I/O Study at GSFC

Joe Schmidt (CSC),
Bruce Pfaff (GSFC),
Hank Jakiela (HP),
Eugene Gooden (HP)

September 21, 2003

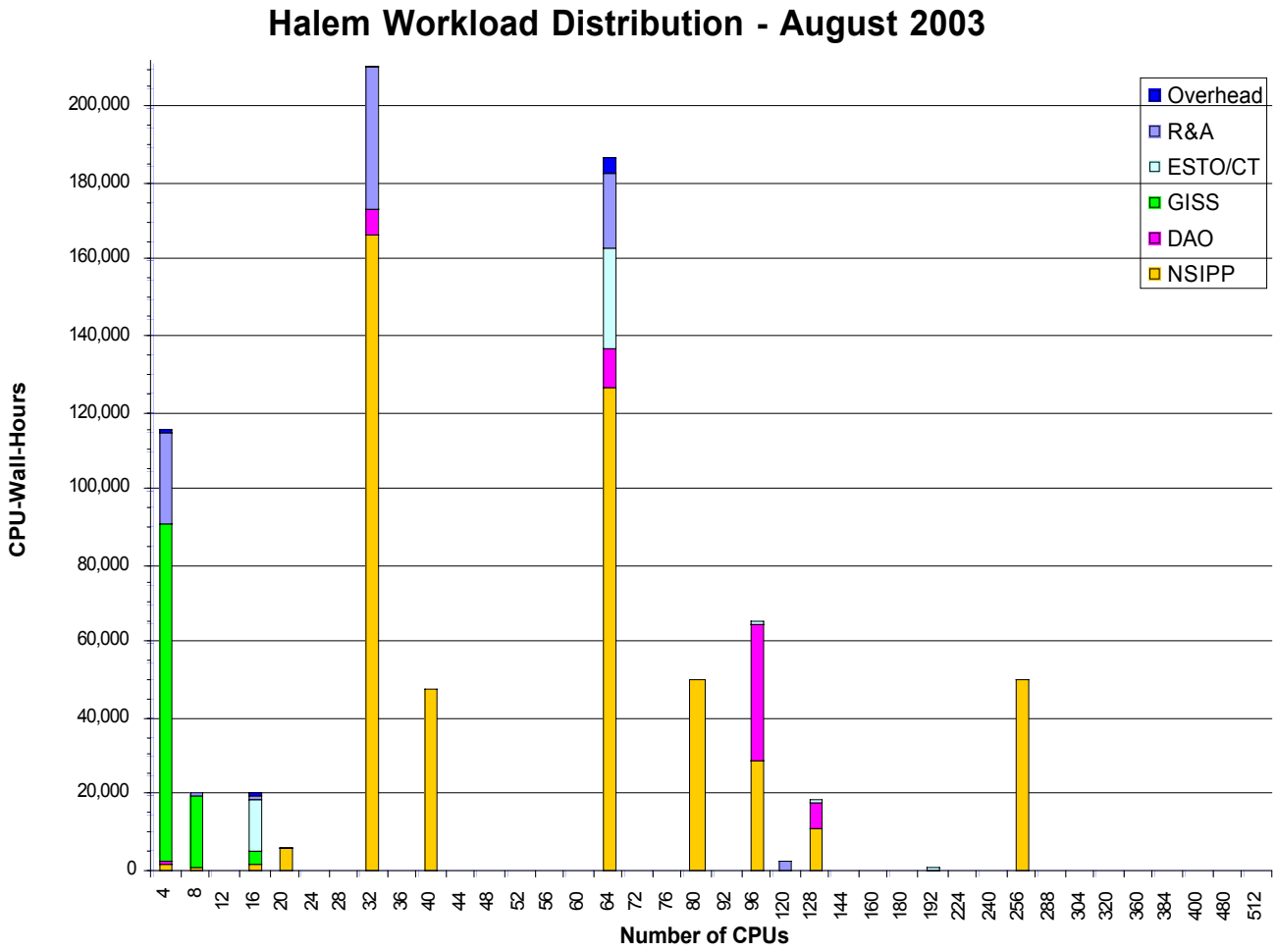
HP SC45 at GSFC

- 1392 processors (880@1.25Ghz, 512@1.00Ghz)
- 8.5 TB of disk (HSG controllers)
- Quadrics
- GigE

I/O Performance is a Critical Bottleneck for Our Users

1. How do we make the I/O more consistent?
2. How do we make the I/O faster?

Workload – Production Runs



I/O Load

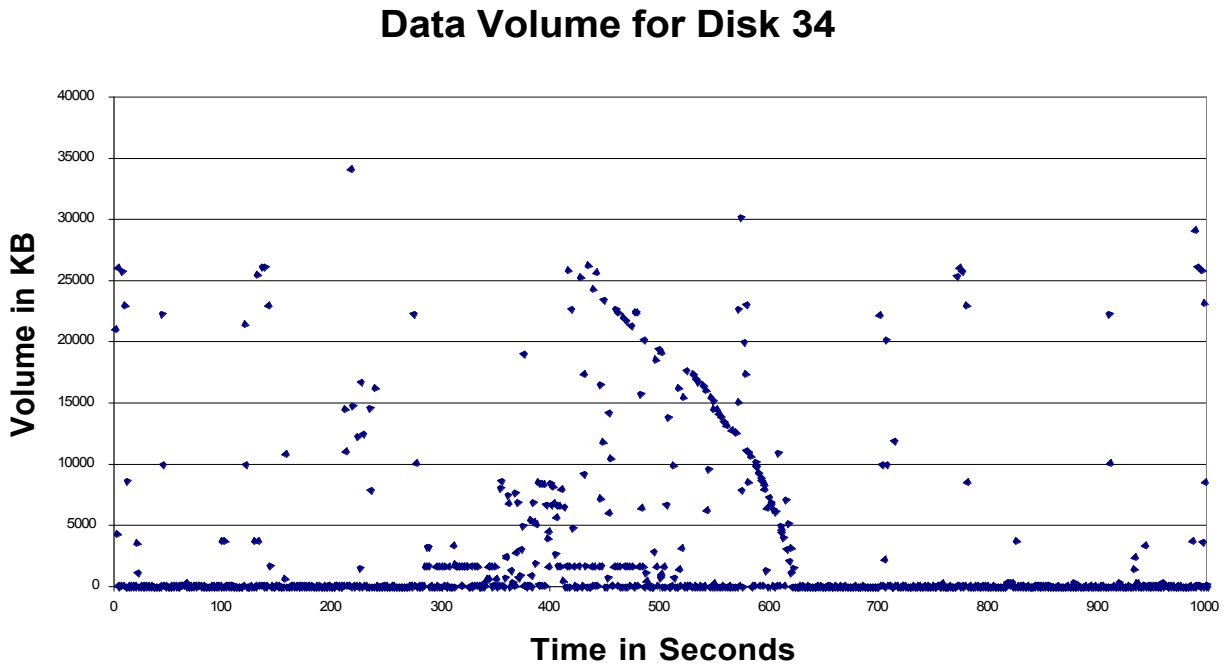
- User perception is large I/O.
  - Estimates are based on file size.
  - Typical file size is 60-700 MB.
- Measured I/O size is smaller.
  - I/O size is governed by block size of the operation and disk fragmentation.

Measure I/O Size With Collect

#DSK	NAME	B/T/L	R/S	RKB/S	W/S	WKB/S	AVS	AVW	ACTQ	WTQ	%BSY
0	dsk68	0/0/0	0	0	0	0	0	0	0	0	0
1	dsk69	0/1/0	0	0	0	0	0	0	0	0	0
2	dsk34	-	46	2813	72	4331	12.2	4.23	1.46	0.01	68.93
3	dsk35	-	0	0	0	0	0	0	0	0	0
4	dsk24	-	19	159	1	15	5.33	0	0.12	0	10.99
5	dsk25	-	0	0	0	0	0	0	0	0	0
6	dsk27	-	0	0	0	0	0	0	0	0	0
7	dsk28	-	0	0	0	0	0	0	0	0	0
8	dsk9	-	0	0	0	0	0	0	0	0	0
9	dsk10	-	0	0	0	0	0	0	0	0	0
10	dsk30	-	0	0	0	0	0	0	0	0	0
11	dsk31	-	0	0	0	0	0	0	0	0	0
12	dsk17	-	0	0	0	0	0	0	0	0	0
13	dsk18	-	0	0	0	0	0	0	0	0	0
14	dsk20	-	0	0	0	0	0	0	0	0	0
15	dsk21	-	0	0	0	0	0	0	0	0	0
16	dsk32	-	32	1454	2	31	4.1	0	0.15	0	55.95
17	dsk33	-	0	0	0	0	0	0	0	0	0
18	dsk61	-	32	3108	138	5994	7.8	6.67	1.34	0.04	58.94
19	dsk91	-	0	0	0	0	0	0	0	0	0
20	dsk6	-	2	23	4	175	4.64	0	0.04	0	67.93

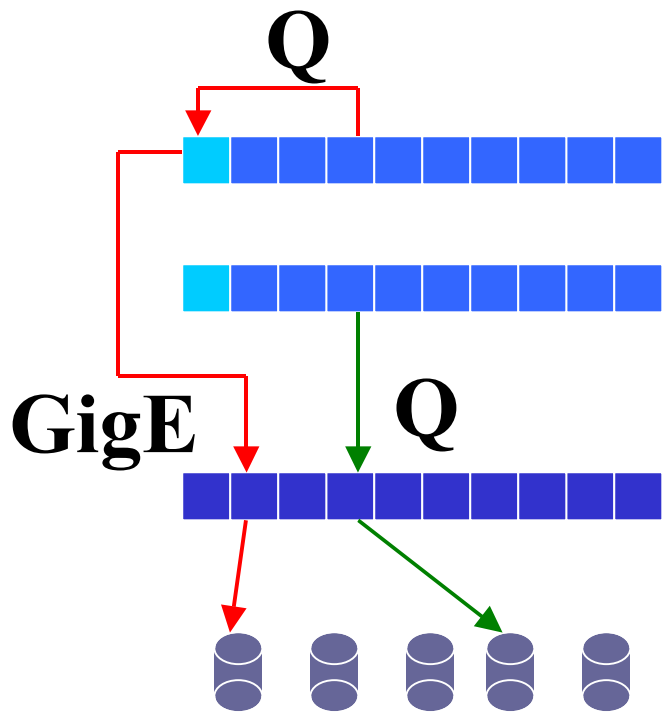
Examine individual results.

- Collect with showfile.



File System Alternatives

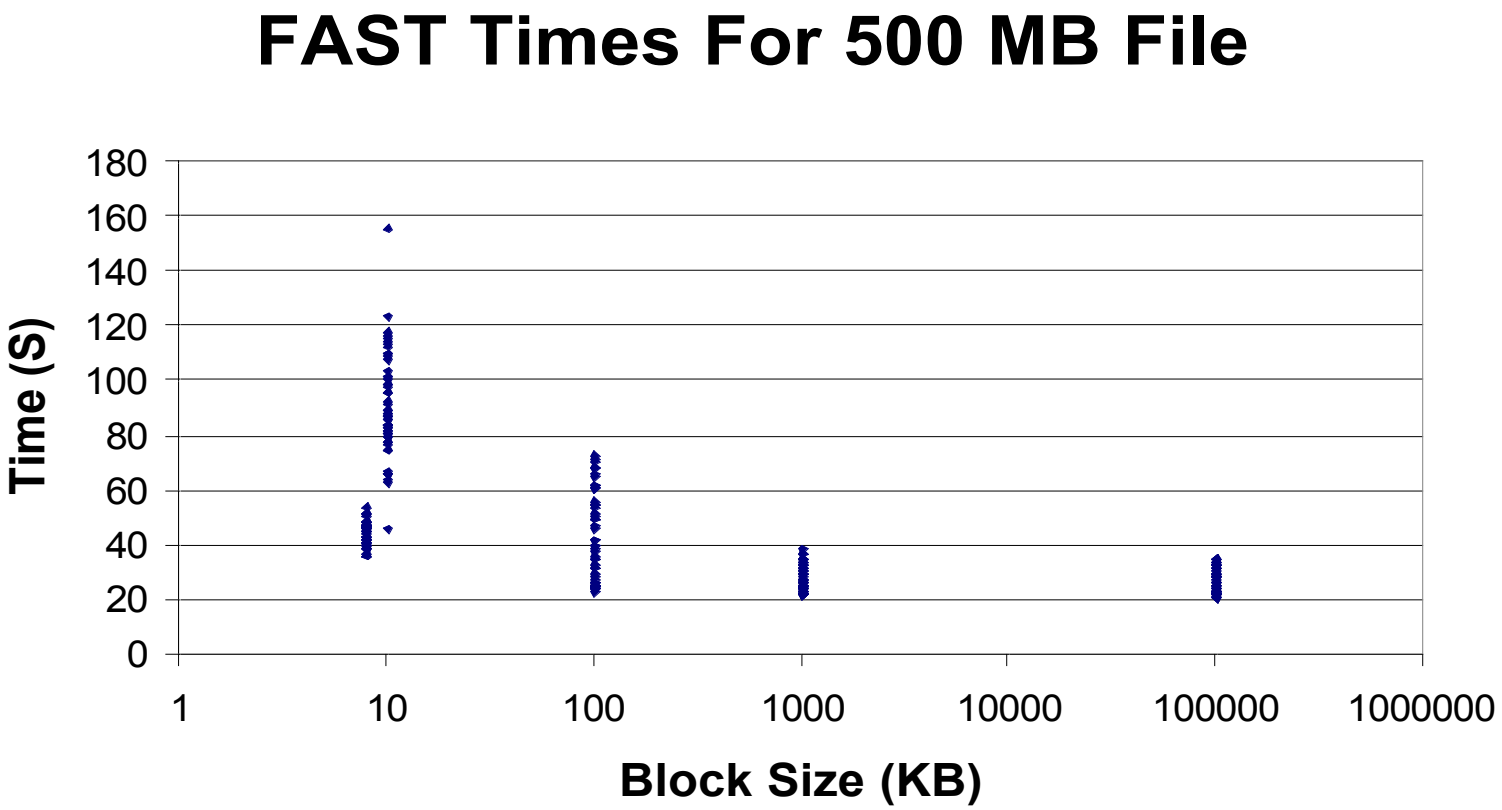
- PFS
- SCFS FAST
- SCFS UBC



PFS

- Useful when many streams write to a file system at once.
- Observed load consists of 1 to 3 streams at a time to a file system.
- Future adoption of parallel I/O techniques would make this viable.

FAST is Appropriate for Half Our Workload



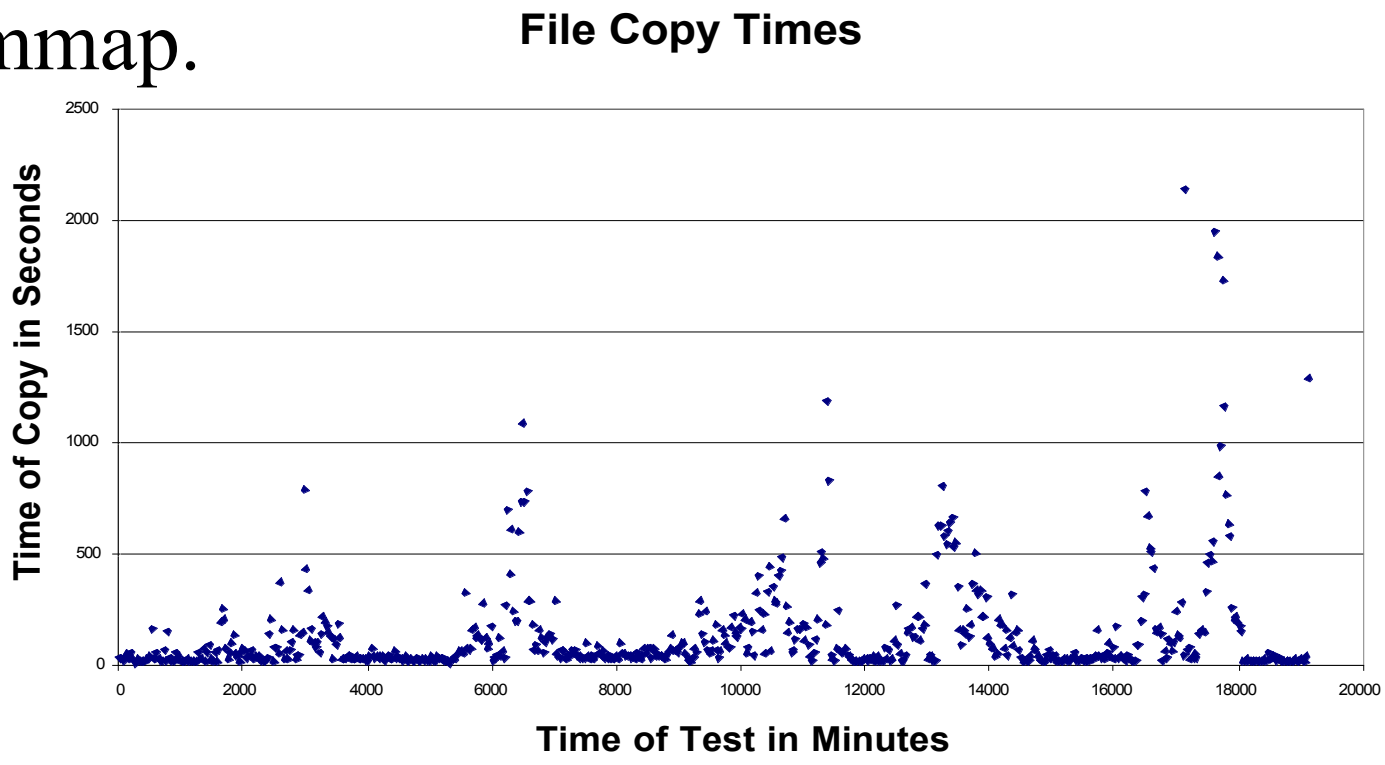
FAST upside

- Bypasses head node.
- Scales with disk performance.
- Uses Quadrics.

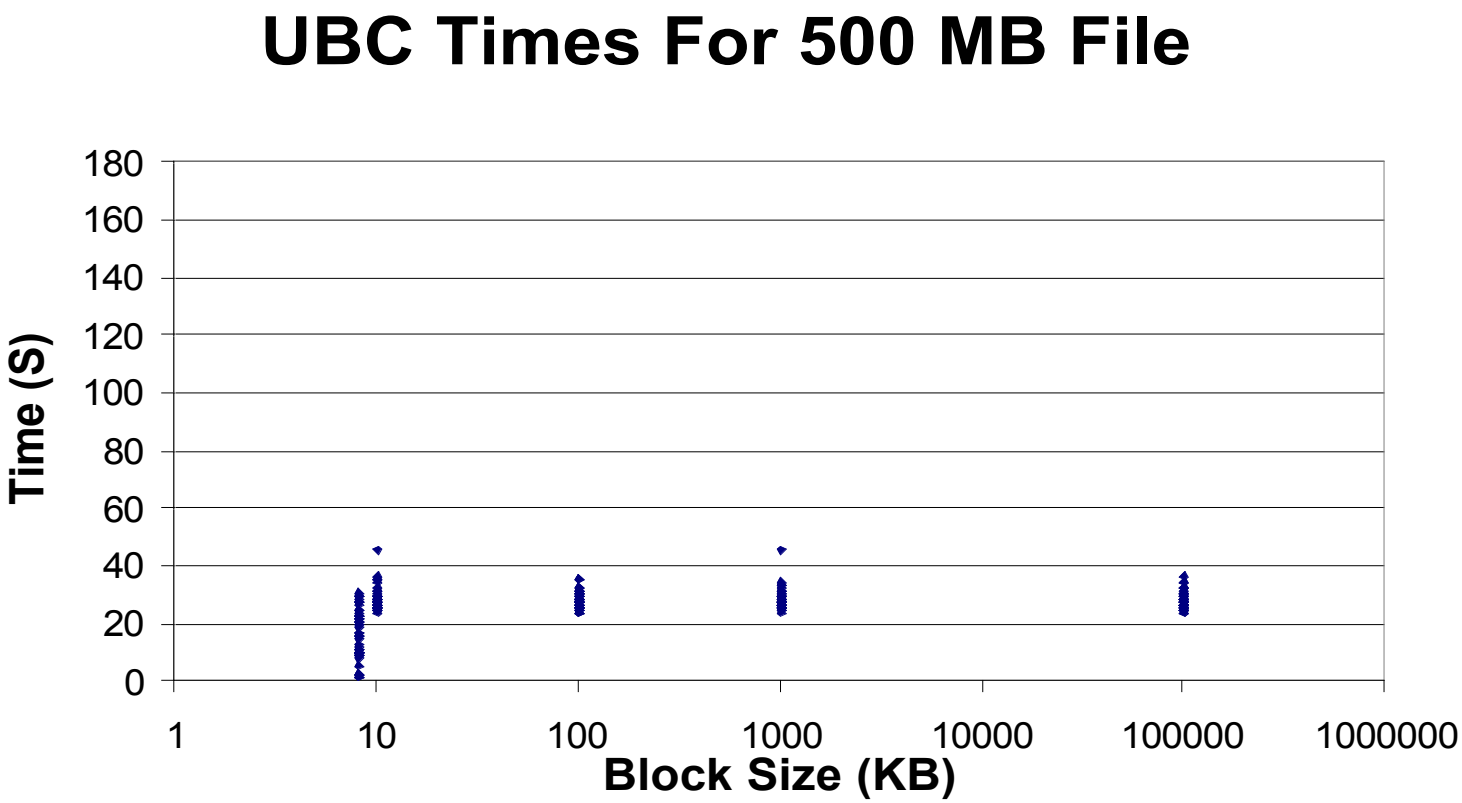
time	rd/s	rdkb	wr/s	wrkb
20:32:16	7	503	0	0
20:32:17	9	623	0	0
20:32:18	16	622	0	7
20:32:19	6	495	0	0
20:32:20	8	751	1	23
20:32:21	3	375	74	6363
20:32:22	6	503	585	59118
20:32:23	2	375	545	52740
20:32:24	8	751	89	10102
20:32:25	4	495	26	2685
20:32:26	1	207	184	20076
20:32:27	0	0	123	8559
20:32:28	0	0	87	7529
20:32:29	0	0	79	8479
20:32:30	0	0	95	8503
20:32:31	0	0	110	8855

FAST downside

- Severely penalizes small block sizes.
- Suffers with fragmentation.
- No mmap.



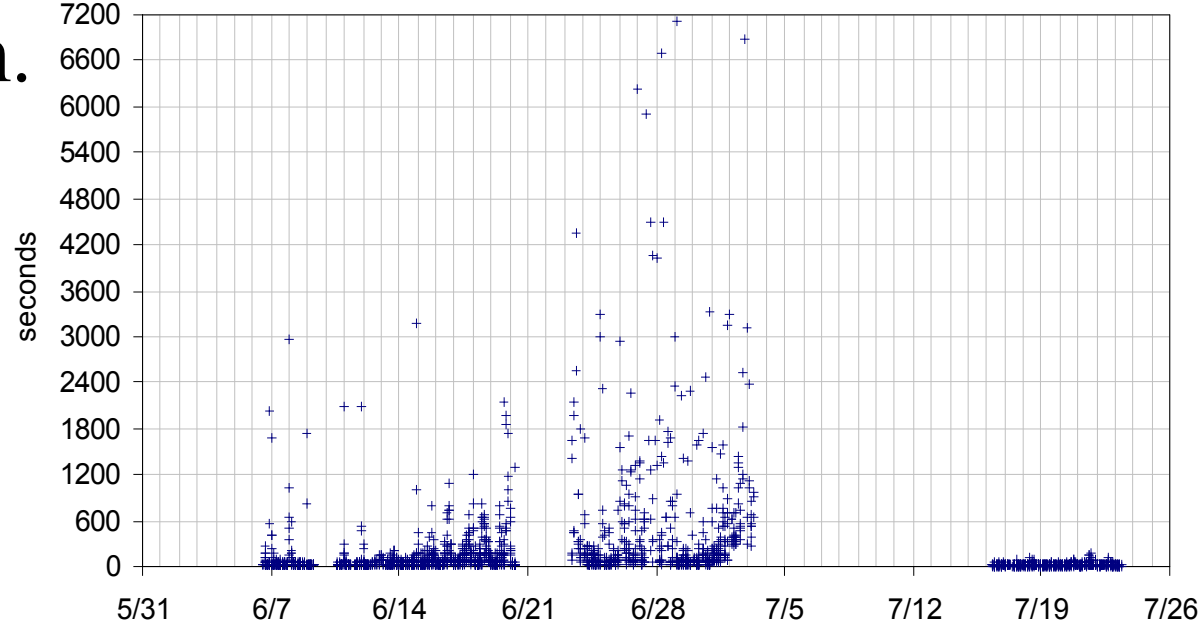
UBC is Appropriate for Half Our Workload



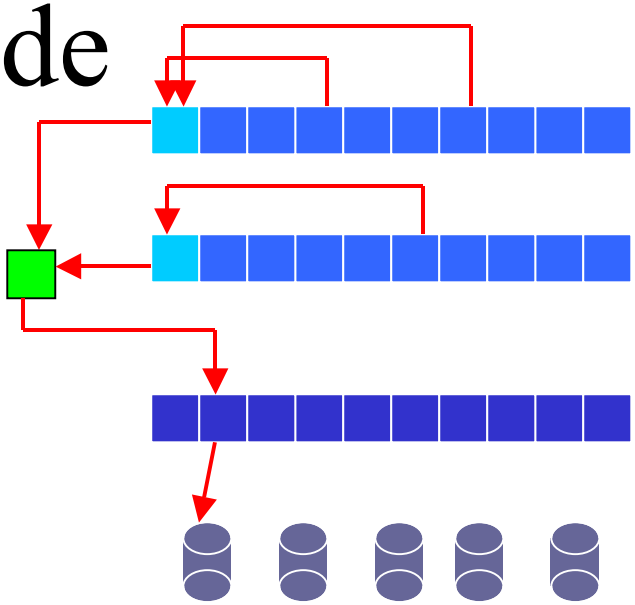
UBC upside

Robust Performance for:

- File system fragmentation,
- Mixed block sizes,
- And contention.



UBC downside

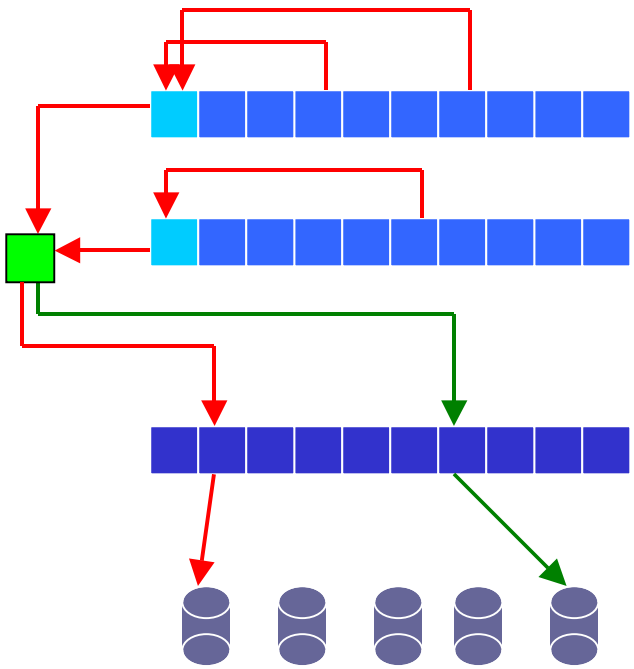


- GigE.
- Requires client side memory.
- Greater demand on I/O serving node.
- Not as efficient for large block sizes.
- Number of potential bottlenecks is larger - more difficult to analyze.

Ongoing Work

Investigate the impact of:

- File system size,
- Attribute caching,
- And rotating file systems.



Conclusion: One solution does **NOT** fit all.