Department of Computer Science

HPS

University of Reading

Tracking User-Perceived I/O Slowdown via Probing

Image under free license (CC0)

Limitless Storage
Limitless Possibilities

https://hps.vi4io.org

Julian M. Kunkel, Eugen Betke

HPC-IODC

2019-06-20

LIMITLESS **POTENTIAL** | LIMITLESS **OPPORTUNITIES** | LIMITLESS **IMPACT**

Outline

Motivation

- Performance of shared file system is load dependent
  - Also background activity may cause delays
- Difficult to judge: observed performance is slower/faster than normal
  - A subcomponent of a file system may be loaded (e.g., metadata)
- Users/staff may wonder for the cause of the experienced performance
  - "Is that caused by my application?"
  - Can lead to support requests
- Maybe a quantification of the file system load similar to `uptime` would help?

Approach

- Running a minimal invasive benchmark (a probe) with a high frequency
  - Mimic the user-visible client behavior
  - Measuring latency for metadata and data operations
  - Overloaded servers will delay the response time
- Generate and analyze generated statistics
- Derive a slowdown factor (file system load)

Why not use server-sided information?

- Client perspective is different (involves network, too)
- Servers experience different types of IO
  - We need to compare standard values!
- Tracking response latencies for op type/size histograms would do
  - Vendors: integrate such a reporting (vendor neutral API!)

Outline

Performance Measurement

University of Reading

Preparation

- Data: Generate a large file (e.g., $>$ 4x main memory of the client)
- Metadata: Pre-create a large pool of small files (e.g., 100k+ files)

Benchmarks

- Repeat the execution of the two patterns every second
- DD: Read/Write a random 1 MB block
- MD-Workbench: stat, read, delete, write a single file per iteration
  - Allows regression testing, i.e., retain the number of files
  - *J. Kunkel, G. Markomanolis. Understanding Metadata Latency with MDWorkbench.*

Executed as Bash script or an integrated tool:
`https://github.com/joobog/io-probing`

Discussion of the Approach

University of Reading

- Monitoring load: load is with 2 MB/s and 5 MD ops/s negligible
- Resource cost: the monitor needs a client node to execute
  - Use it exclusively or share it?
  - Adds costs
- Representativity: systems have 100's of I/O servers
  - For 100 servers, after 100s statistically most servers were probed
- Understanding high-frequency of latency changes
  - Apply statistics for data reduction
- Interpretation of the slowdown factor
  - Must be easy to understand
  - Must adjust over the life time of a system

Computing the Slowdown

- The user-perceived slowdown is defined as $s = \frac{t_{observed}}{t_{expected}}$
- The expected time is the `median` over the observation period
  - As performance many change due to user load and system aging, the value could be updated, e.g., every month or after system upgrades
- The observed time is the latency from the benchmark
  - To increase robustness computing a statistics over a sliding window
  - The computation can be adjusted to the use case or to ensure a service level
    - median: indicating the slowdown of the faster 50% of operations
      That also means that the other 50% operations are slower than that
    - quantiles, e.g., Q90: slowdown for 90% of the operations

Outline

Test Systems

University of Reading

- JASMIN, the data analysis facility of the UK
  - Precreation: 200k files, 200 GB data file
  - 60 days of data
  - Script runs exclusively on a node
- Archer, the UK national supercomputer service
  - Precreation: 200k files, 200 GB data file
  - 30 days of data
  - Script runs on a shared interactive node
- Mistral, the HPC system at the German Climate Computing Centre
  - Precreation: 100k files, 1.3 TB data file
  - 18 days of data
  - Tool runs on a shared interactive node

Understanding the Timeseries

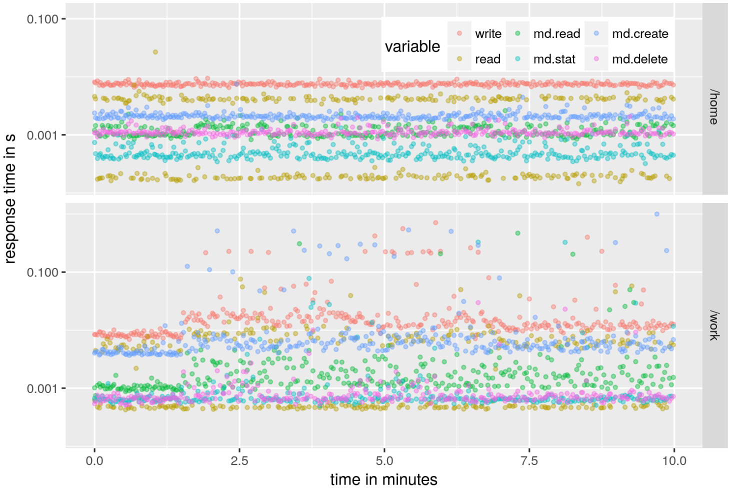
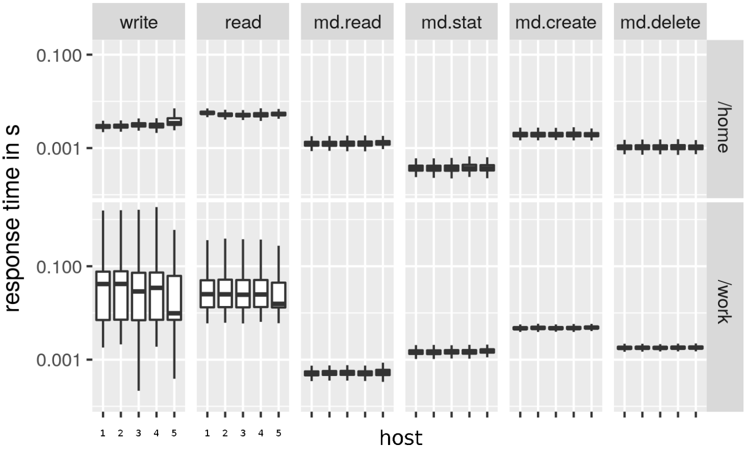


Figure: Jasmin every data point for 10 minutes from one node

- Every probe (1s) for 10 min
- For two file systems
- Home is stable
- Work shows irregularities

Robustness of Statistics on Hosts

University of Reading



- 5 hosts: 1 hour data
- 2000 measurements
- Statistics are similar
- Home is stable
- Work shows irregularities

Figure: Jasmin boxplot statistics from five different hosts

IO-500 Response Time on Archer

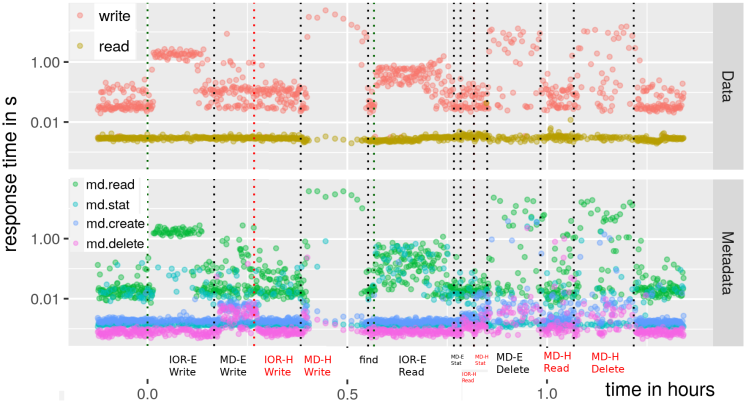


Figure: Response time (all measurements)

- Run on 100 nodes
  score 8.45
- The IO-500 various phases
  Data and metadata heavy
- First, all measurements

Validating Slowdown on All Measurements

University of Reading

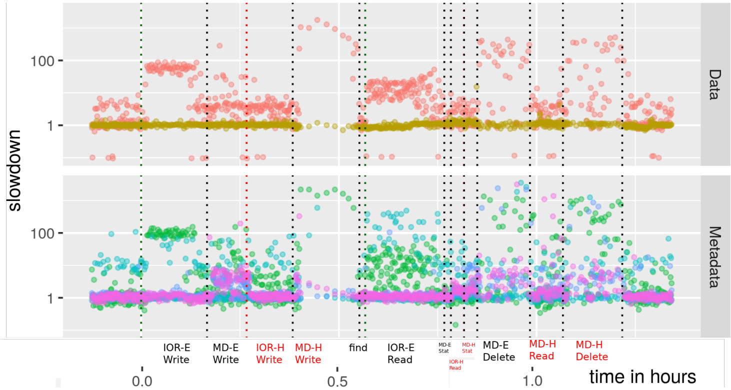


Figure: Slowdown (all measurements)

- Computed median slowdown
  Expected: median of 30 days
- Influence of phases is visible
- MDHard 1000x slowdown
  Influences data latency!
  10s of seconds latency
- IOREasy 100x slowdown
- IORHard not too much
- Data read is stable

Validating Slowdown: Reduced Data

University of Reading

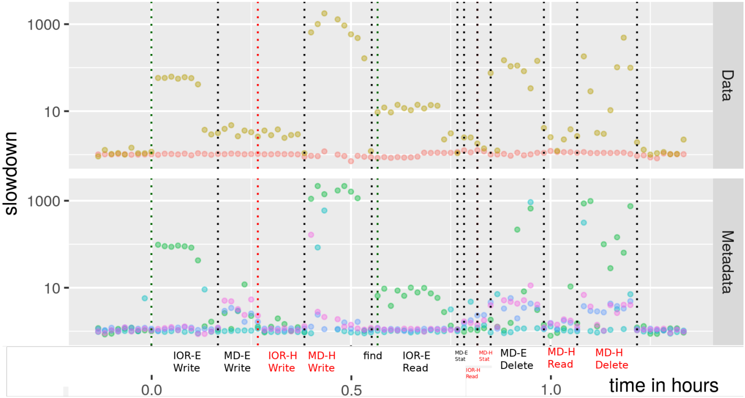
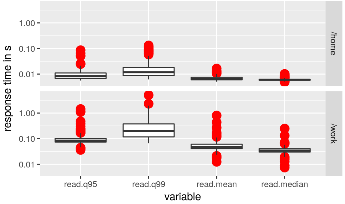


Figure: Slowdown (60s mean statistics)

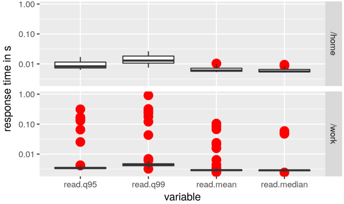
- Data reduction: 60s mean
- More robust, clearer to see

Boxplots for 4h Statistics

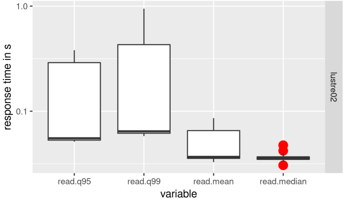
University of Reading



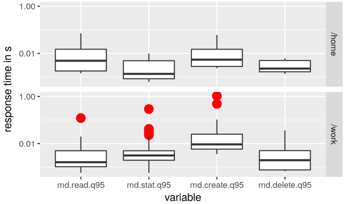
(a) JASMIN read (60 days)



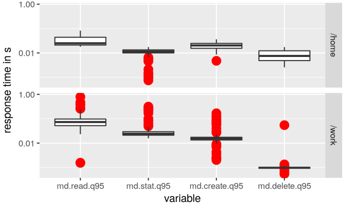
(b) Archer read (30 days)



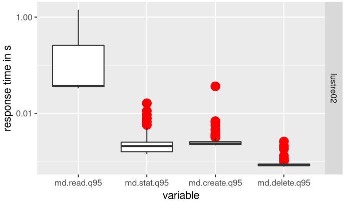
(c) Mistral read (18 days)



(d) JASMIN metadata



(e) Archer metadata



(f) Mistral metadata

Timelines of 4h Statistics

University of Reading

Figure: JASMIN data timeline

- Compare different statistics
- Maximum latency of 100s (!)
- Q99 and Q95 are more robust
  Still, delay is not acceptable

Timelines of 4h Statistics

University of Reading

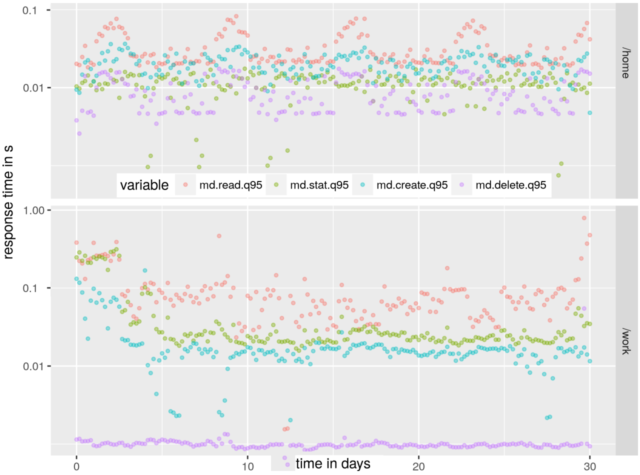


Figure: Archer metadata timeline

- Use Q95, 5% ops are slower
- Home: regular slower except for stat
- Work: slower at beginning

Timelines of 4h Statistics

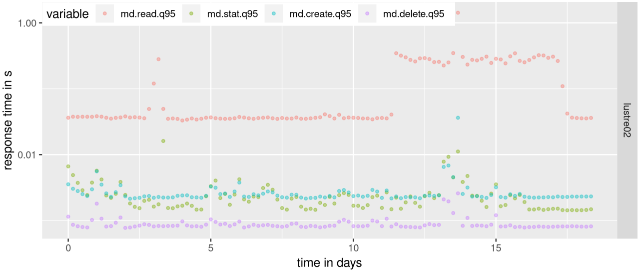


Figure: Mistral metadata timeline

- Use Q95, 5% ops are slower
- Change in behavior at day 12
  Reason: unknown

Slowdown for 4h Statistics

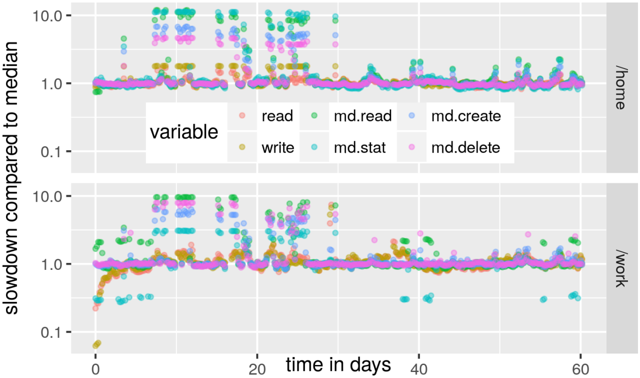


Figure: JASMIN, computed on 4 hour intervals

- Slowdown: Using the median
- Typically value is 1
- Sometimes 10x slower
- Values below 1, unusual (caching)
- Good to see long-term issues

Slowdown for 10 Min Statistics

University of Reading

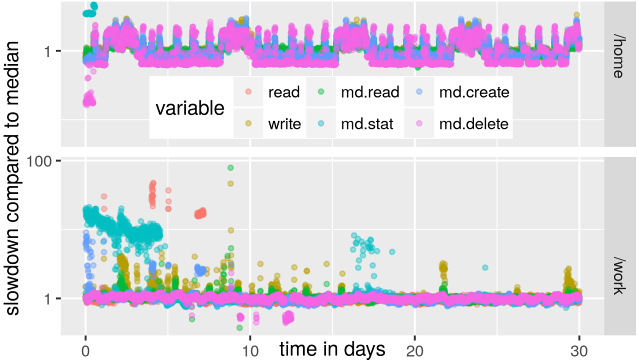


Figure: Archer, computed on 10 min intervals

- Home: periodic behavior visible
  Weekly but also daily
- Work: sometimes 10x slower
- Could be made accessible to users / data center-staff

Outline

University of Reading

Summary

University of Reading

- Understanding user-slowdown by probing
  - Measuring latency from client-side
  - Reducing frequency using statistics
- Statistical data reduction depending on use case
  - Quantile depending on service level
  - Interval depending on needs
  - No silver bullet at the moment
- Effective to identify slow periods
  - Validated with IO-500 behavior
  - Observed background activity
- Some IOs are very slow (100s)!
  - Did you ever wait so long for a touch/cat?
- Future work: long-term studies and utilize splines