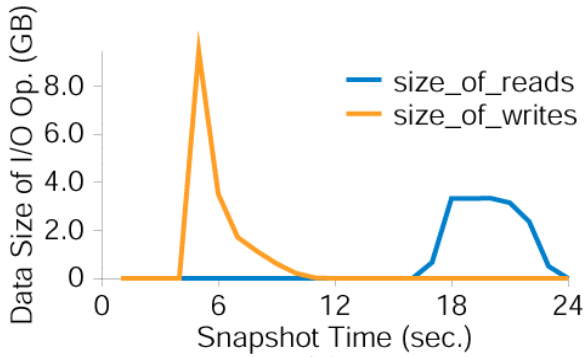


m1 write

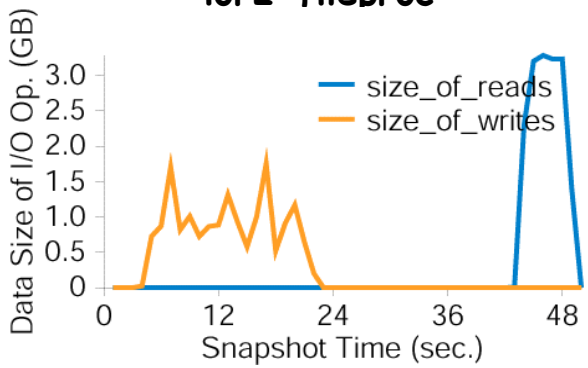


m1 restart

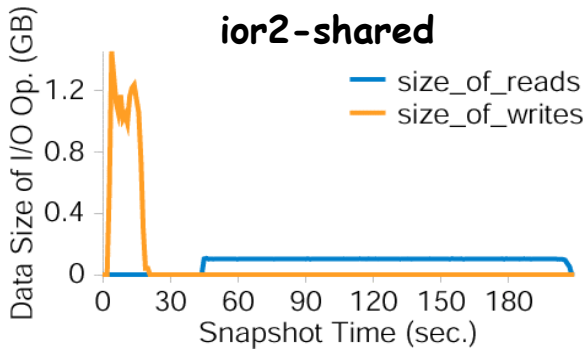
I/O Access Characteristics



ior2-filepproc



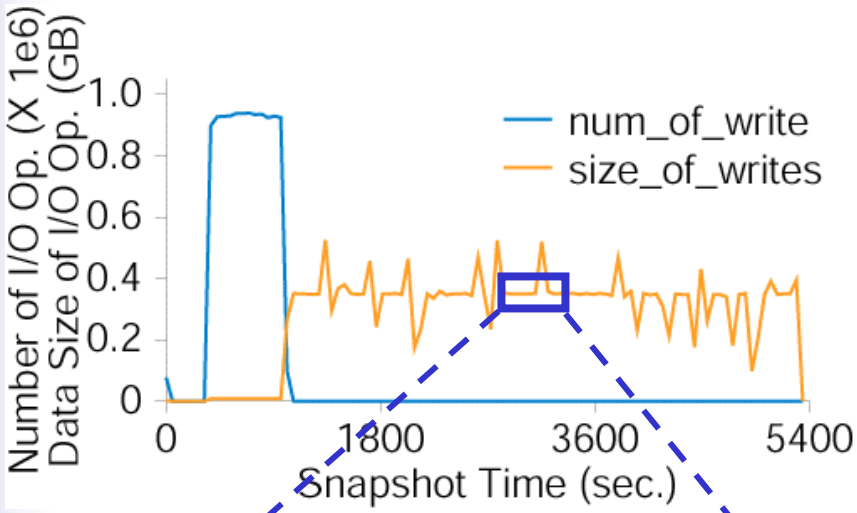
ior2-shared



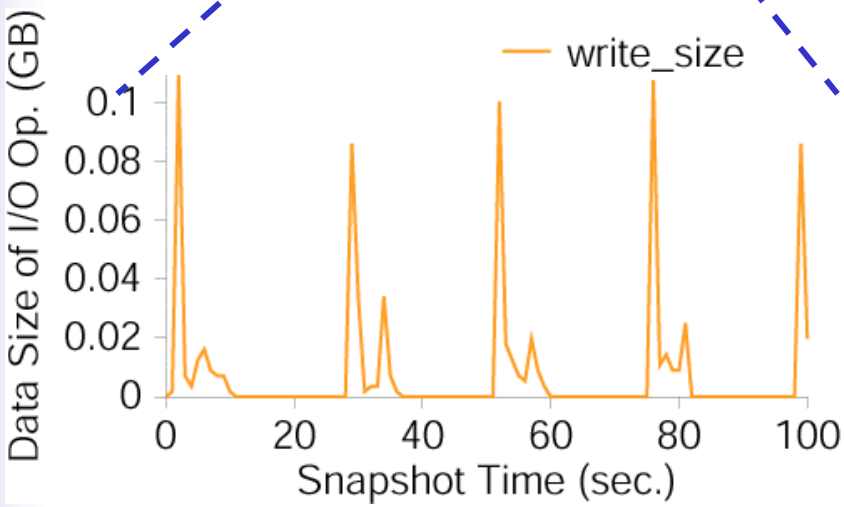
ior2-stride

- Each node begins with large sequential writes and then reads back another node's output to verify the data.
- The shared-file configurations decrease the bandwidth utilization by factors from 5 to 10.

I/O Access Characteristics



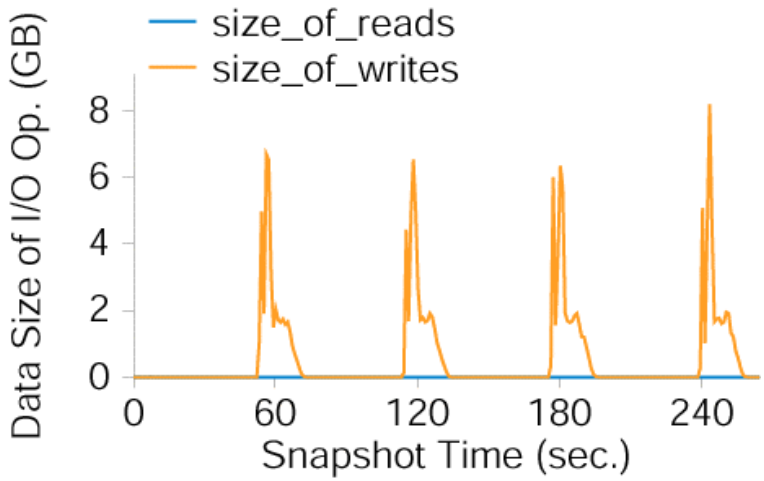
f1-write



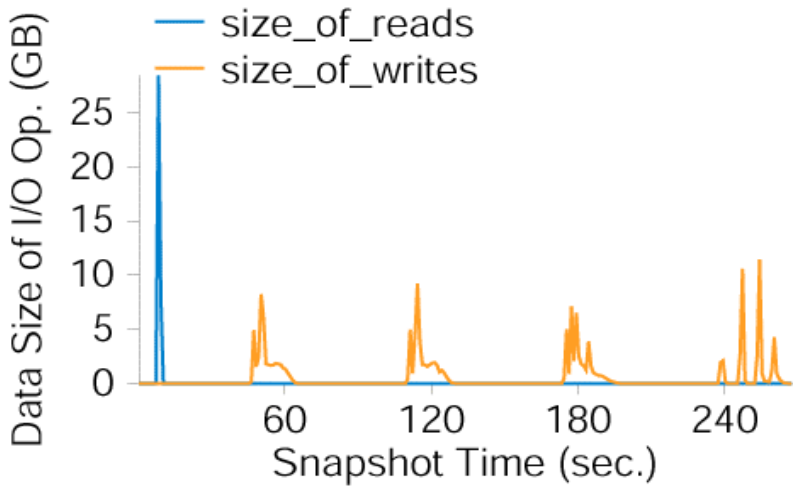
f1-write-short

- One master node collects very small writes (tens of bytes) from the rest of the cluster.
- After small writes, a group of nodes (48) dump results in very large chunks into a shared file.
- Writes are very bursty, interleaved with long computation phases.

I/O Access Characteristics



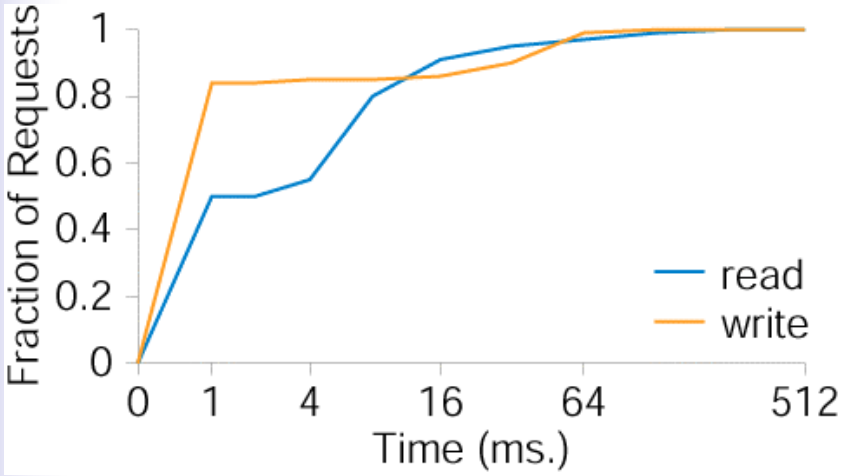
m1-write



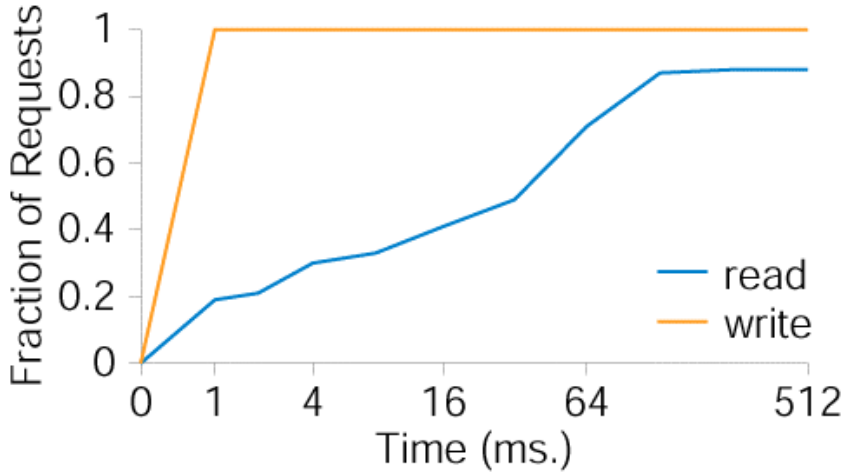
m1-restart

- 1620 processors create result files simultaneously.
- Results are dumped to file in very large chunks.
- The write curves show the similar shape.

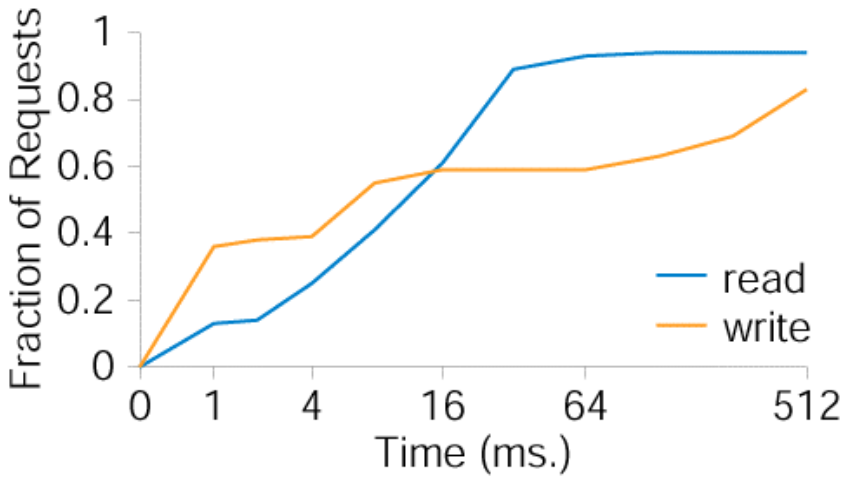
I/O Burstiness



ior2-fileproc



f1-write



m1-write

File Opens

Applications	Overall # of file opens			# of Data File Opens		
	R/W	R-Only	W-Only	R/W	R-Only	W-Only
ior2	6,656	5,121	0	1,024	0	0
f1-write	3,871	6,870	718	98	10	34
f1-restart	3,773	6,179	0	0	343	0
m1-write	17,824	22,681	12,960	0	1,620	12,960
m1-restart	17,824	21,061	12,960	0	0	12,960

File Opens (Cont.)

Applications	Avg. open time		Avg. I/Os per open		Avg. I/O size per open	
	Overall	Data file	Overall	Data file	Overall	Data file
ior2-fileproc	.4 sec	4.5 sec	44.4	512.0	2.8 MB	32.8 MB
ior2-shared	.7 sec	5.2 sec	44.4	512.0	2.8 MB	32.8 MB
ior2-stride	7.6 sec	26.5 sec	44.4	512.0	2.8 MB	32.8 MB
f1-write	20.2 sec	504 sec	14.8	142161	2.4 MB	3393 MB
f1-restart	.02 sec	.1 sec	.5	1	<< 1 MB	<< 1 MB
m1-write	1.2 sec	3.9 sec	4.2	15.3	3.7 MB	8.5 MB
m1-restart	1.2 sec	2.4 sec	4.3	17	3.1 MB	6.5 MB

Conclusions

- Each application has only one or two typical request sizes.
- Large requests from several hundred KBs to several MBs are very common.
- Almost all I/O data are transferred through large requests.
- All applications show very bursty access patterns.
- Lustre file system is not well optimized for file sharing.
- Data files are usually opened for a relatively long time, and a large amount of I/Os are performed during each open.

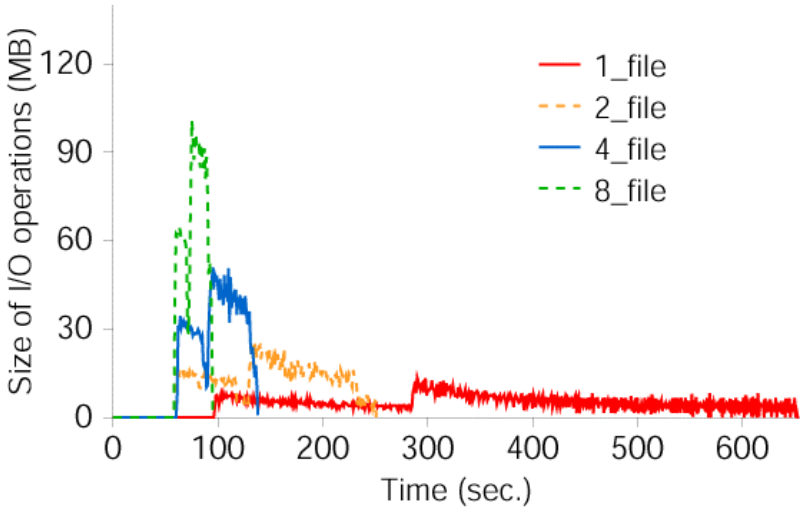
Acknowledge

- This research is supported by Lawrence Livermore National Laboratory, Los Alamos National Laboratory, and Sandia National Laboratory.

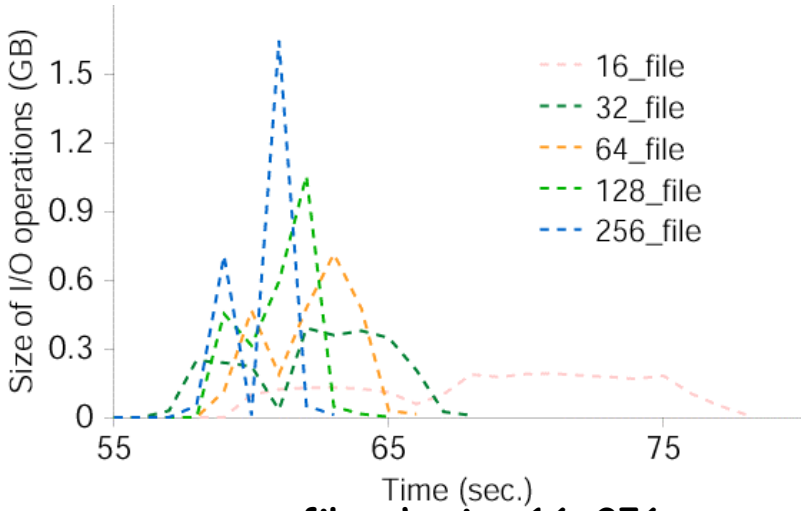
- We are also grateful to our sponsors: National Science Foundation, USENIX Association, Hewlett Packard Laboratories, IBM Research, Intel Corporation, Microsoft Research, ONStar, Overland Storage, and Veritas.

Backup Slides

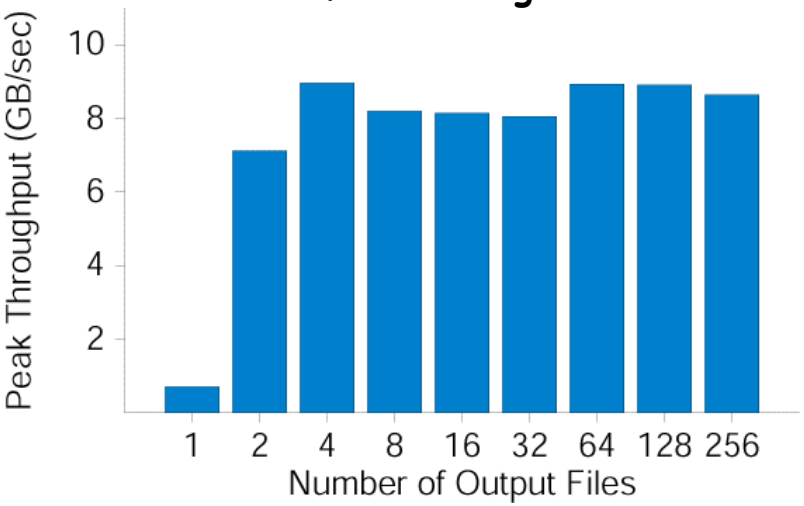
File Sharing Traces



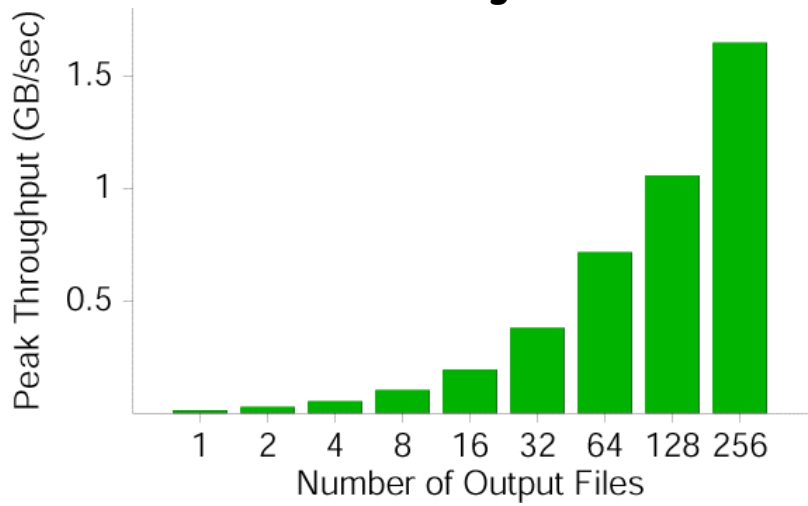
file-sharing 1~8



file-sharing 16~256

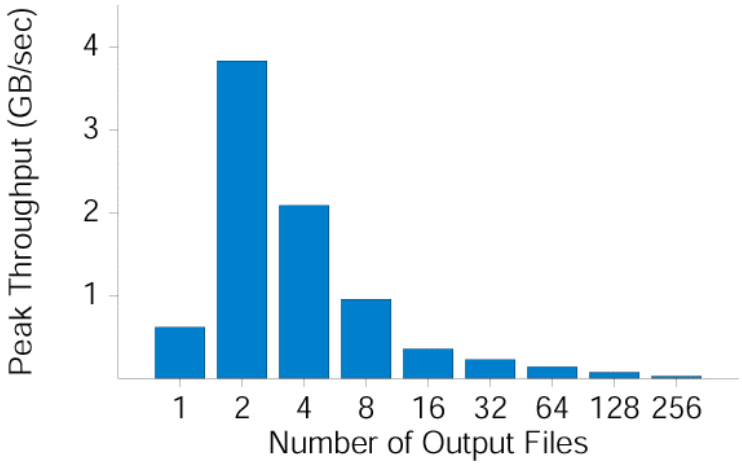


Read peak throughput

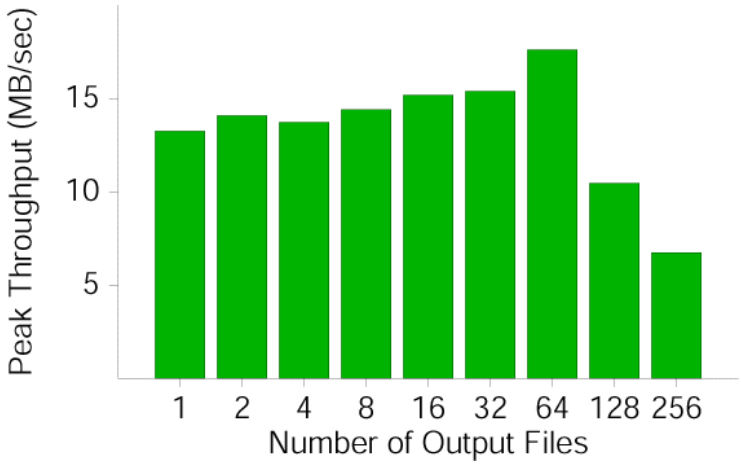


Write peak throughput

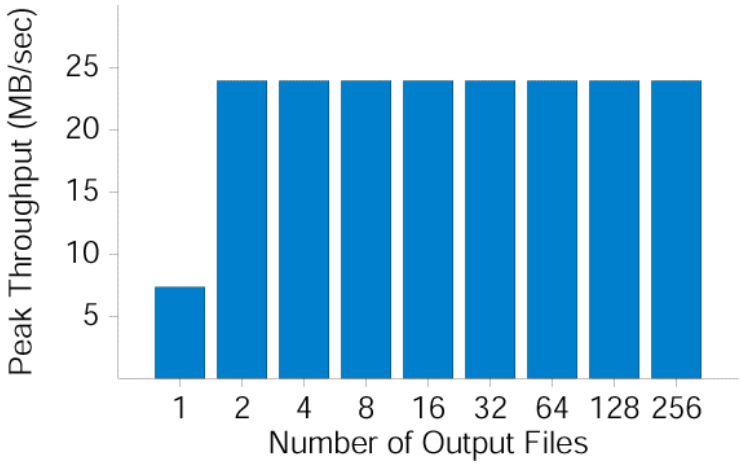
File Sharing Traces



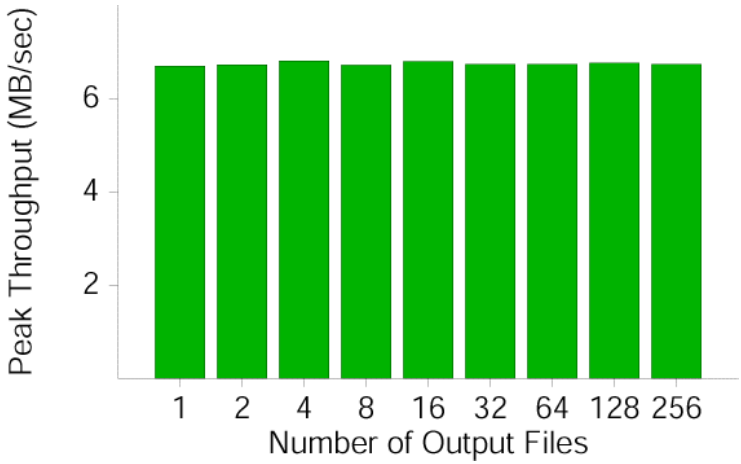
Read peak throughput per file



Write peak throughput per file

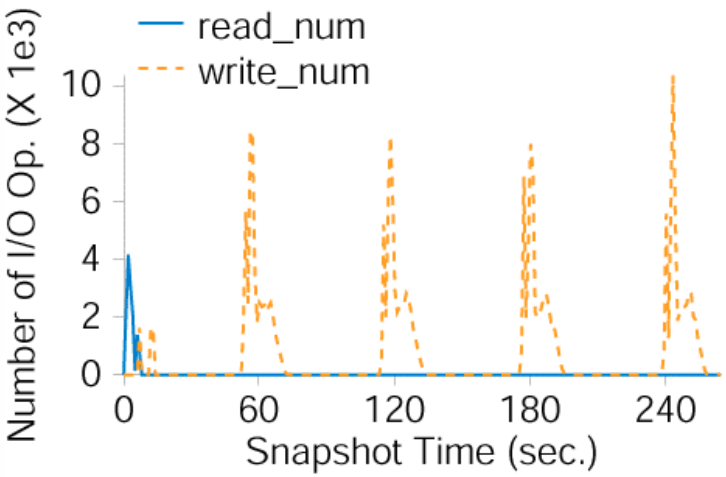


Read peak throughput per node

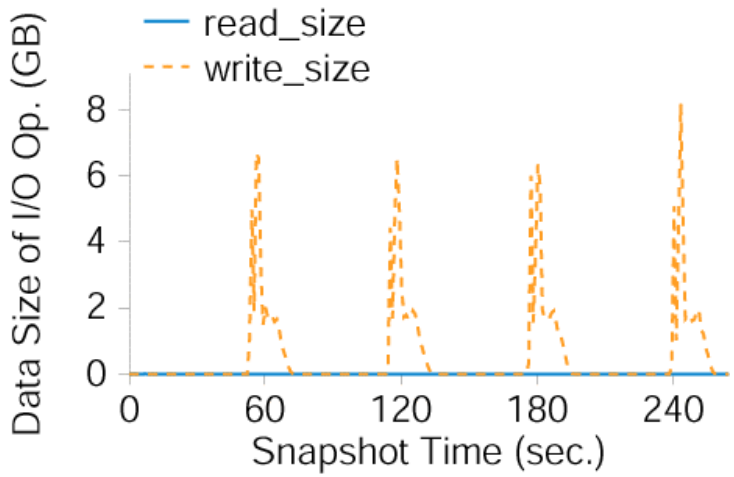


Write peak throughput per node

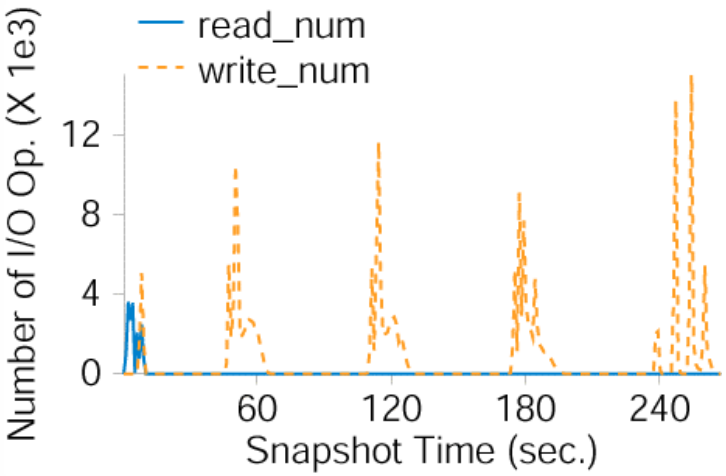
I/O Requests Characteristics



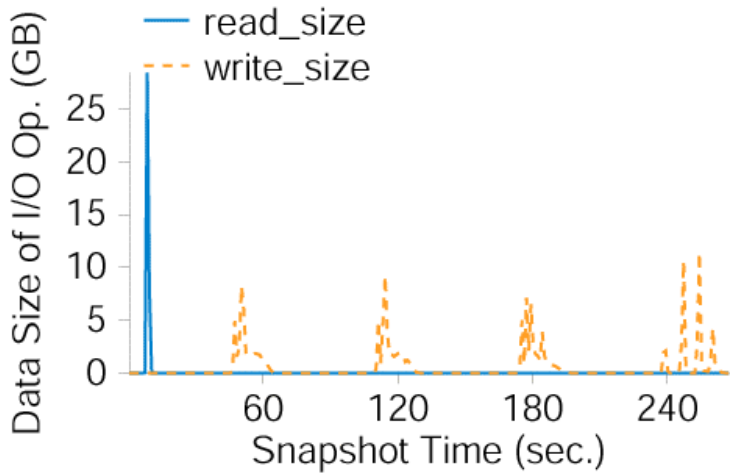
m1-write-num



m1-write-size

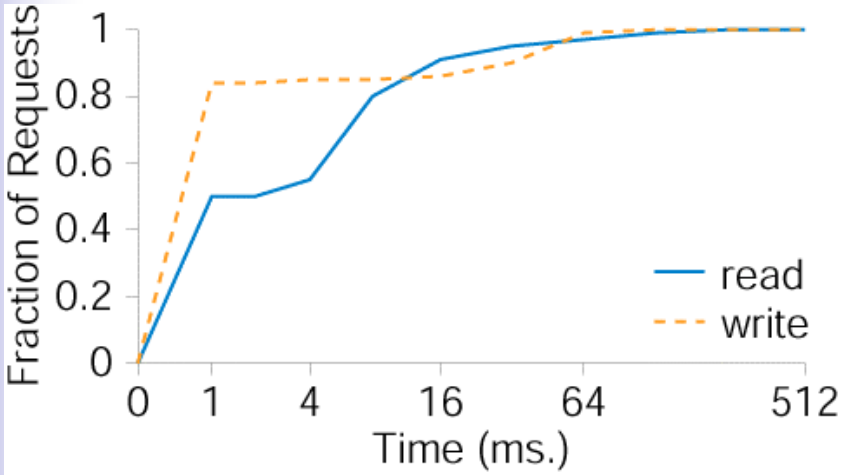


m1-restart-num

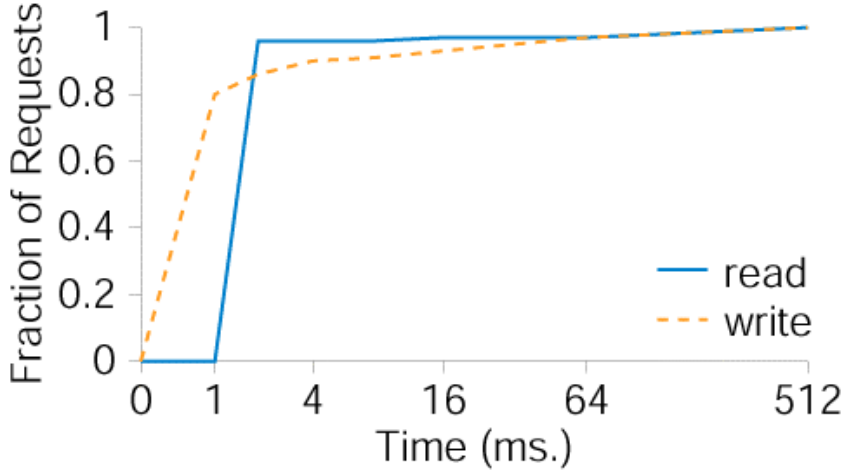


m1-restart-size

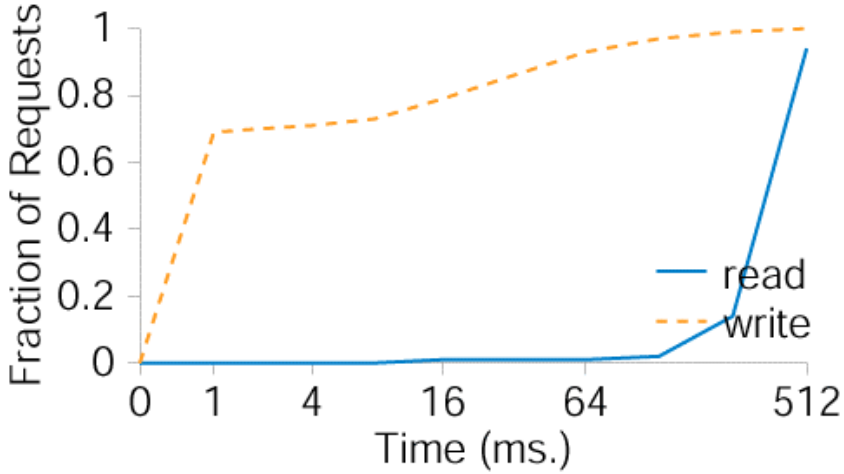
I/O Burstiness – ior2 Benchmark



ior2-fileproc

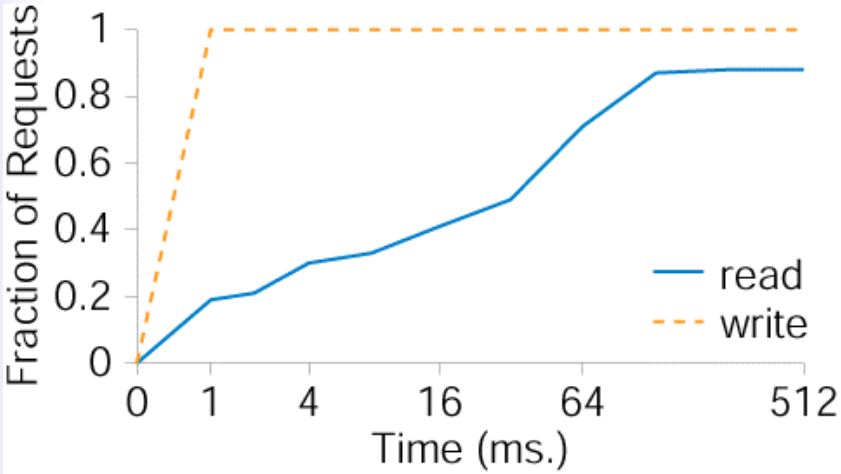


ior2-shared

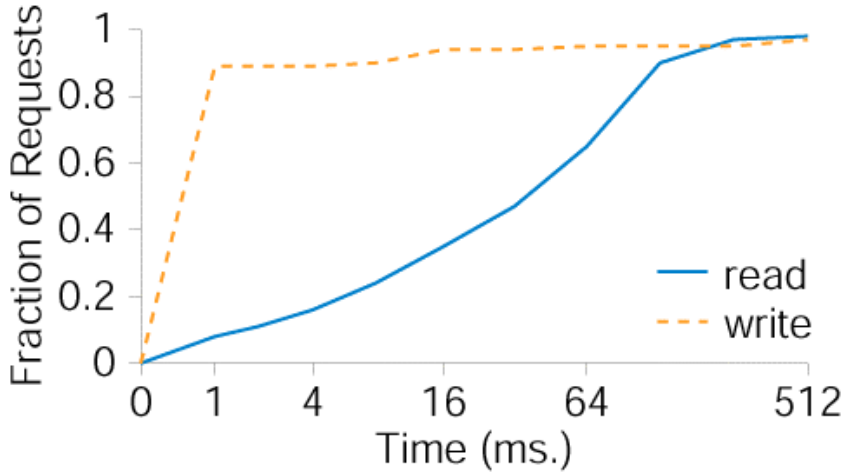


ior2-stride

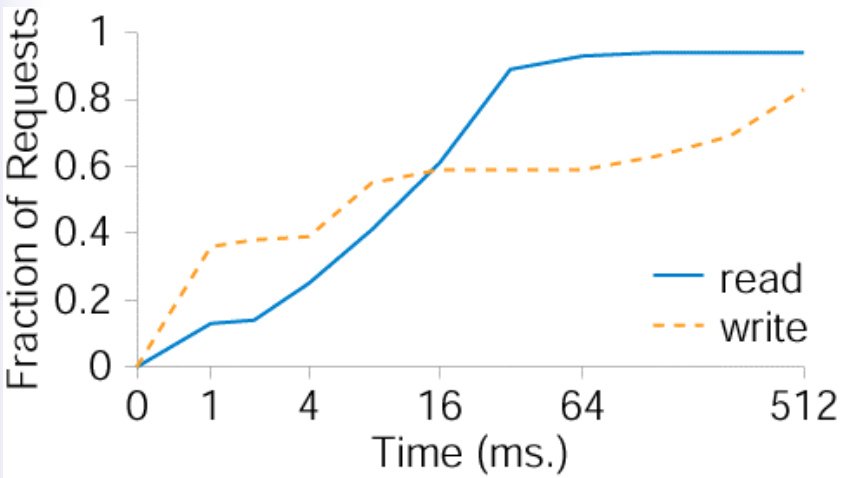
I/O Burstiness – f1 and m1



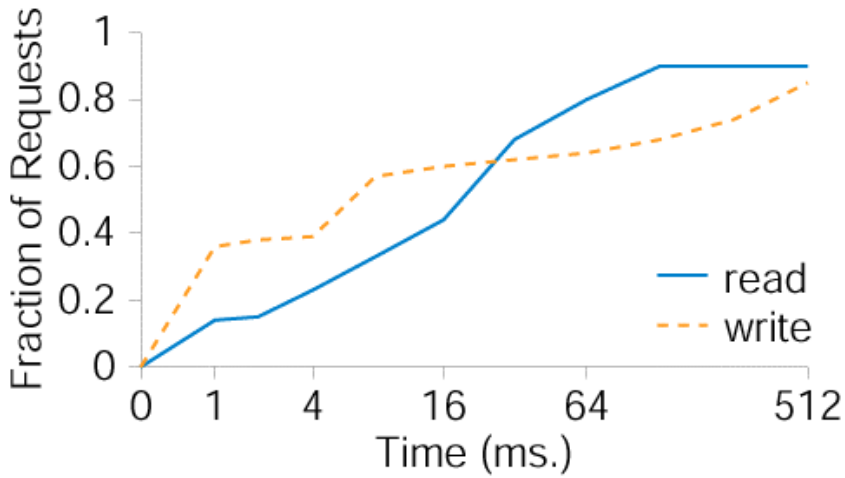
f1-write



f1-restart



m1-write



m1-restart

File System Workload Analysis For Large Scale Scientific Computing Applications

Feng Wang, Qin Xin, Bo Hong, Scott A. Brandt,
Ethan L. Miller, Darrell D. E. Long,
University of California, Santa Cruz
Tyce T. McLarty
Lawrence Livermore National Laboratory

UC Santa Cruz


Storage Systems Research Center