Integrating Portable and Distributed Storage

Niraj Tolia

Jan Harkes, Michael Kozuch, M. Satyanarayanan

PARALLEL DATA LABORATORY
INTEL RESEARCH PITTSBURGH
Carnegie Mellon University

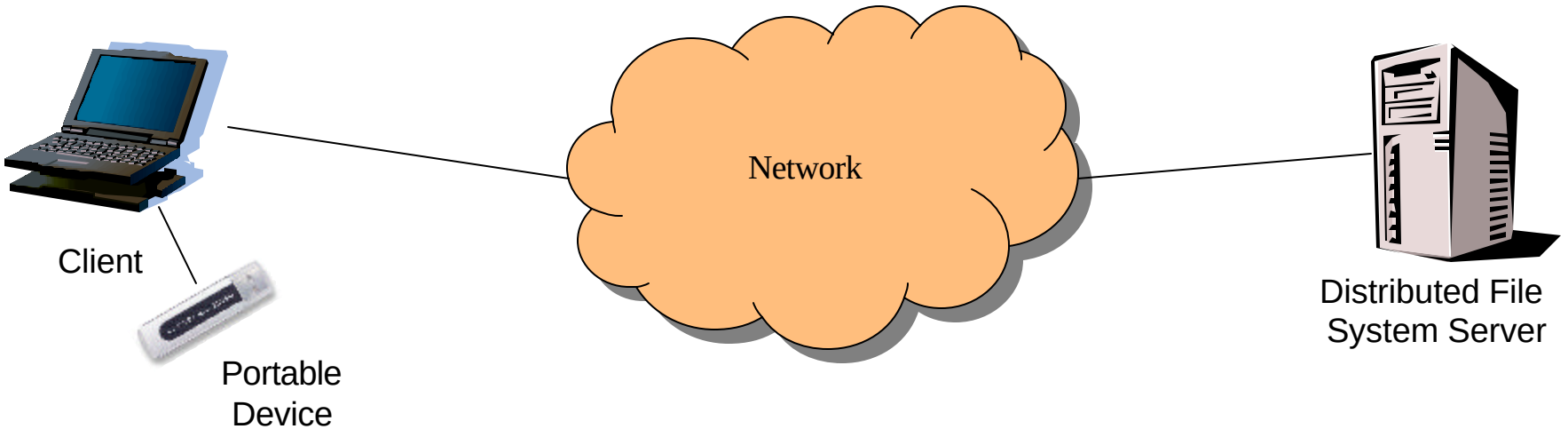
Motivation

- Explosive growth of portable storage devices
- Are these glorified floppy disks?
- In a world of ever-improving networking, why do we still carry them around?

Integration Value

- Portable Devices
  - Performance
  - Availability
  - Ubiquity
- Distributed File Systems
  - Robustness
  - Sharing and Collaboration
  - Consistency
  - Capacity

10,000 Foot View



- *Lookaside Caching* - All cache lookups check portable device
- Minimal disruption of existing usage model

Outline

- Motivation
- Lookaside Caching
- Benchmarks and Evaluation
- Related Work
- Conclusion

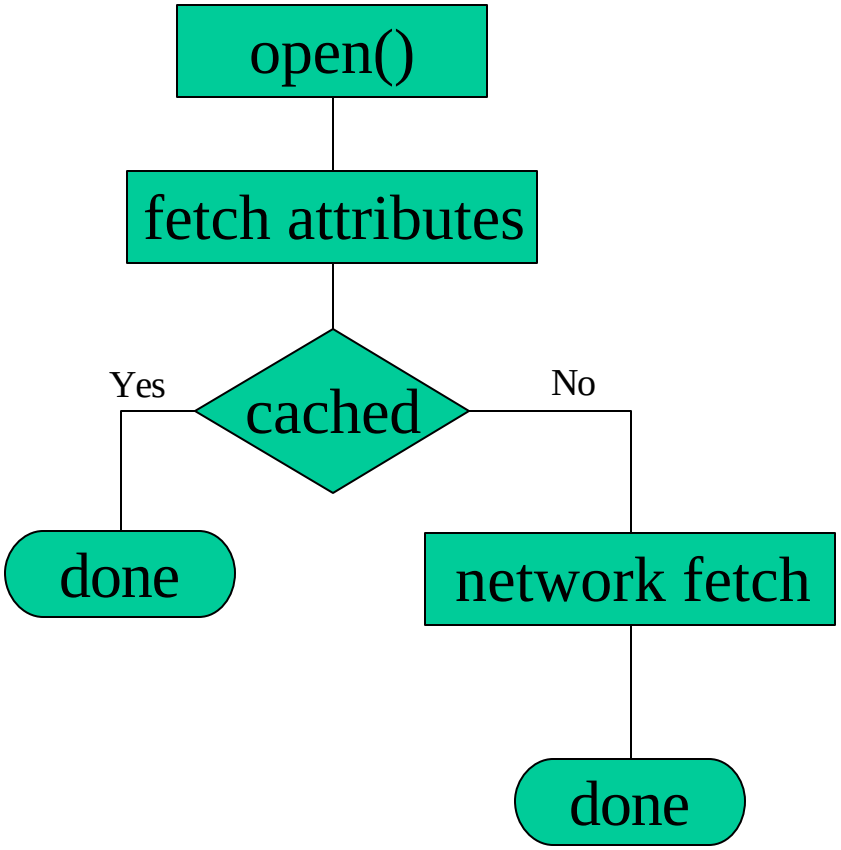
Lookaside Caching

- Technique to combine the strengths of dist. file systems and portable storage
- Design Goals
  - Be tolerant of user error
  - No compromise of robustness or consistency
- *Simplicity!*

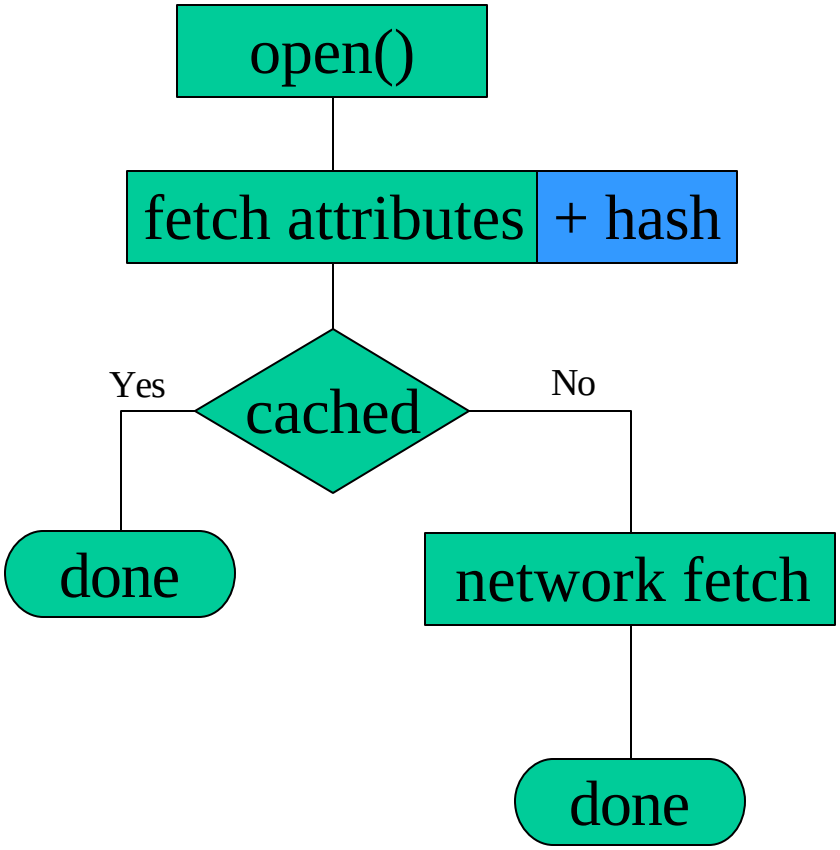
Design Decisions

- File server is the authoritative source
- No change made to native file system
  - Allows flexibility in the choice of device
  - Allows user to look at files stored on the portable device without a file system client
  - Only addition is an index file

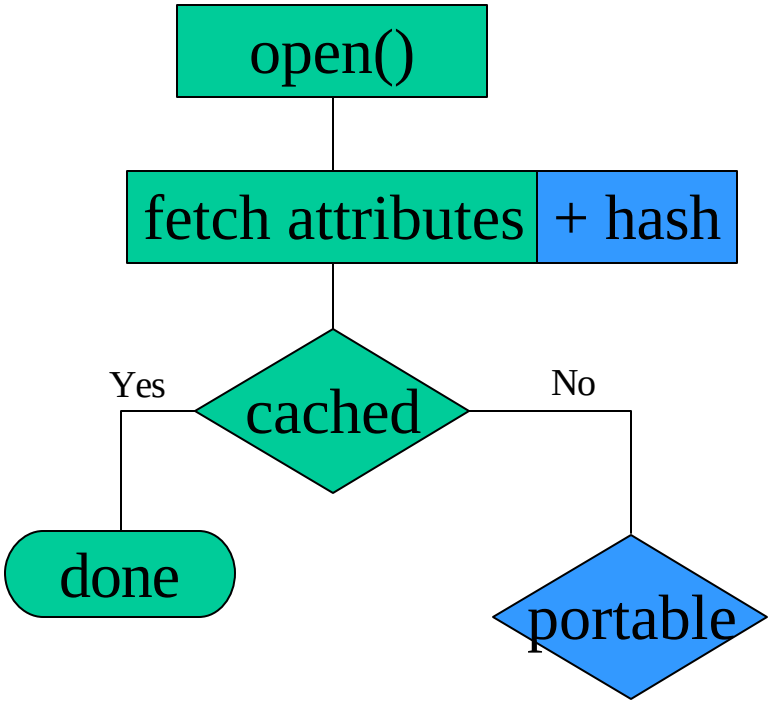
Lookaside Caching Flowchart



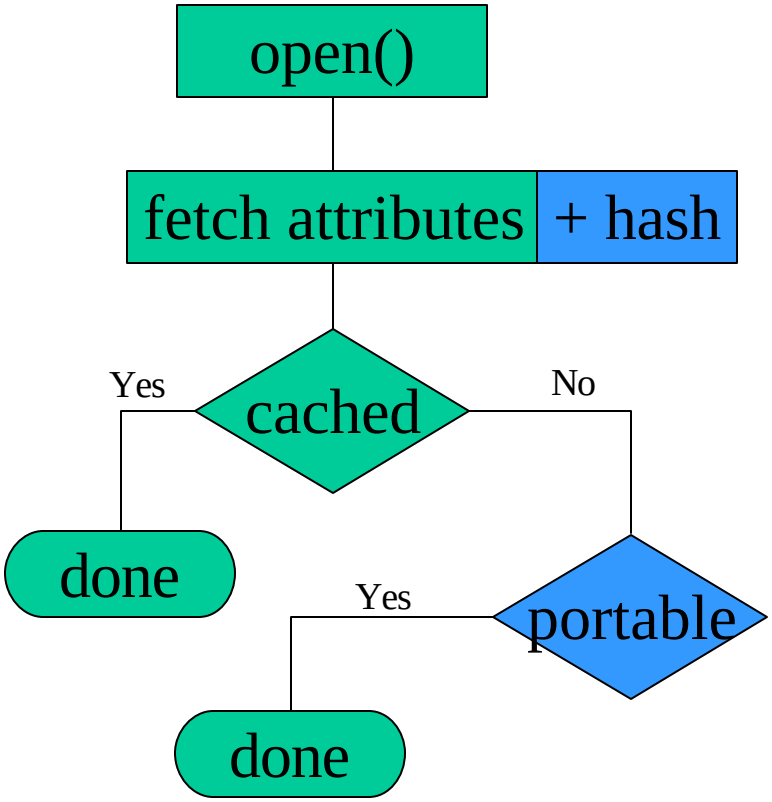
Lookaside Caching Flowchart



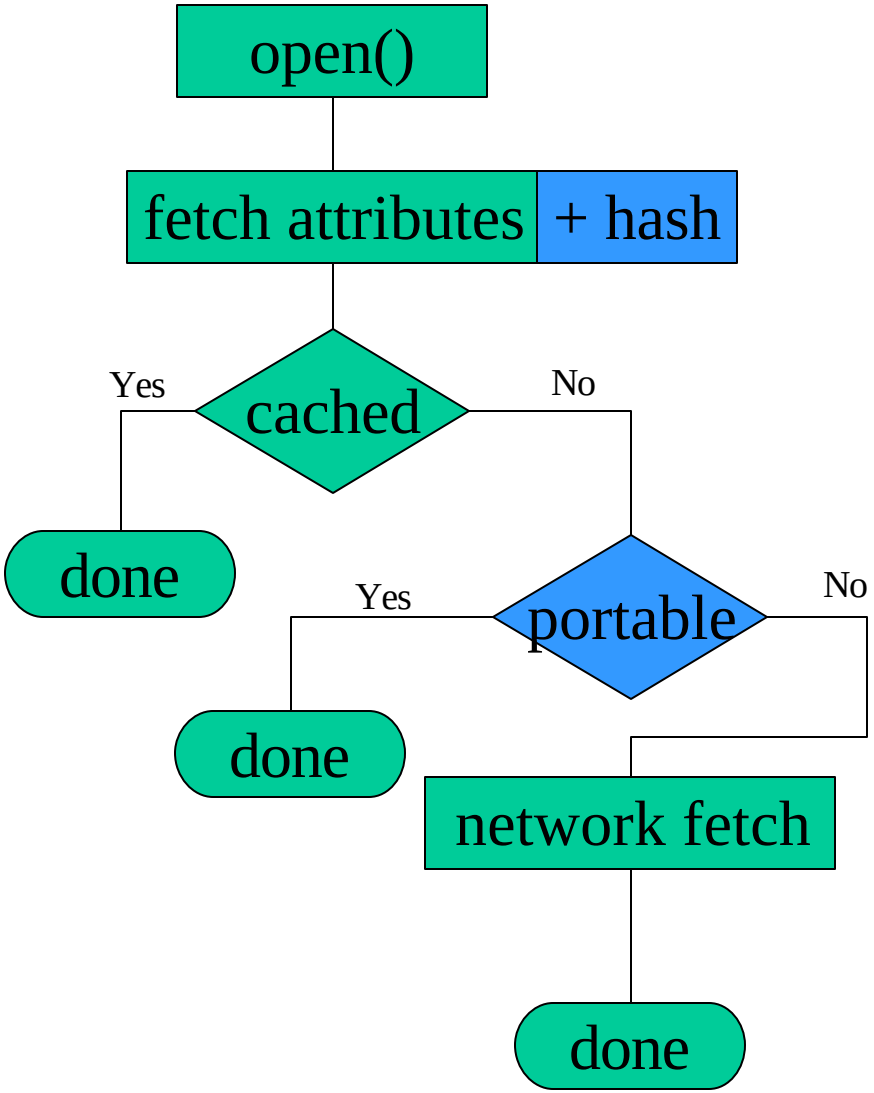
Lookaside Caching Flowchart



Lookaside Caching Flowchart



Lookaside Caching Flowchart



Implementation

- File system metadata extended to include a cryptographic hash (SHA-1)
- Based upon whole-file hashing
- Implemented in Coda 6.0

- *mkdb* utility to generate index file
  - Index generated from normal file system tree
  - Lazy update process
  - Allows users to change data

Outline

- Motivation
- Lookaside Caching
- Benchmarks and Evaluation
  - Benchmark Descriptions
  - Evaluation Setup
  - Results
- Related Work
- Conclusion

Benchmarks

- Kernel Compile
- Client File System Trace
- Virtual Machine Migration

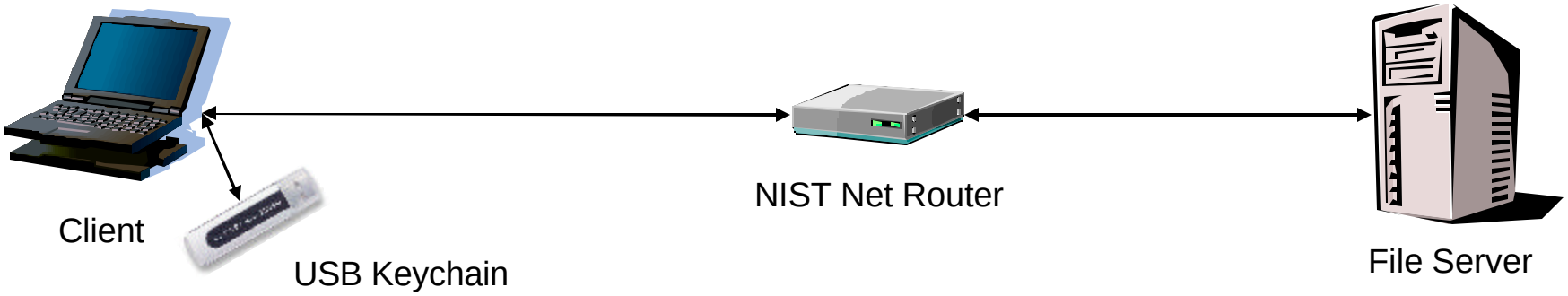
Benchmark Description

- Kernel Compile
  - Compiled Linux 2.4.18
  - Compile reads 80.7 MB (118 MB source tree)
  - Portable device contained either 2.4.18, 2.4.17, 2.4.13, 2.4.9, or 2.4.0
- Client File System Traces
  - Four desktop clients from the DFSTrace traces
  - No think time modelled

Benchmark Description

- Virtual Machine Migration
  - Based on the Internet Suspend/Resume project
  - Time taken to resume a migrated Virtual Machine and execute a MS Office-based productivity benchmark

Evaluation Setup



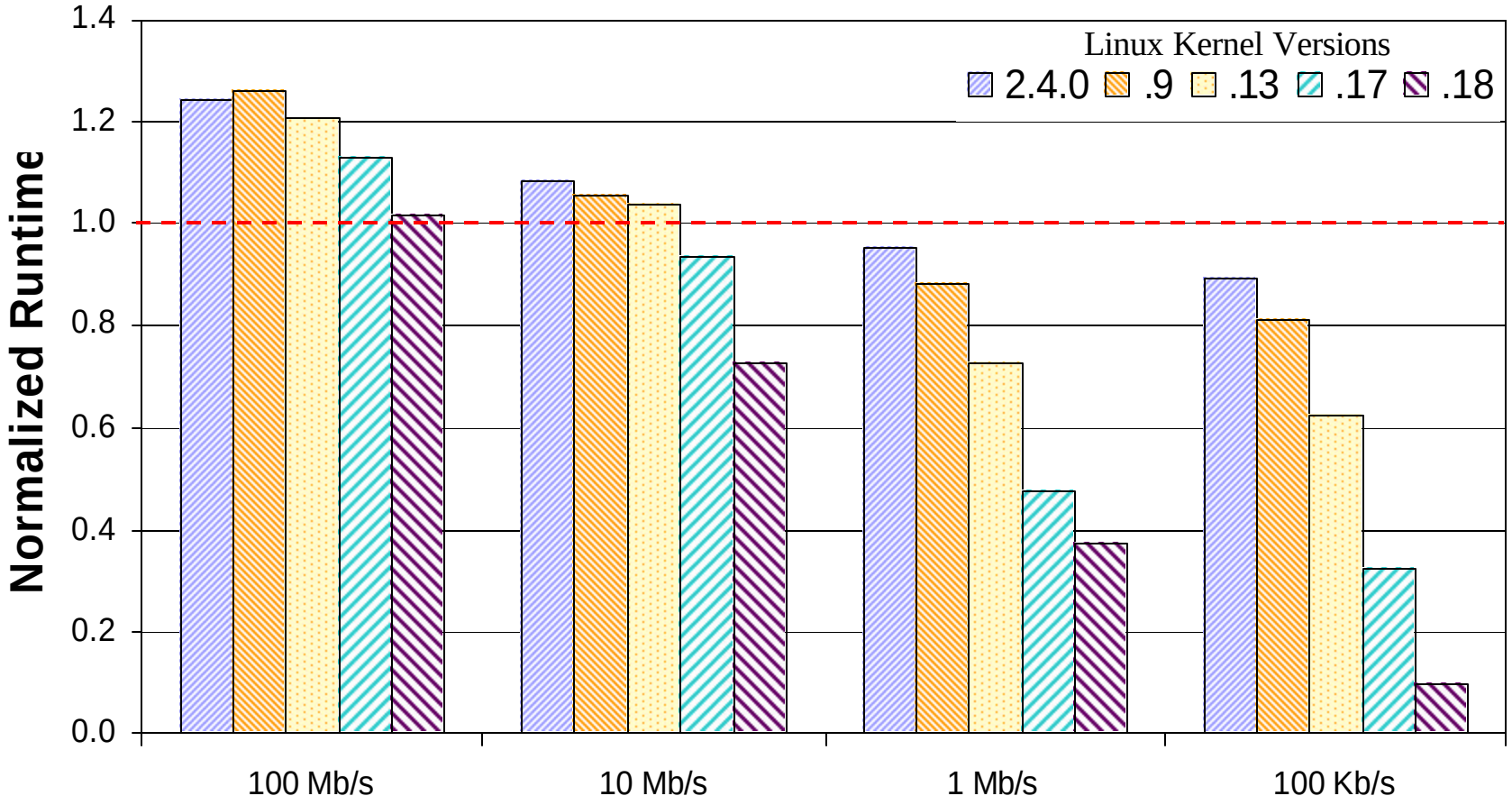
- Hi-Speed USB Flash device
- Evaluated at different bandwidths
  - 100 Mb/s, 10 Mb/s, 1 Mb/s + 10 ms, and 100 Kb/s + 100 ms

USB Portable Device Bandwidth

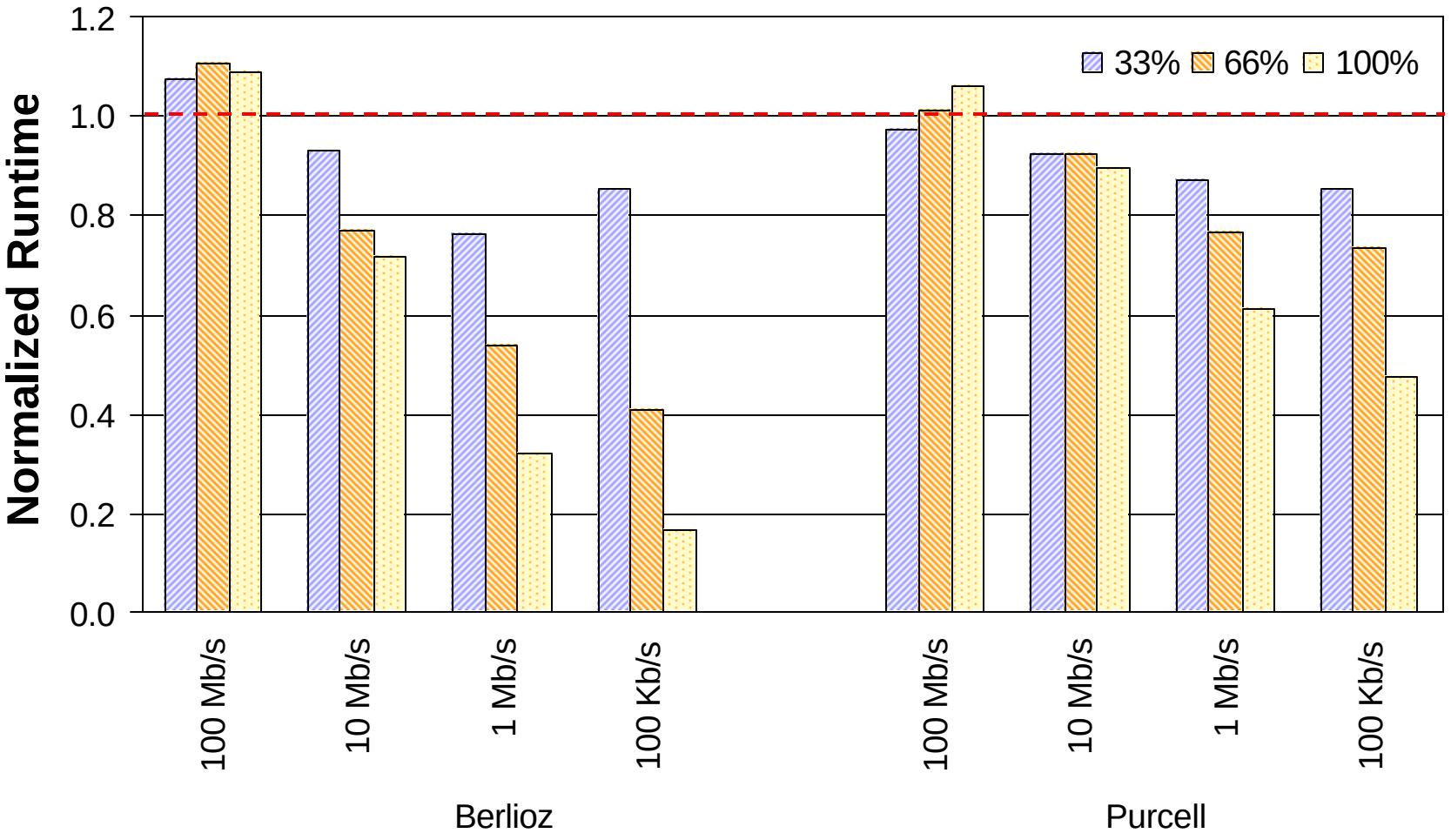
- Max below manufacturer ratings
- Low bandwidth observed for small files

Size (KB)	Read (Mb/s)	Write (Mb/s)
4 KB	6.3	7.4
64 KB	16.7	25.0
1 MB	28.6	25.8
100 MB	29.4	26.5

Linux Kernel Compile

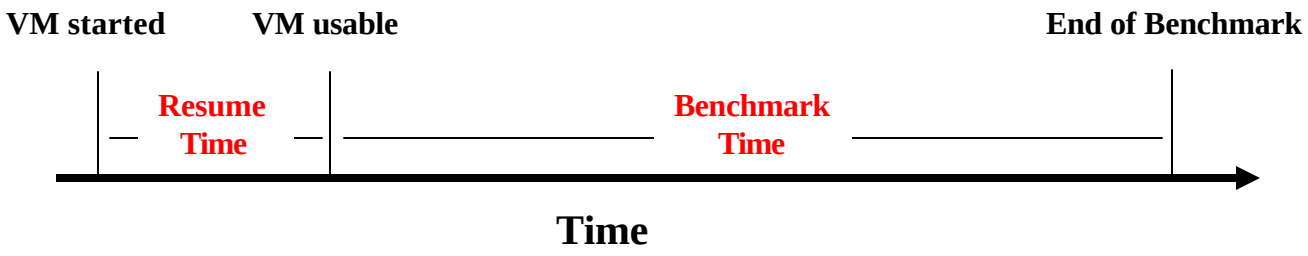


Trace Replay



Internet Suspend/Resume

- Two metrics of performance
  - Resume Time
  - Benchmark Time



Internet Suspend/Resume

- Resume Time
  - Portable device had VM memory image (100% hit rate)

	No Lookaside (s)	With Lookaside (s)	Win %
100 Mb/s	14	13	7.1%
10 Mb/s	39	12	69.2%
1 Mb/s	317	12	96.2%
100 Kb/s	4301	12	99.7%

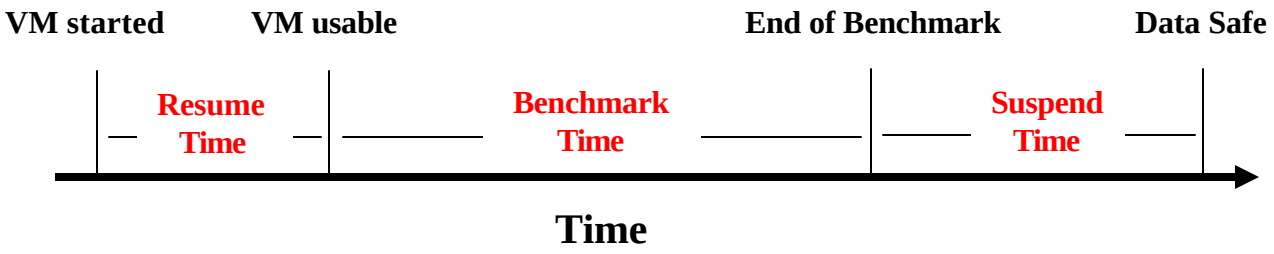
Internet Suspend/Resume

- Benchmark Time
  - Portable device had an unmodified Windows XP + Office image (~50% hit rate)

	No Lookaside (s)	With Lookaside (s)	Win %
100 Mb/s	173	161	6.9%
10 Mb/s	370	212	42.7%
1 Mb/s	2688	1032	61.6%
100 Kb/s	30531	9530	68.8%

- Slowdown

Internet Suspend/Resume



- Suspend Time
  - Lookaside caching does not currently handle this
  - Apart from portable storage, you can also use staging servers, etc.
  - Topic of ongoing research

Related Work

- Mobile Storage Solutions
  - Segank, PersonalRaid, Footloose, Personal Server
- Commercial Offerings
  - Migo, KeyComputing
- Other Content Addressable Systems
  - CASPER, LBFS
  - CFS/Chord, PAST/Pastry, Venti, Pastiche

Conclusion

- Demonstrated the integration of portable and distributed storage
- Shown that a simple optimization can have a dramatic effect on performance
- Future Work
  - While invented for portable devices, lookaside caching can be extended to use off-machine storage