TBBT-Trace Based file system Benchmarking Tool

Ningning Zhu, Jiawu Chen, Tzi-cker Chiueh
Stony Brook University
Daniel Ellard
Harvard University
Fast'04 Work In Progress

Synthetic FS benchmarks & Drawbacks

- **Synthetic Macro-benchmarks**
  - SPECsfs (NFS)
  - SDET
  - Postmark
  - SSH-Build
  - TPCC
  - Andrew Benchmark
- **Synthetic Micro-benchmarks**
- **Hybrid Benchmarks**
  - hBench

Outdated

Unrealistic

Misleading

FS traces, contributions, and trends

- 1985 Ousterhout's trace
- 1991 Sprite trace analysis
- 1999 Vogels, FS Usage in Windows NT
- 2000 Roselli *et al*, A Comparison of FS workloads
- 2003 Ellard *et al,* NFS trace study

Workload characterization to guide file system design

Larger, passive, realistic

Goal of TBBT toolkit

- Realistic
- Up to date

- Easy to use
- Scalable
- Light-weight

Challenges

Initial image

Inactive files,

Aging

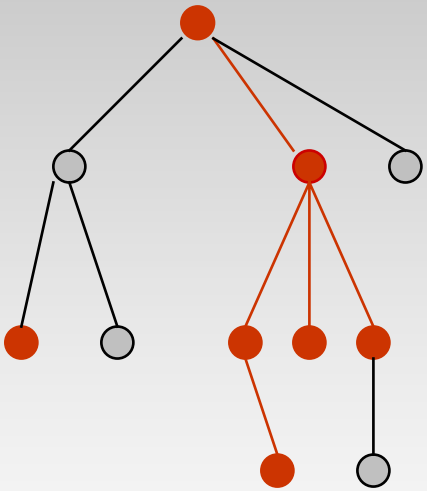
Concurrency

Error handling

Disk/CPU usage

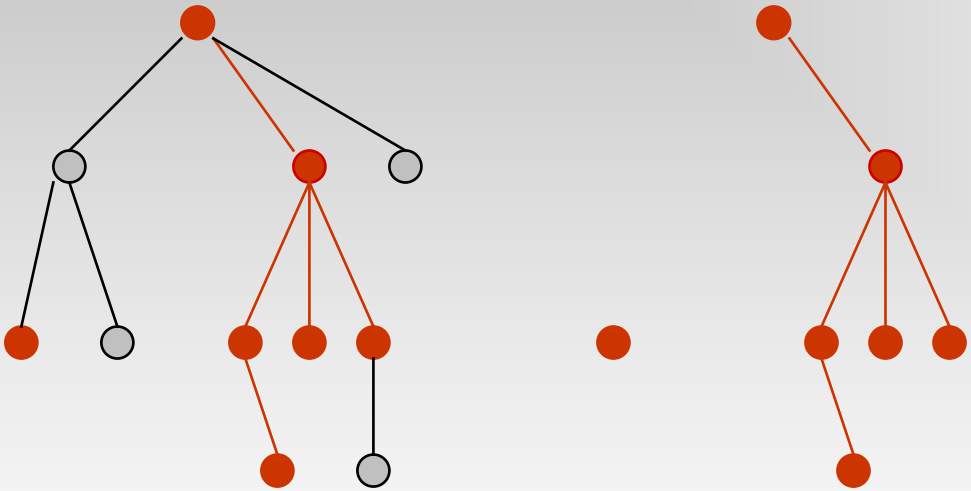
File System Initialization

- Actual image



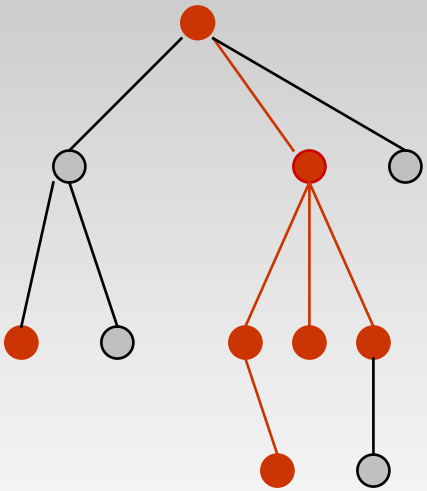
File System Initialization

- Actual image Extracted image

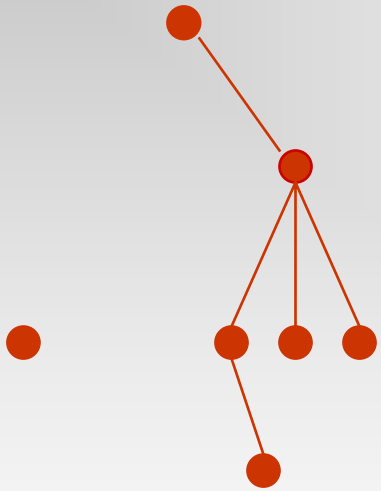


File System Initialization

- Actual image



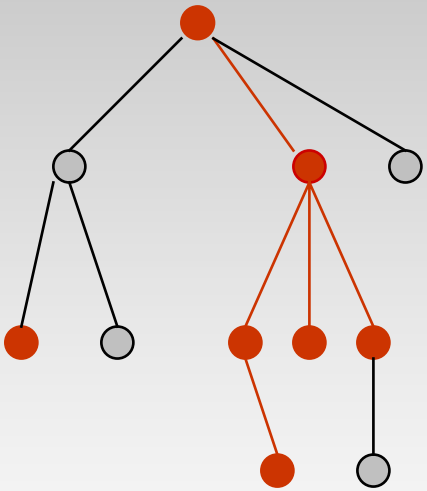
Extracted image



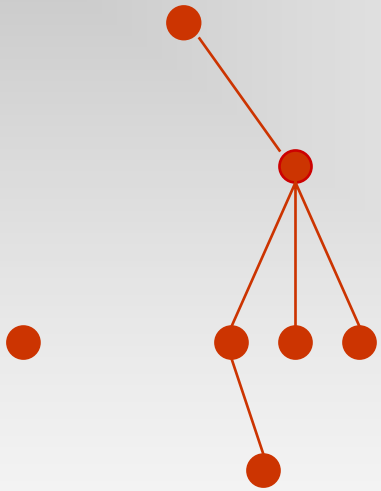
Initial image

File System Initialization

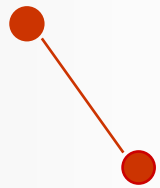
- Actual image



Extracted image

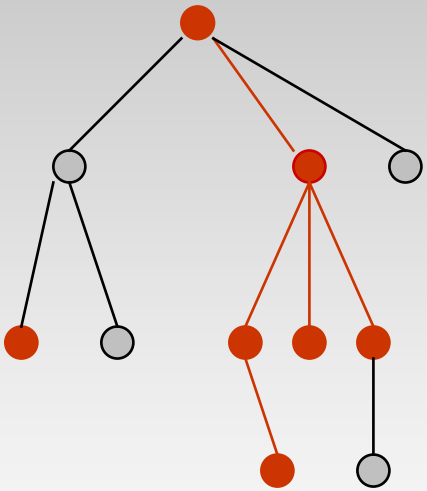


Initial image

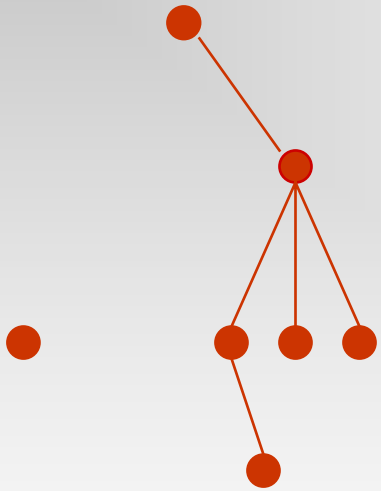


File System Initialization

- Actual image



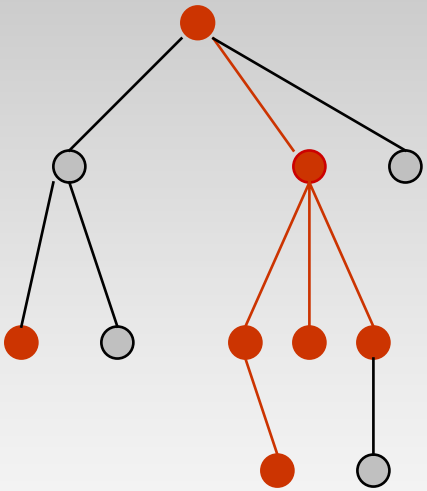
Extracted image



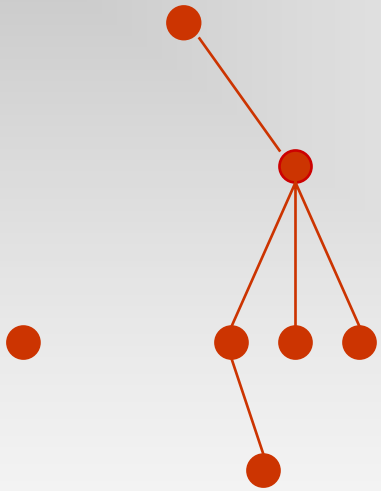
Initial image

File System Initialization

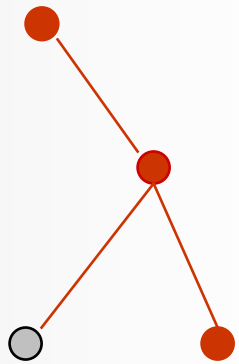
- Actual image



Extracted image

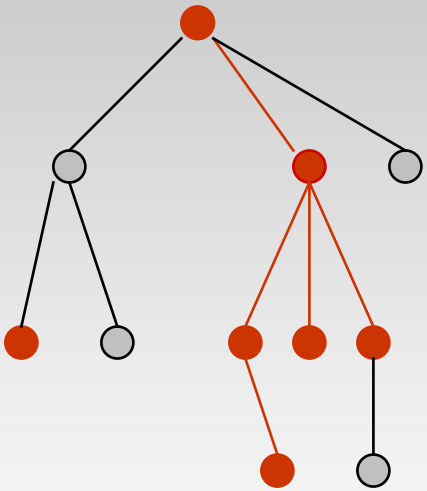


Initial image

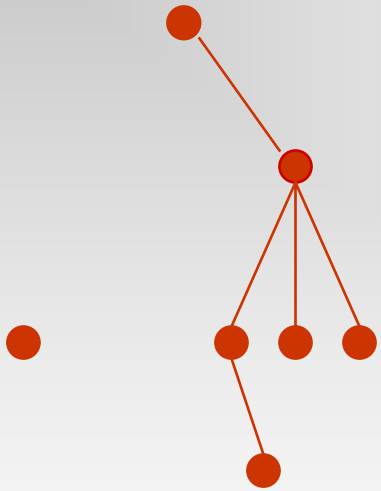


File System Initialization

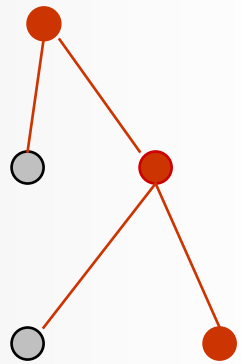
- Actual image



Extracted image

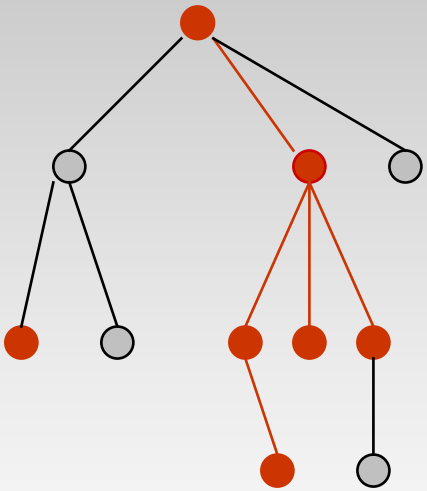


Initial image

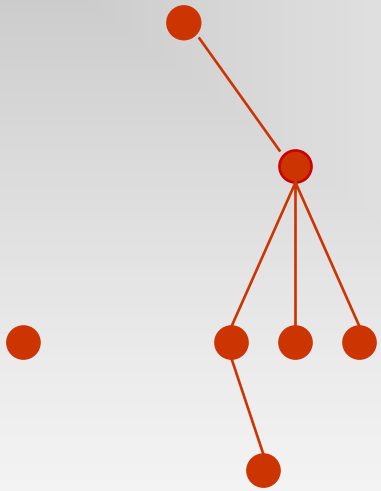


File System Initialization

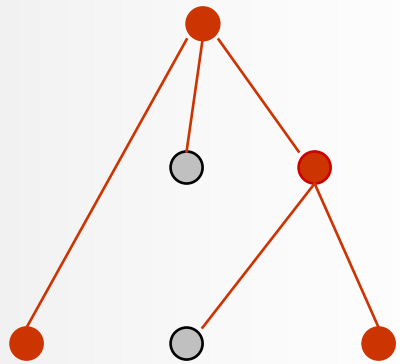
- Actual image



Extracted image

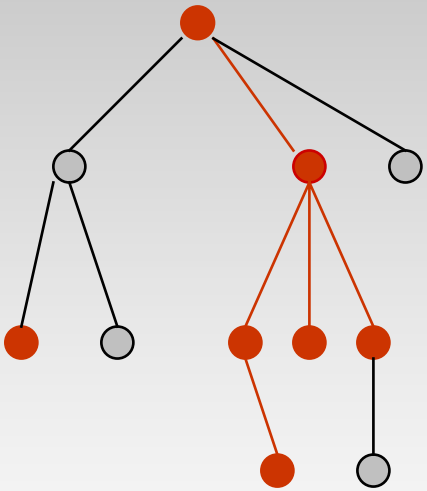


Initial image

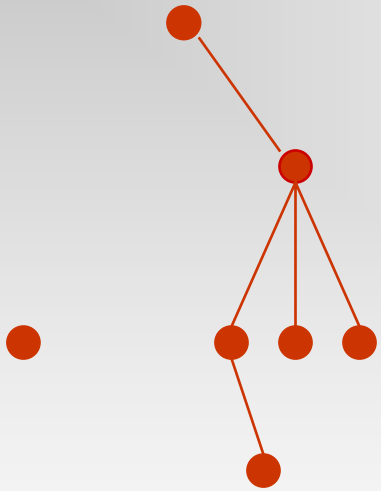


File System Initialization

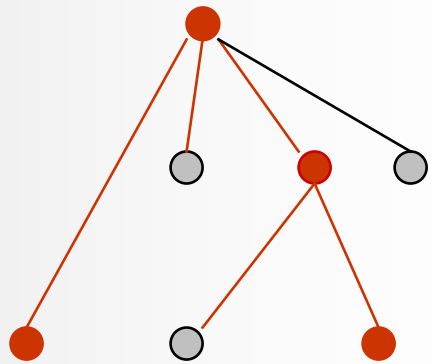
- Actual image



Extracted image

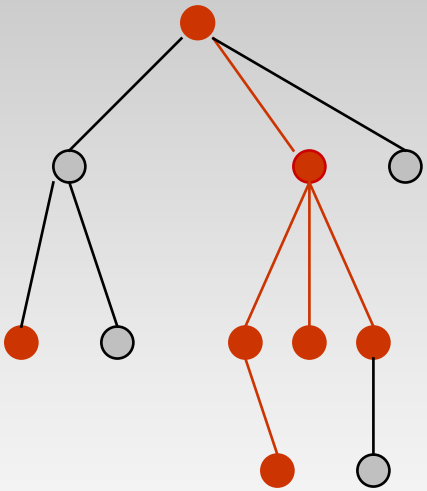


Initial image

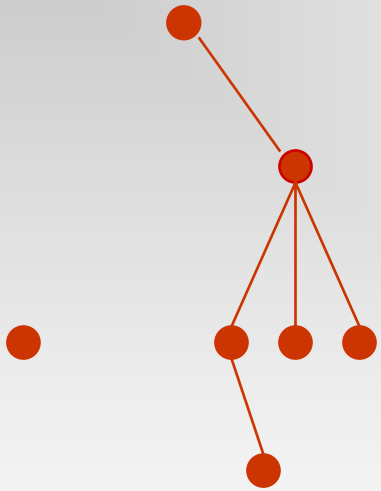


File System Initialization

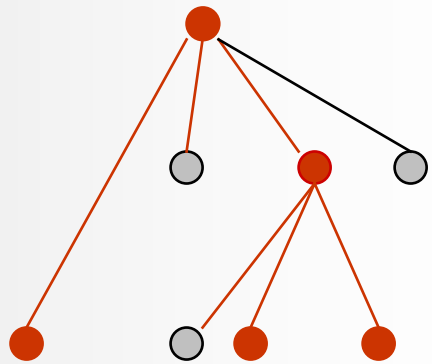
- Actual image



Extracted image

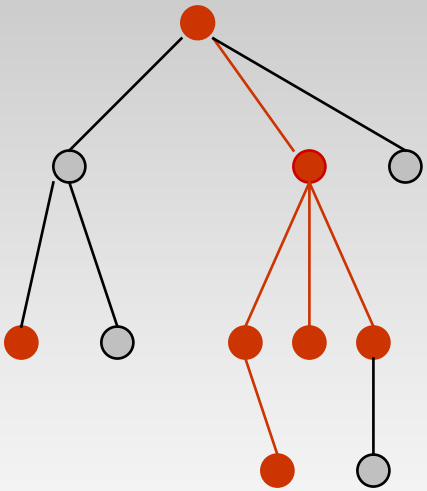


Initial image

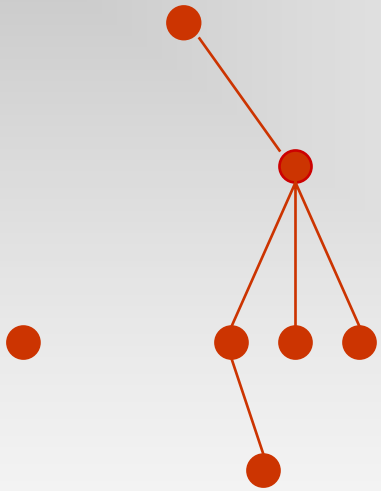


File System Initialization

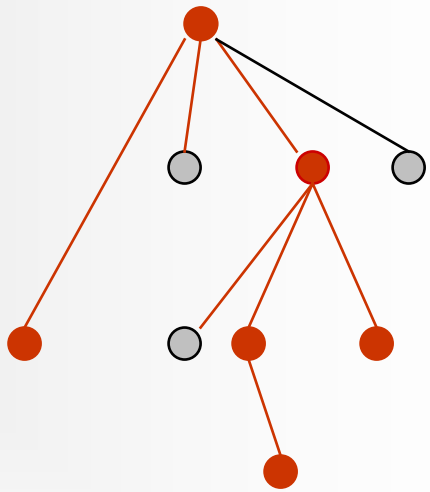
- Actual image



Extracted image

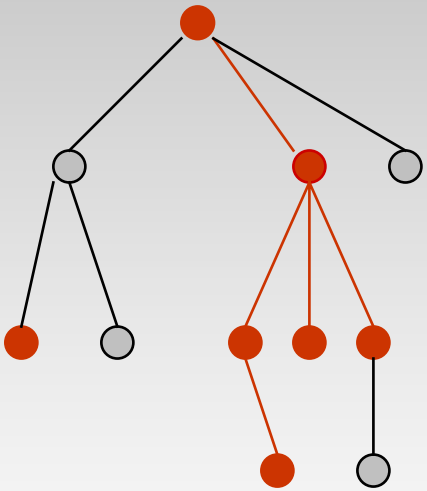


Initial image

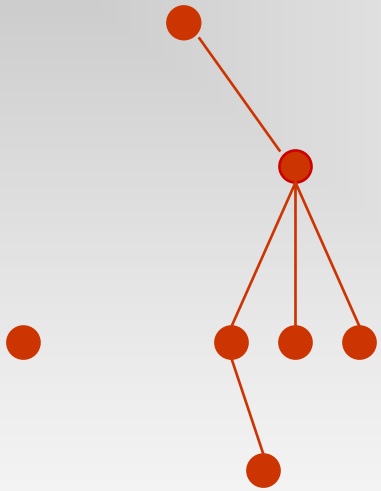


File System Initialization

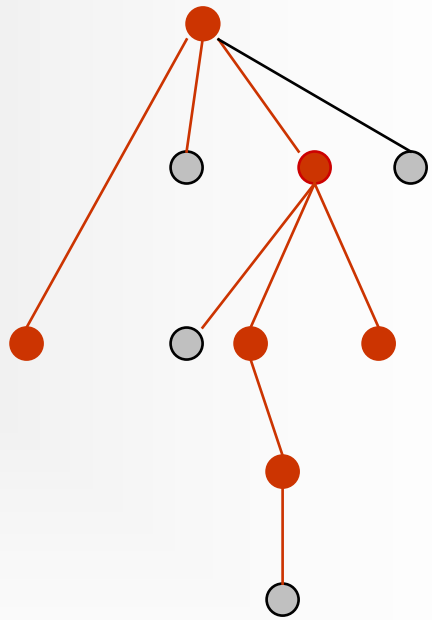
- Actual image



Extracted image

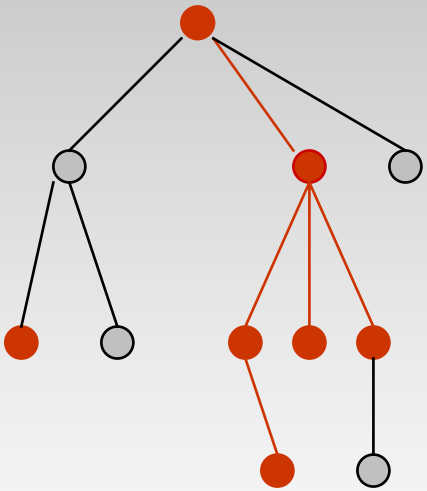


Initial image

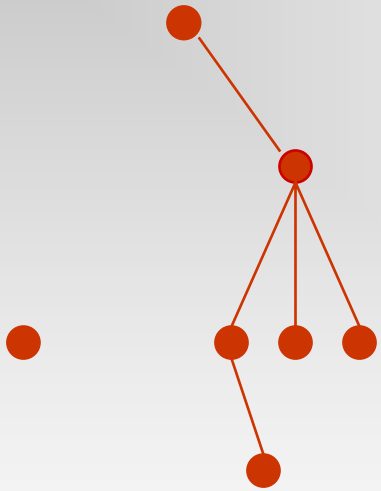


File System Initialization

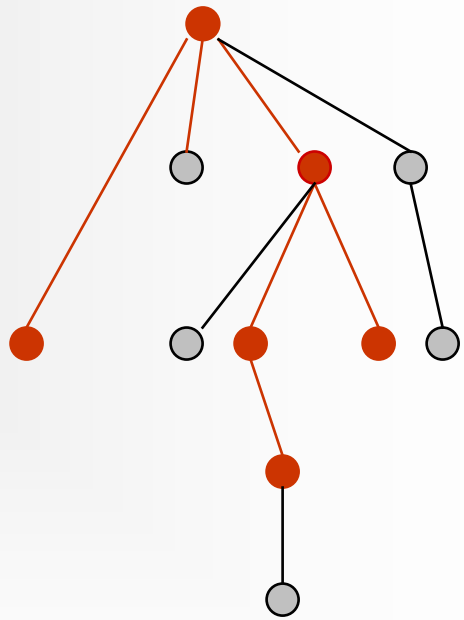
- Actual image



Extracted image

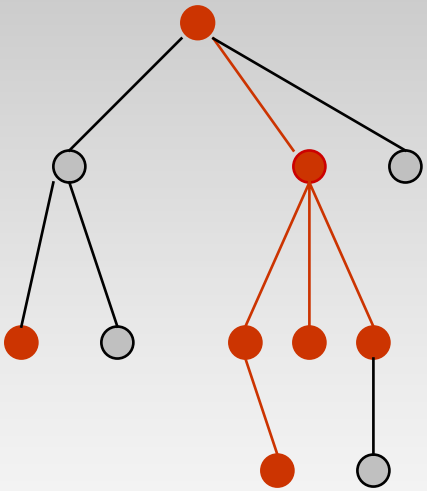


Initial image

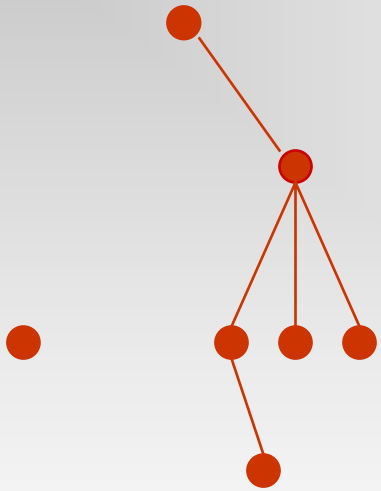


File System Initialization

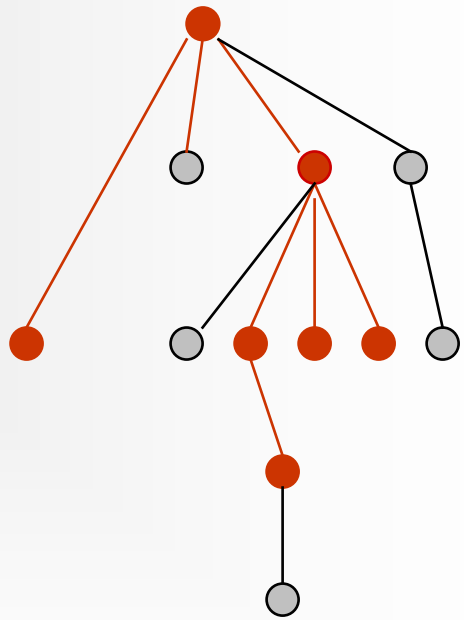
- Actual image



Extracted image

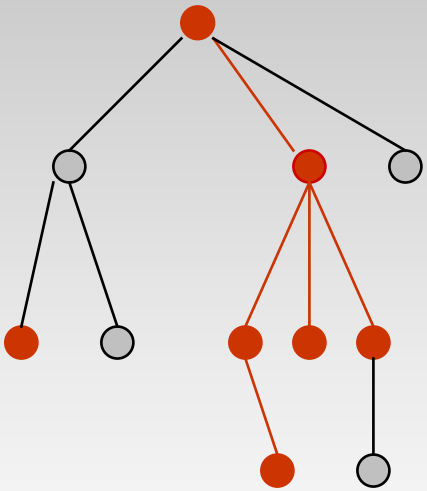


Initial image

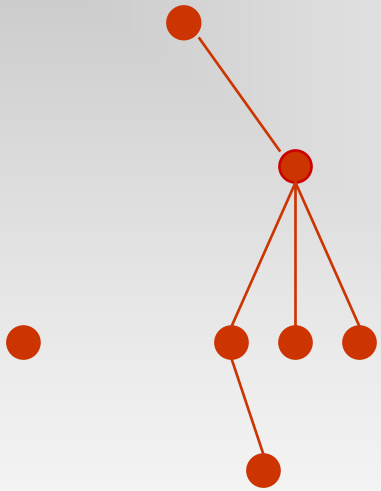


File System Initialization

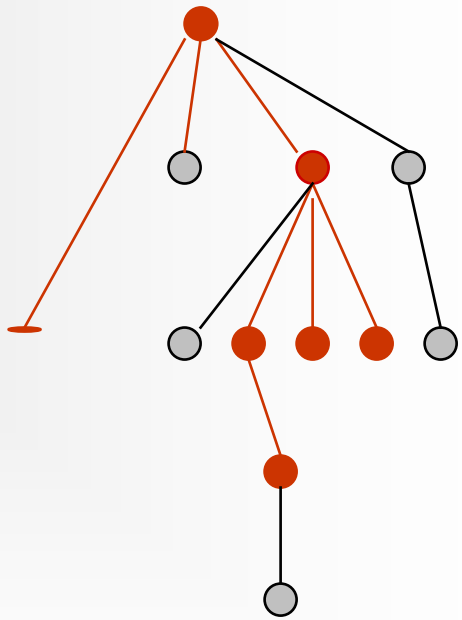
- Actual image



Extracted image

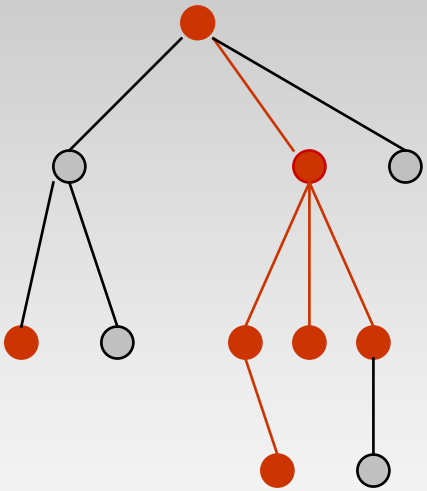


Initial image aging

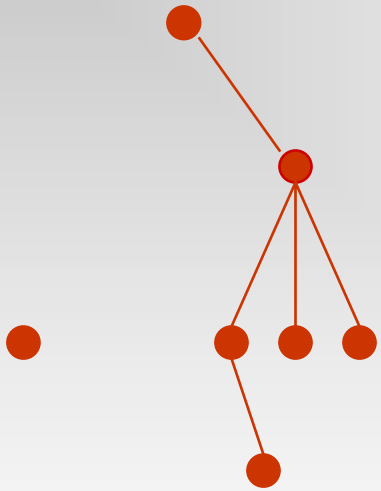


File System Initialization

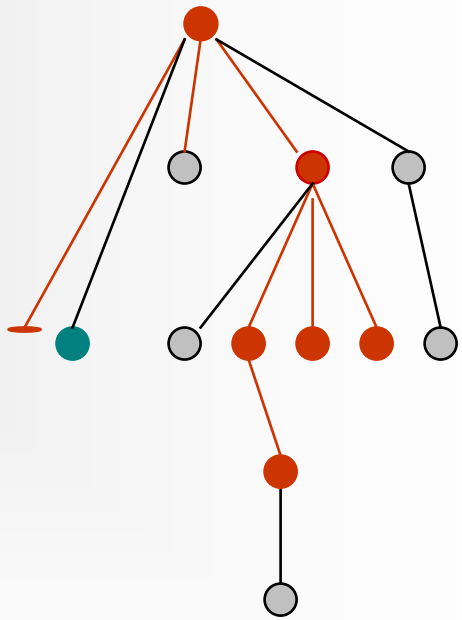
- Actual image



Extracted image

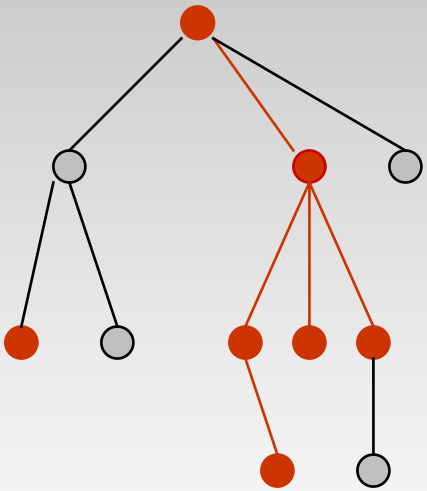


Initial image aging

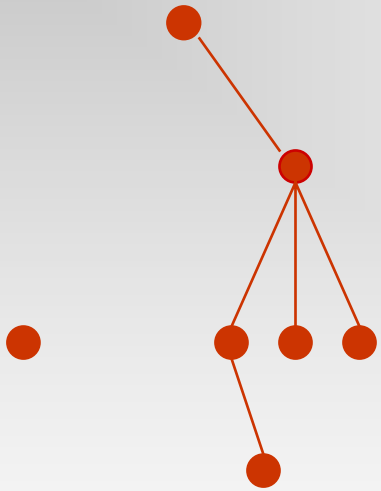


File System Initialization

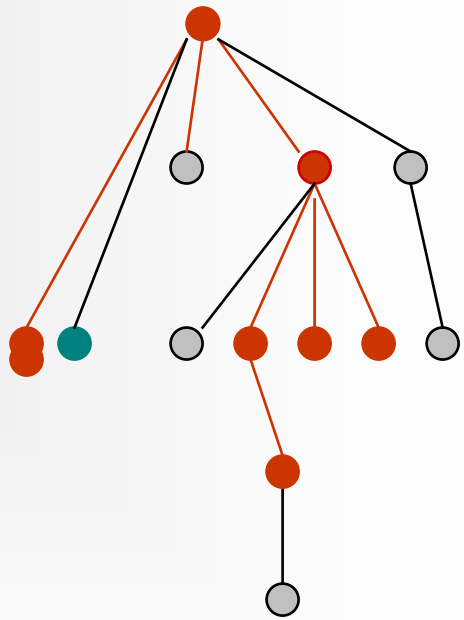
- Actual image



Extracted image

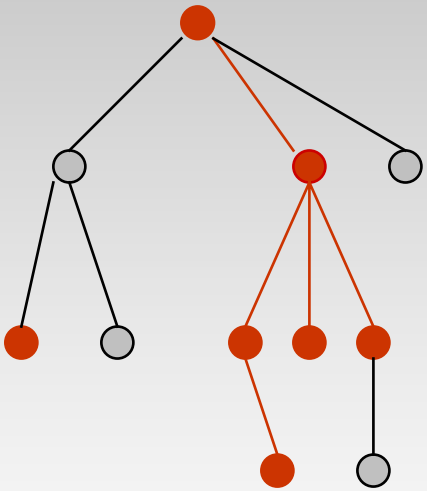


Initial image aging

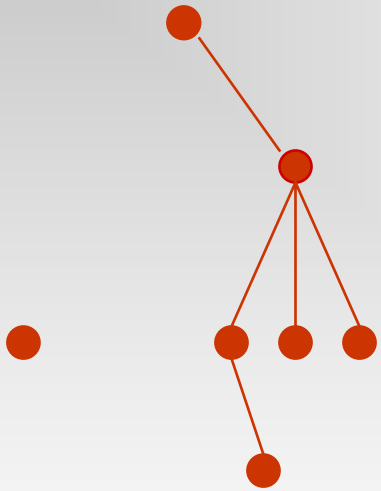


File System Initialization

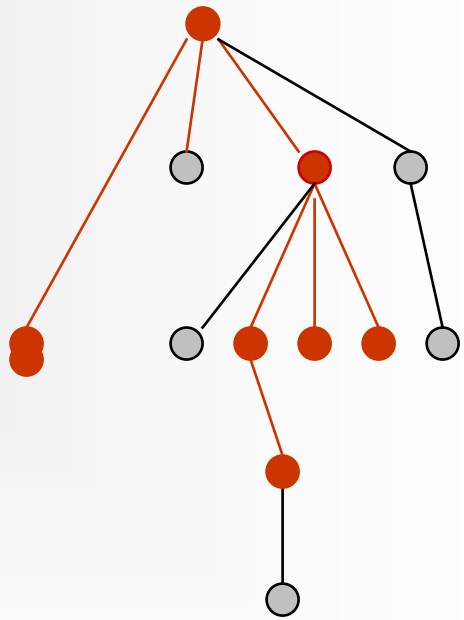
- Actual image



Extracted image



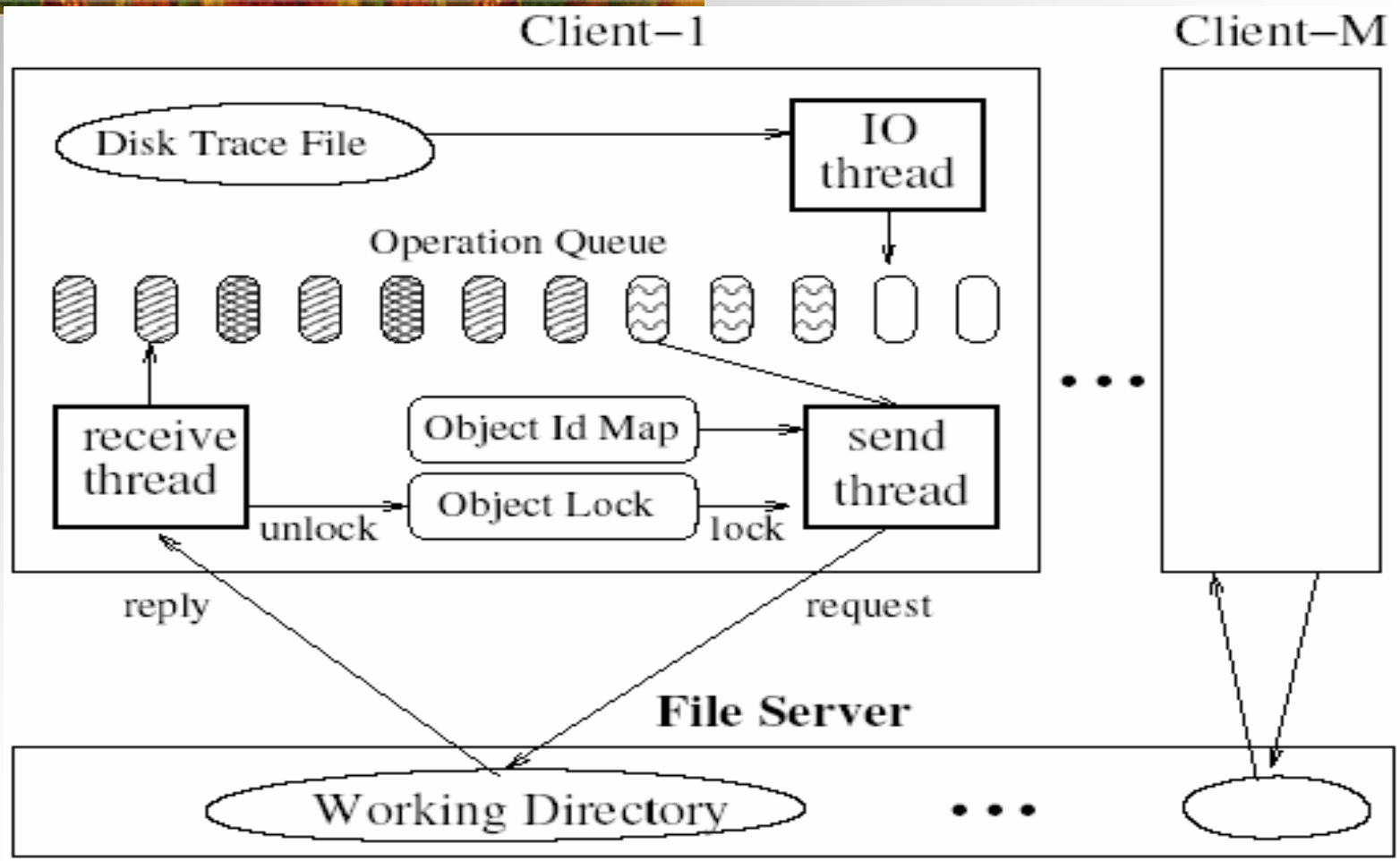
Initial image aging



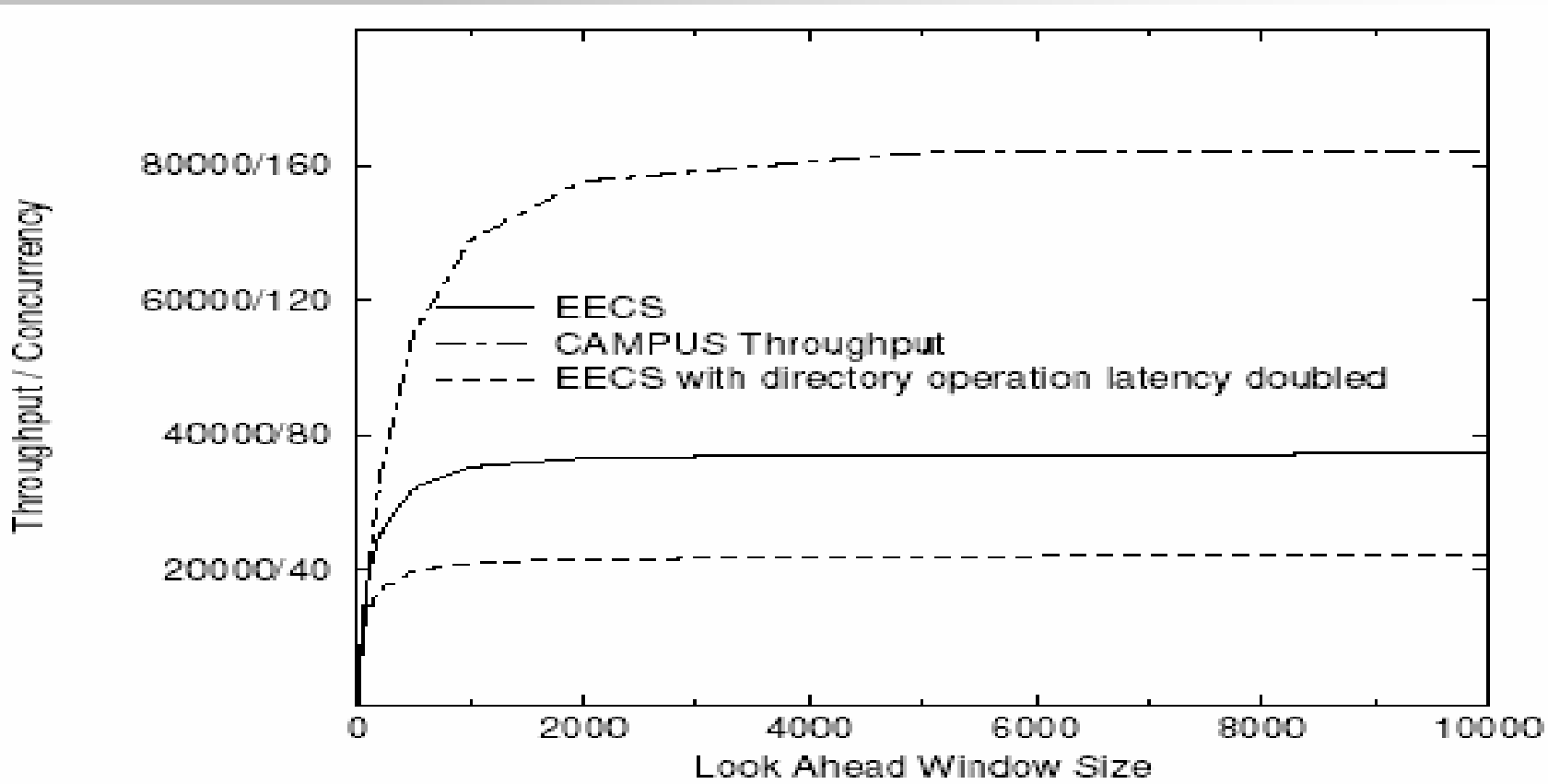
Dependency Analysis

Operation	modify	access
Read/getattr ***obj***		Obj
Write/setattr ***obj***	Obj	Obj
Lookup ***dir name([obj])***		Dir,[obj]
Create/mkdir ***dir name(obj)***	Dir,obj	Dir,obj
Remove/rmdir ***dir, name([obj])***	Dir,[obj]	Dir,[obj]

TBBT load generator



Trace Concurrency



NFS/RFS evaluation by TBBT and SPECsfs

	original load		scale-up		peak load	
Operation	**SPEC**	**TBBT**	**SPEC**	**TBBT**	**SPEC**	**TBBT**
Throughput	33	30	189	180	1231	1807
getattr	5.1	0.6	0.9	1.5	2.1	0.7
lookup	2.9	0.9	0.8	2.0	2.0	1.2
read	9.6	3.1	5.3	4.8	5.4	4.7
write	9.7	2.2	4.4	3.8	4.6	2.5
create	0.5	0.7	0.7	0.9	17.3	0.7

Table 2: NFS latency/throughput for EECS trace at Oct 21, 2001.

	original load		scale-up		peak load	
Operation	**SPEC**	**TBBT**	**SPEC**	**TBBT**	**SPEC**	**TBBT**
Throughput	32	30	187	180	619	1395
getattr	4.0	0.7	2.2	1.2	3.2	0.8
lookup	4.4	0.7	2.8	1.3	2.6	1.0
read	10.8	3.3	8.4	4.1	18.1	4.9
write	11.6	5.4	7.4	4.0	11.1	2.8
create	0.7	1.0	5.1	1.3	16.3	1.2

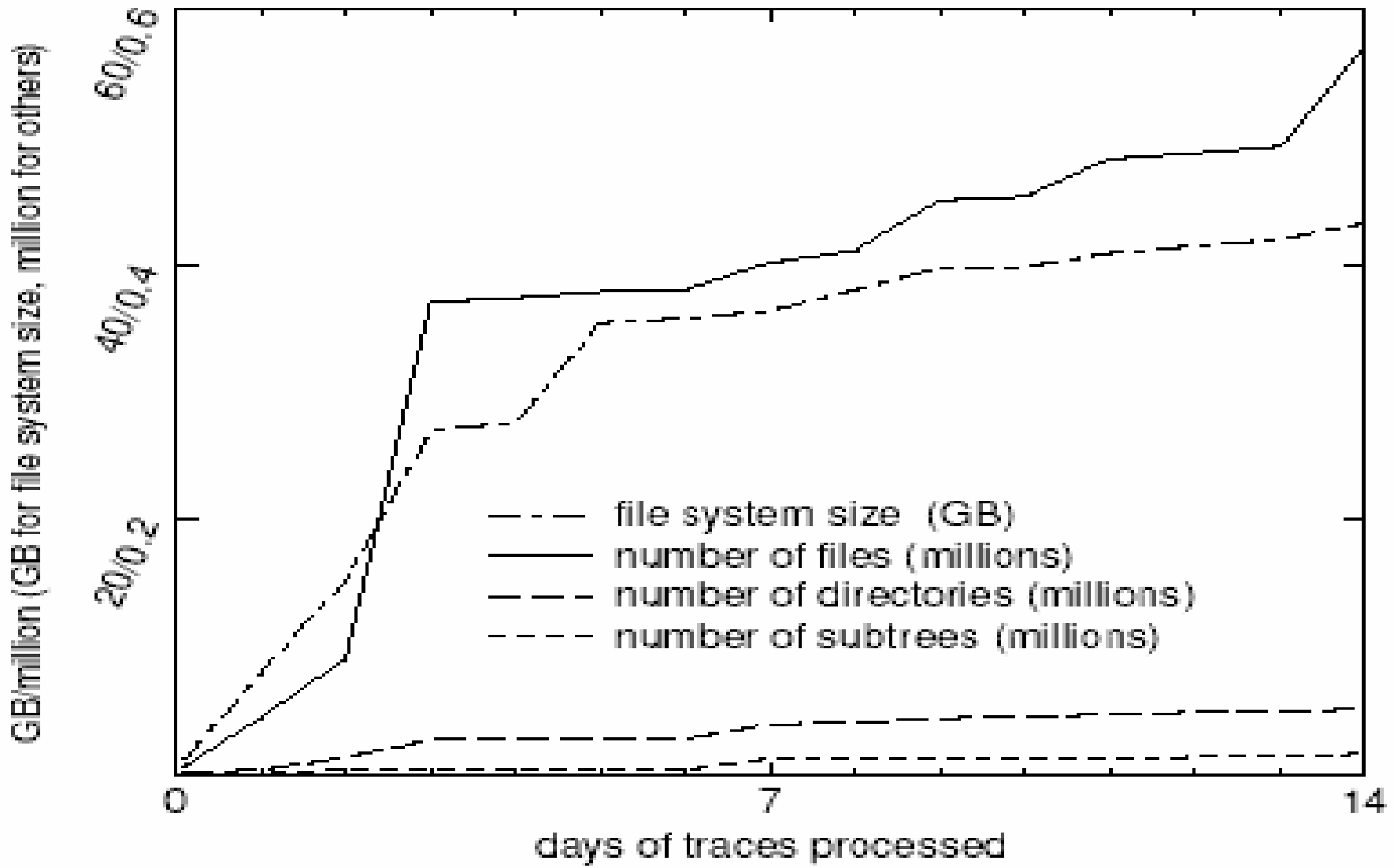
Table 3: RFS latency/throughput for EECS trace at Oct 21, 2001.

Conclusion

- TBBT is a scalable, flexible, and efficient toolkit for file system evaluation.
- The trace-driven nature makes it capable of capturing the diverse workload features and their fast evolvement.

Thanks! Questions?

File System Hierarchy Discovery



Synthetic Workload Generator