Capture, conversion, and analysis of an intense NFS workload

Eric Anderson
HP Labs

aka Industrial strength NFS tracing

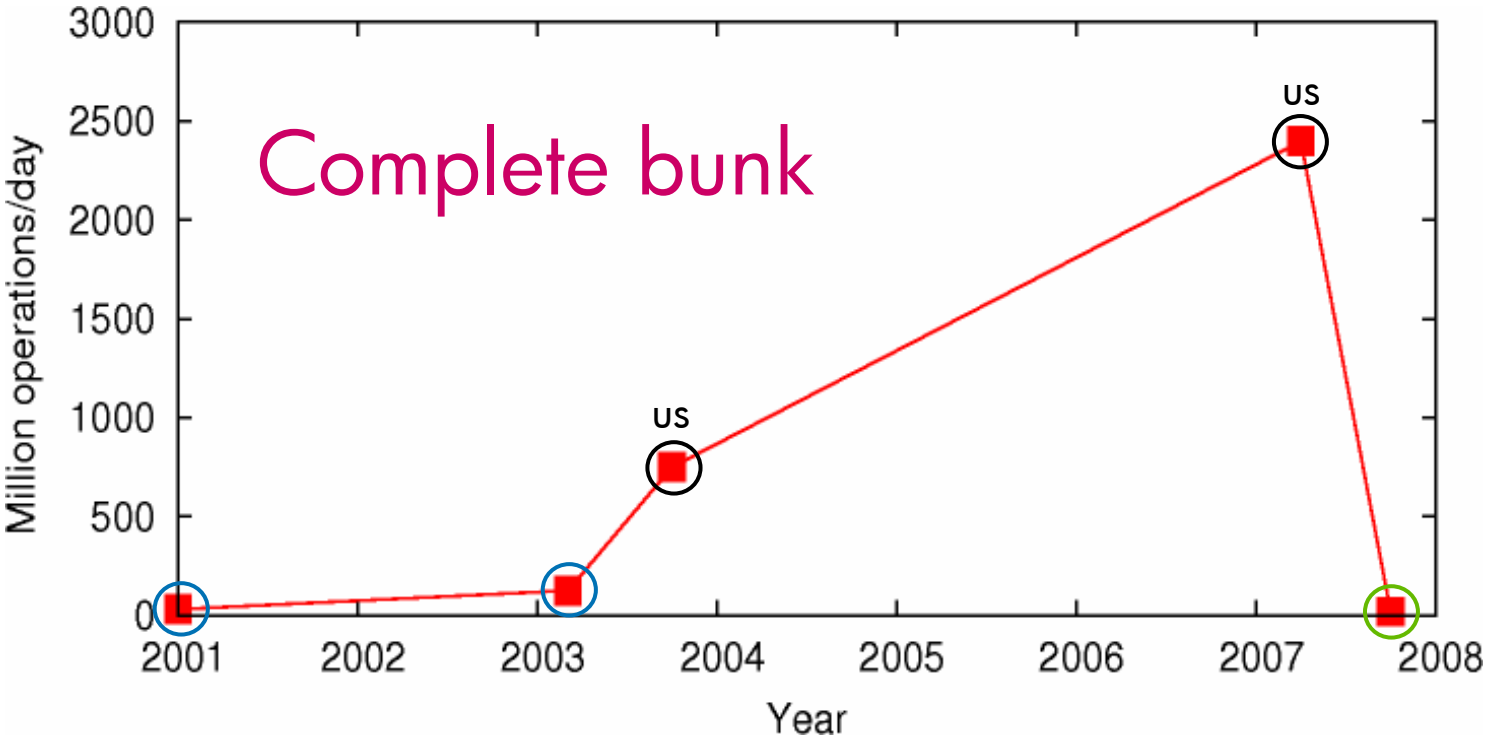
Industrial strength NFS tracing

- Wanted to collect customer NFS traces
- Applying existing techniques failed
- Going to explain how we did it
  - Many incremental improvements
  - Need most of them
  - Details in paper
- Summary:
  - If you take traces, re-read the paper, apply the lessons
  - Our workload is quite different from previous ones

Why do we take traces?

- Understand "real" workloads
  - How many operations occur?
  - How big are the files?
  - How cacheable are they?
  - How sequential are the accesses?
  - What trends are present?
- Evaluate new systems
  - Figure out new possible designs
  - Estimate performance on "real" workloads

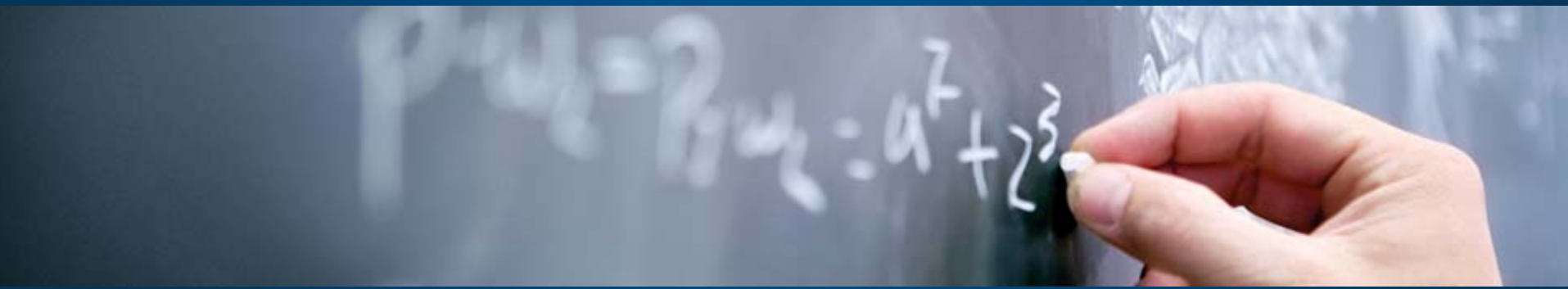
Why new traces?



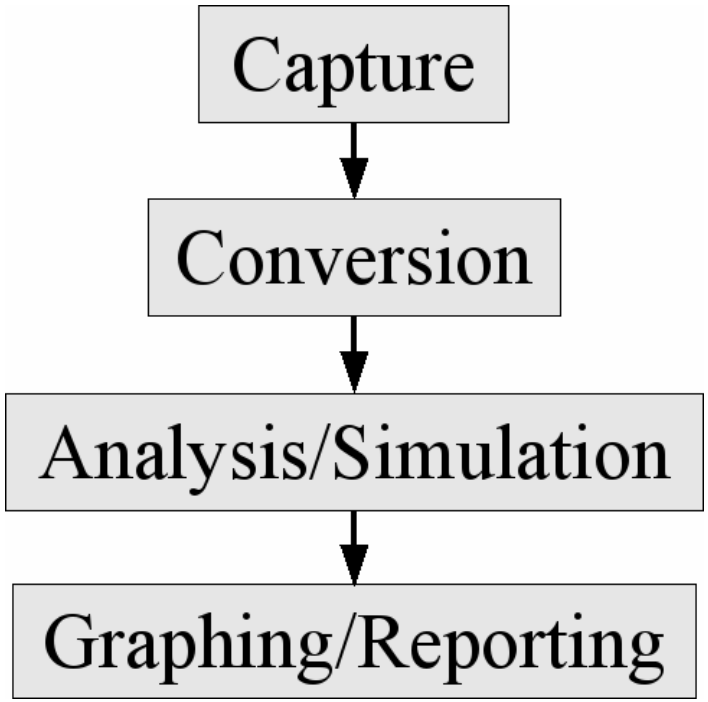
Existing tools insufficient → Develop new ones

Workloads highly variable → Collect many more traces

Improved tools



Overall trace analysis process



environment → raw form

raw → cooked

cooked → data

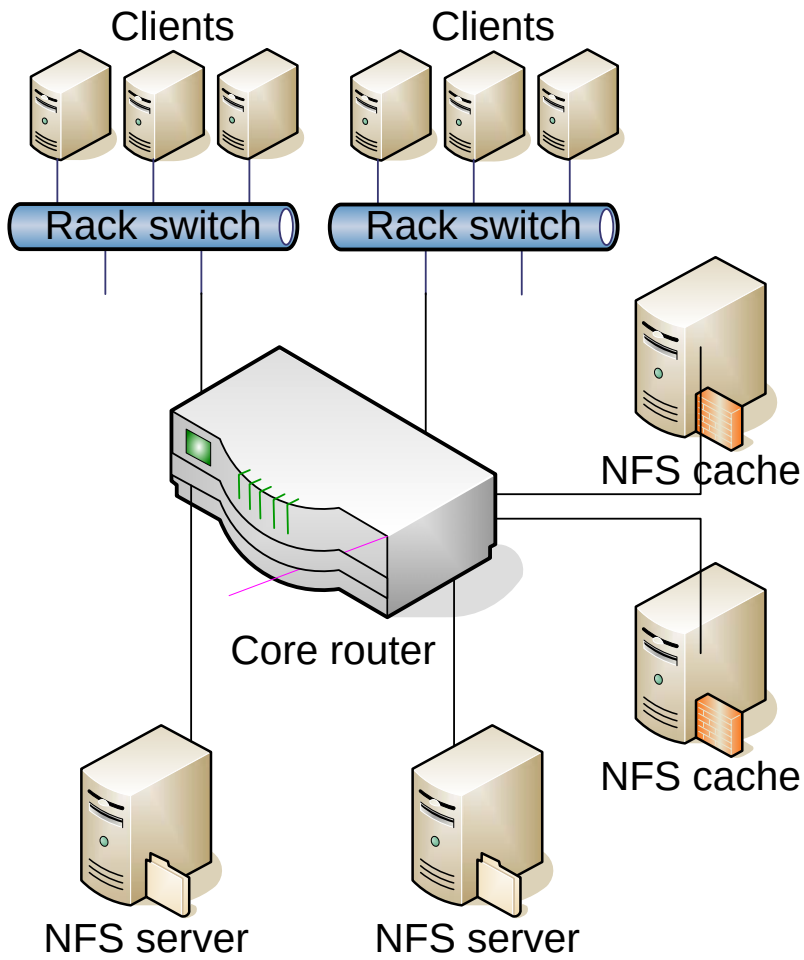
data → information

Details in paper

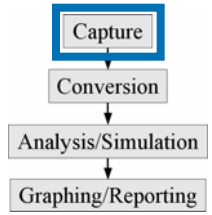
Tools, traces are open source

The customer

- Feature animation (movie) company
  - Read models, textures, animation curves
  - Write intermediates and pictures
  - ~3 years/movie
- Dramatis personae:
  - Thousands of clients (render-farm)
  - Tens of NFS servers
  - Twenties of NFS caches
  - Many rack switches
  - Few core routers



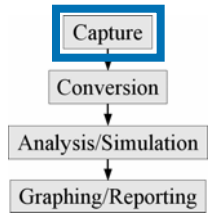
Capture (2003)



- Challenge:
  1. Non-intrusive data capture
  2. Parse readdir, etc.
  3. Enable offline conversion
  4. NFS traffic bursts >1Gbit/s
  5. Prefer long capture times (days)

- Solution:
  1. Port mirroring on switch
  2. Full packet capture
  3. Capture to parallel JBOD
  4. Special Linux-specific capture tool (*lindump*)
  5. Dynamic compression via tmpfs buffer

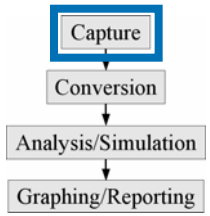
LABS hp



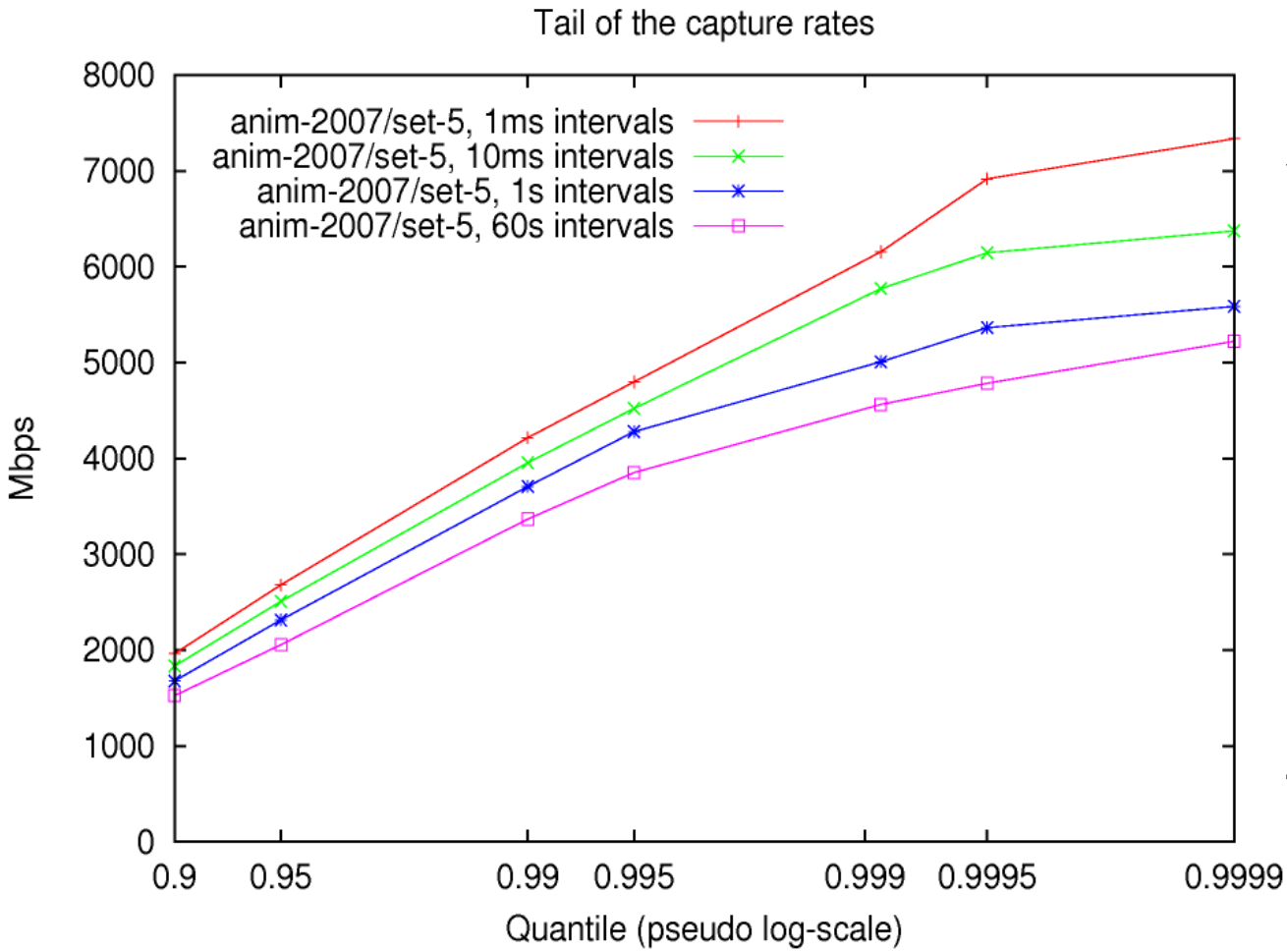
Capture, improved

- 2004: new switches with smaller buffers
  → 10Gb/s network interface card
  In-driver packet capture (*driverdump*)

- 2007: sustained 5Gb/s
  → Special capture card (*endacedump*)
  Integrated dynamic compression



Capture: observed rates

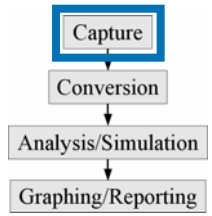


Measured workload is bursty

Capture tool can sustain 5Gb/s

Capture tool can burst up to 7.5Gb/s

LABS hp



Capture: discussion

- **No more papers reporting packet drops**

Expected data rate, available resources

<1Gbit/s

< 3Gbit/s,
Kernel hacker

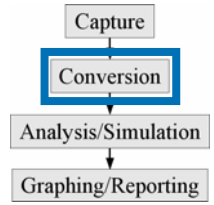
<10Gbit/s,
Capital budget

Use *lindump*
(2003 tool)

Implement *driverdump*
(2004 tool)

Purchase *endacedump*
(2007 tool)

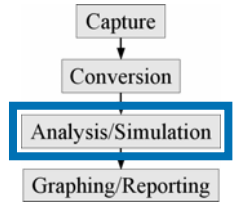
Conversion



- Challenge:
  1. Flexible logical representation
  2. Efficient physical representation
  3. Rapid trace conversion
  4. Trace anonymization

- Solution:
  1. Relational data model, multiple tables
  2. DataSeries structured serial data format
  3. Two-pass parallelism
  4. Reversible encryption

Analysis techniques

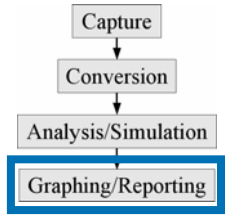


- Challenge:
  1. Huge (50 billion row) data sets
  2. Large intermediates
  3. Many possible grouping options
  4. Bursty, non-normally distributed data

- Solution:
  1. Custom DataSeries analysis
  2. Streaming analysis
  3. Develop efficient data cube
  4. Use approximate quantiles

LABS hp

Graphing/Reporting techniques



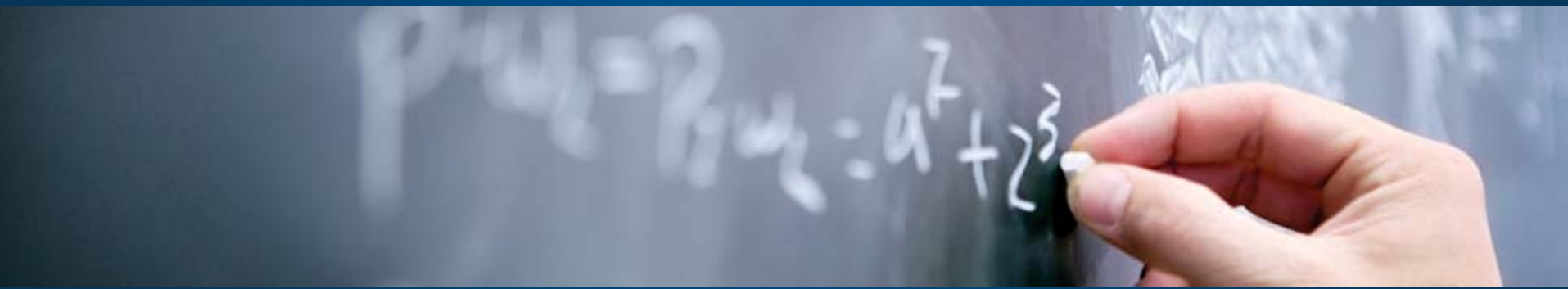
- Challenge:
  1. Moderate-size summary data
  2. Many possible graphs
- Solution:
  1. Store data in SQL database
  2. Select with mercury-plot

Example mercury-plot command:

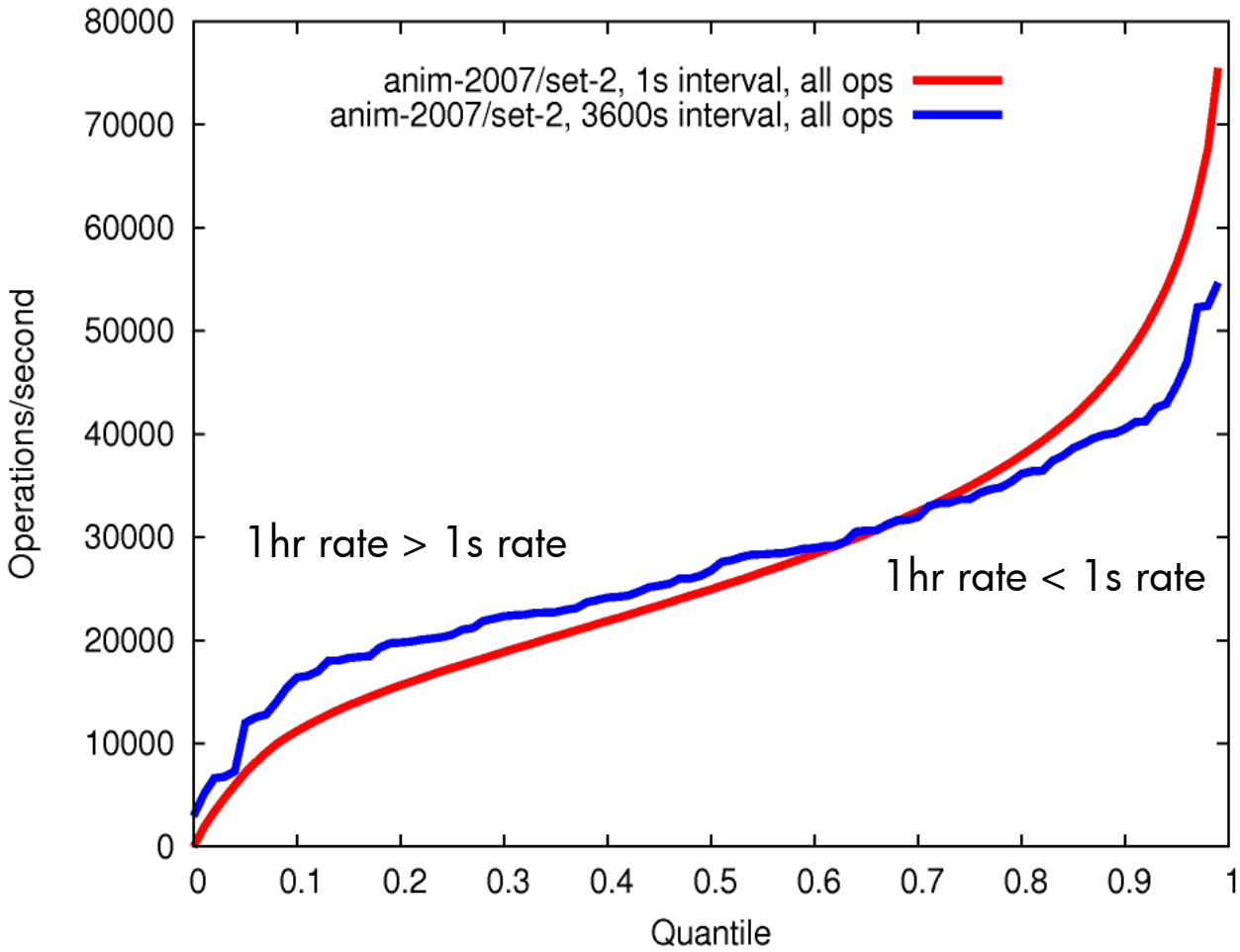
```
plot quantile as x, value as y from nfs_hostinfo_cube
   where operation = 'read' and direction = 'send'
```


LABS hp

Collect more traces

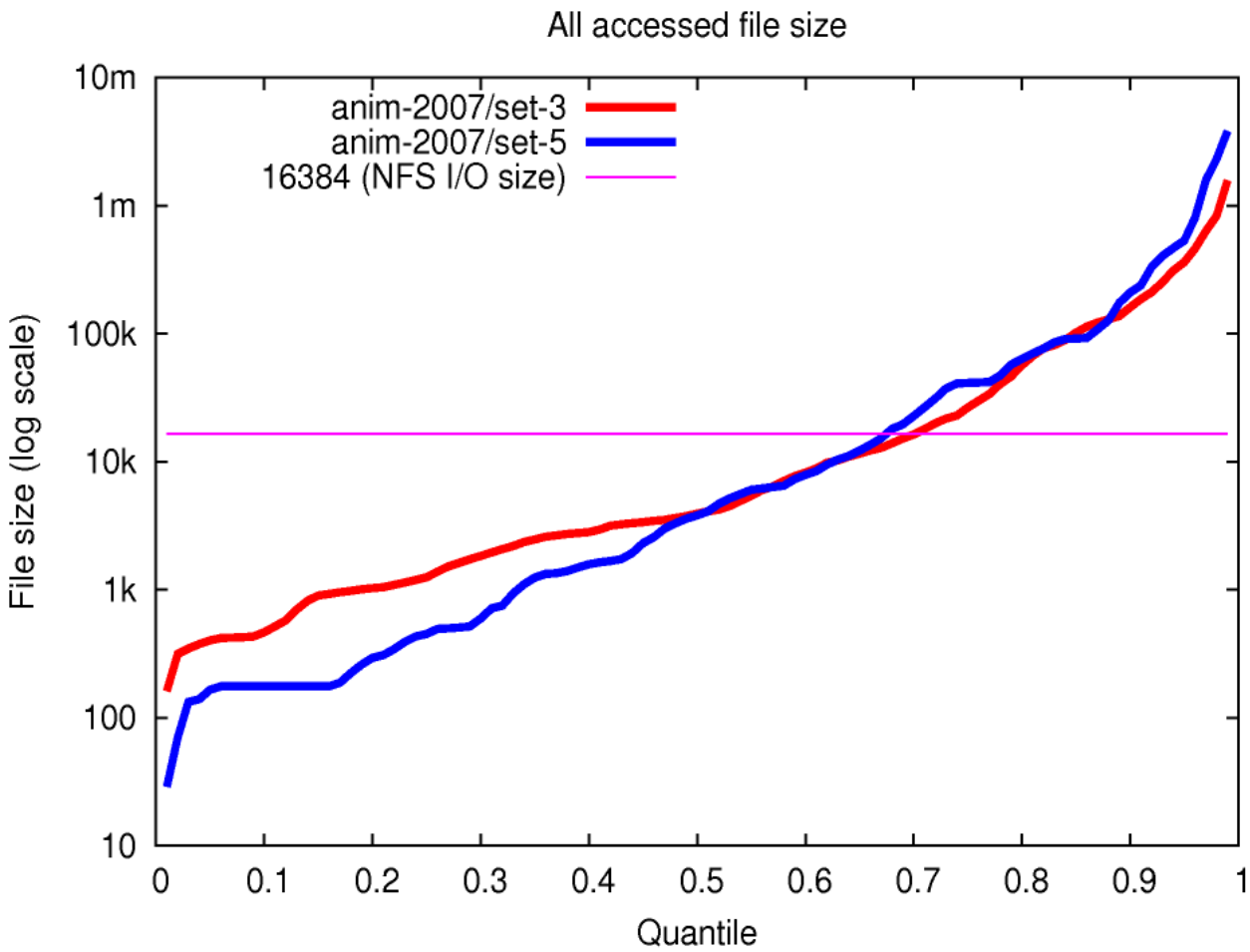


Analysis: distribution of operation rate



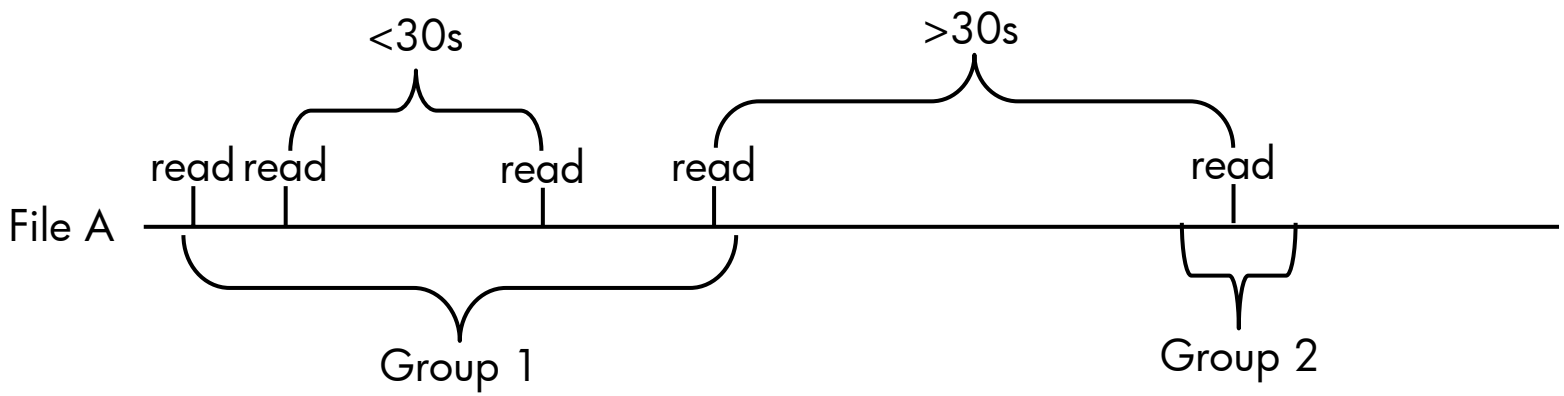
- Shows NFS-level burstiness
- Validates use of quantiles rather than mean and stddev

Analysis: distribution of file sizes



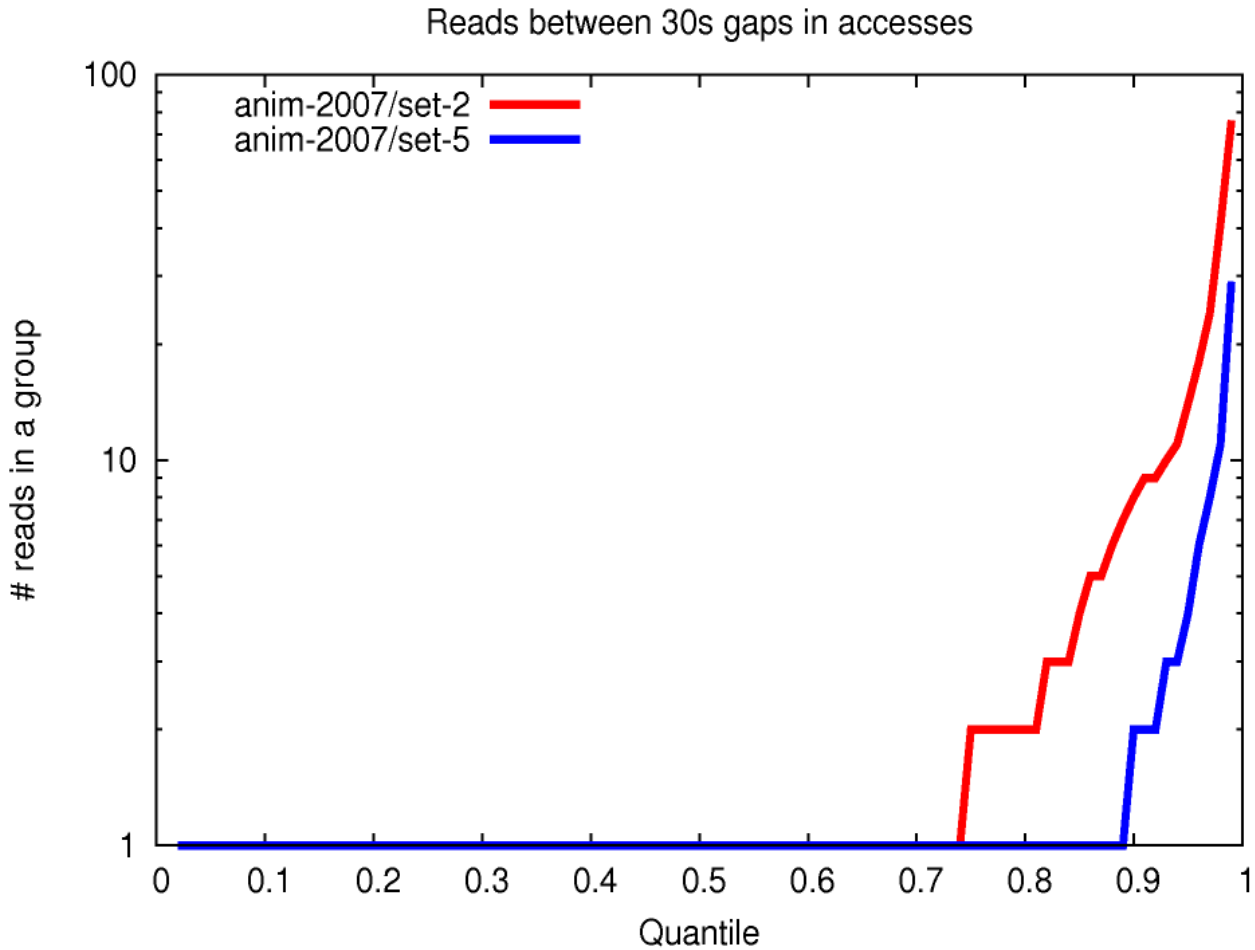
- Each accessed file counted once
- Most files are small
- Moderately wide size distribution
- Horizontal line is NFS read and write size

Analysis: reads in a single group



Each group is the set of reads with a maximum inter-read gap of 30 seconds

Analysis: reads in a single group



- Most I/Os all alone
- Side effect of many small files.
- Occasional large groups (~100 I/Os)
- Need cross-file prefetching

Conclusion

- Capture techniques

  → no more packet loss
- Conversion and analysis techniques

  → handle huge datasets on moderate hardware
- Workload is very different:
  - Very intense
  - Small files
- Much more detail and discussion in paper
- Tools and traces open source

Questions?

Author/Speaker: eric.anderson4@hp.com

Software: http://tesla.hpl.hp.com/opensource/

Datasets: http://apotheca.hpl.hp.com/pub/datasets/animation-bear/

http://iotta.snia.org/traces/list/NFS

Tracing BoF: 8:30-9:30 pm, San Francisco A


LABS hp