GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

HPS
https://hps.vi4io.org

Julian M. Kunkel

Determining I/O-Weather



Definition: I/O-Weather

Weather

- The state of the atmosphere at a particular place and time as regards heat, cloudiness, dryness, sunshine, wind, rain, etc.

I/O-Weather

- The state of the **I/O-system** at a particular node and time as regards observable performance characteristics for workloads

Motivation

- I/O-Weather is mostly unknown to us until we try to utilize I/O
  - Server-sided monitoring doesn't neccesarily help to prepare
    - System down, e.g., Thunderstorm
    - But somewhat loaded may be OK for many workloads?
- Difficult to judge weather: is it "rainy" or sunny
  - Complex system state, concurrent usage of the shared file system
  - A subcomponent of a file system may be loaded (e.g., metadata)
  - Is it due to software updates/intermediate or permanent hardware issues?
- Users/staff may wonder for the cause of the experienced weather
  - "Is that caused by my application?" Can lead to support requests!

Alternatives

Questions

- Number of metrics
- Metric definition
  - Client or server?
  - Relative, absolute?
  - Capturing which parts of the storage?

How many metrics?

- Could use a single weather indicator or multiple
  - Single summarizes weather behavior - rainy
    - Maybe a quantification of the file system load similar to `uptime`?
  - Multiple scores help to understand md/data limits
    aka, humidity x%, sunnlevel 3, windy 50 miles/h
  - Both scores probably helpful

Client vs. Server

Client side

- Pro: Client perspective is accurate representaion of performance
- Drawback: we cannot afford to measure on every client(?) to all servers disturbance by communication ...

Server side

- Pro: Well defined, fewer nodes
- What performance information data can we get?

- Vendors: please provide metrics for response times (additional to throughput)

Alternatives: Relative vs. Absolute

Absolute

- Just reports the observation - like system monitoring
- Problem: 50 GByte/s, what is the meaning of this?

Relative

- Relative to e.g., best-performance, e.g., 95% of IO500 easy run
- Problem: High performance on system may mean the system is overloaded or that everything is fine and we just use it well...
  - It is good to see 95% utilization but if users need 5x the time for I/O?
- Maybe user slowdown represents better user-perceived weather?

Alternatives: Risk Metric[1]

- Risk: Quotient of operation statistics from scaled average
  - e.g., average number of IOs, throughput
- Could combine multiple risk metrics into one risk score (for MDS, DS or both)
- High score means system is more utilized than usual - user perception
- Pro: Easy to obtain and compute
- Drawback: Risk does not increase further on a saturated system
  - With more requests you can get even slower

[1] Paper: Analysis of parallel I/O use on the UK national supercomputing service, ARCHER using Cray's LASSi and EPCC SAFE (Andrew Turner, Dominic Sloan-Murphy, Karthee Sivalingam, Harvey Richardson, Julian Kunkel), CUG, Montreal, Canada, 2019-10-11

Alternatives: Slowdown[2]

- Utilizing probing, i.e., periodic small-scale micro-benchmarks like a calibrated external calibrated measurement device
- Relate performance with either average behavior or best-case
- Pro: Actual slowdown compared to expectation
- Drawback: need to run on a node, e.g., login node
  Jitter in many data points - should do some smoothing

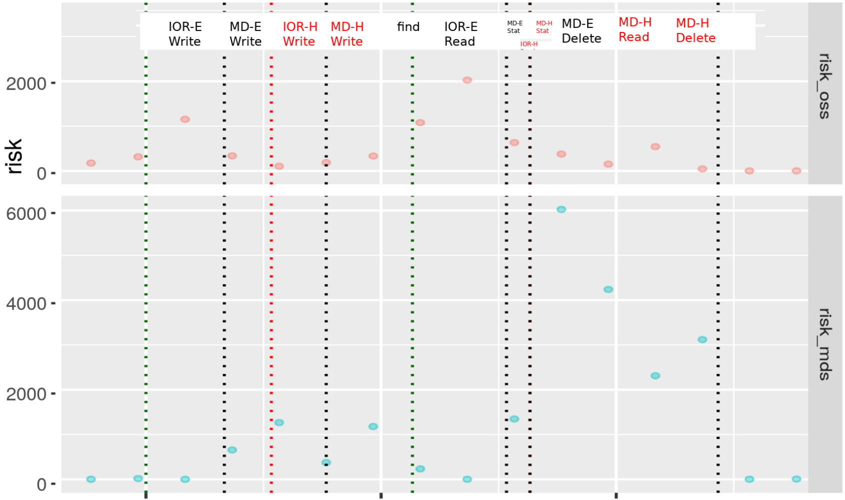
[2] Paper: Tracking User-Perceived I/O Slowdown via Probing (Julian Kunkel, Eugen Betke), In High Performance Computing: ISC High Performance 2019 International Workshops, Frankfurt/Main, Germany, June 20, 2019, Revised Selected Papers, Lecture Notes in Computer Science, Springer, HPC-IODC workshop, ISC HPC, Frankfurt, Germany

Outline

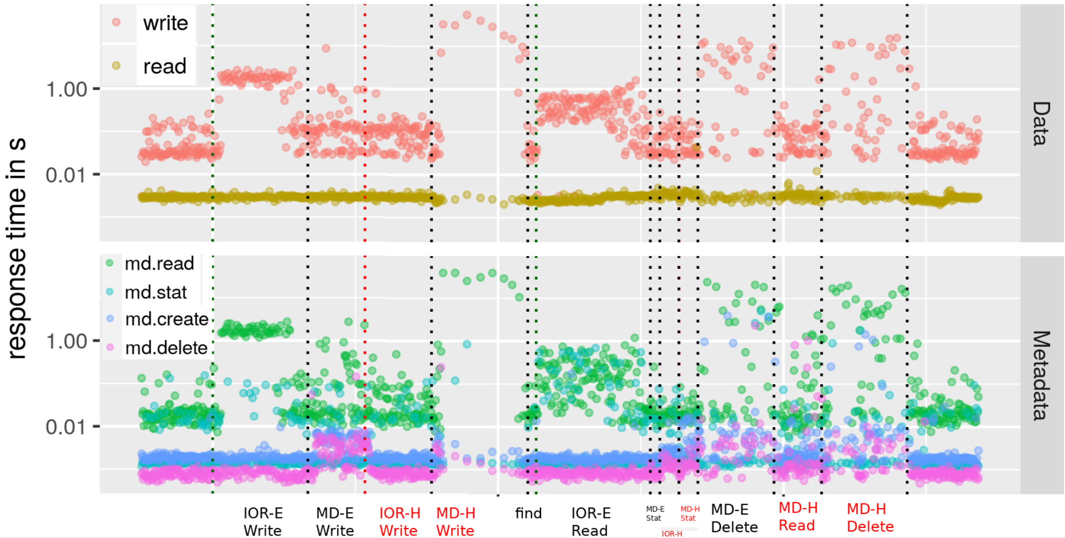
Evalution

- Let's look at some observations
- Risk, response time, slowdown
- Data was measured on EPCC's Archer system
- IO500 run on 100 nodes
  - IOREasy write: 12.973 GB/s
  - MDEasy write: 58.312 kiops
  - IORHard write: 0.046 GB/s
  - MDHard write: 34.324 kiops
  - find: 239.300 kiops
  - IOREasy read: 9.823 GB/s
  - MDEasy stat: 64.173 kiops
  - IORHard read: 1.880 GB/s
  - MDHard stat: 63.166 kiops
  - MDEasy delete: 13.195 kiops
  - MDHard read: 20.222 kiops
  - MDHard delete: 10.582 kiops
  - SCORE: 8.45

Risk (measured every 5-6 mins)



Response time measured by micro-benchmarks



Slowdown

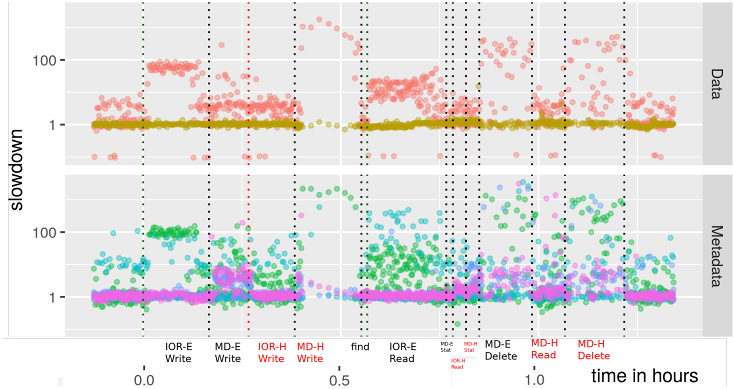
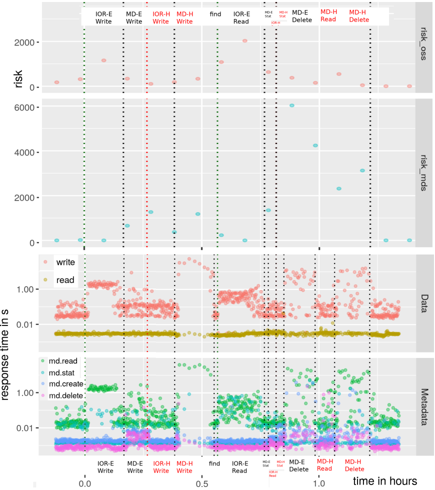


Figure: Slowdown (all measurements)

- 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

Conclusions



- I/O-Weather can illustrate abnormal behavior
- Additional metrics to "performance statistics"
- Both risk and slowdown metrics are useful
- Interpreation depends on use-case

Probing Approach

- Many sites run periodic regression tests, e.g., nightly
  - Helps to identify performance regressions with updates
- Instead, we run a non-invasive benchmark (a probe) with a high frequency
  - Mimic the user-visible client behavior
  - Measuring latency for metadata and data operations
- Generate and analyze generated statistics
- Derive a slowdown factor (file system load)

Probing: Performance Measurement

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`

Test Systems

- 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

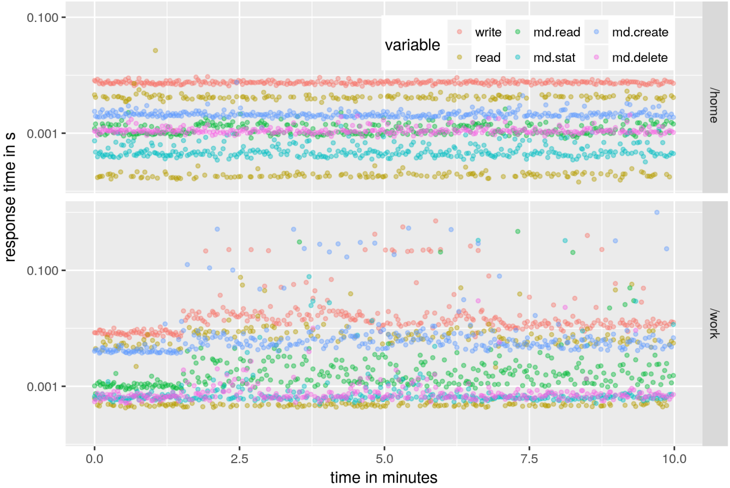


Figure: Jasmin every data point for 10 minutes of one node

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

IO-500 Response Time on Archer

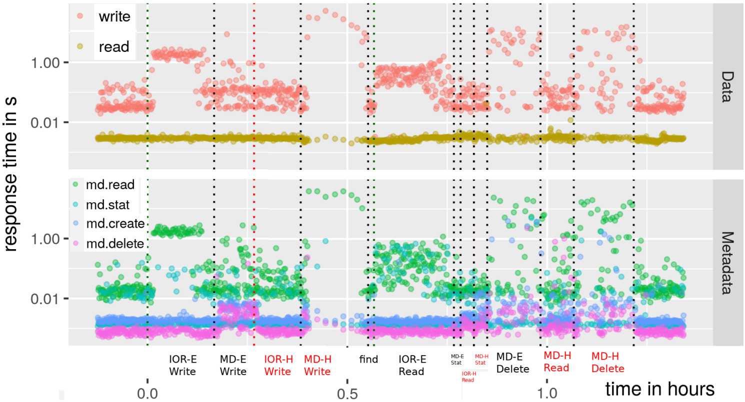


Figure: Response time (all measurements)

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

Validating Slowdown on All Measurements

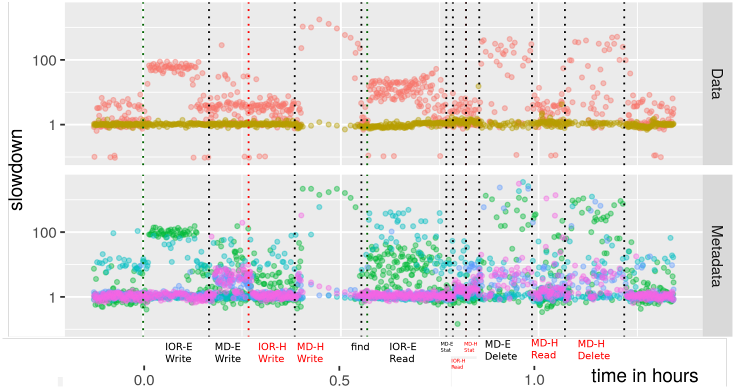


Figure: Slowdown (all measurements)

- 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

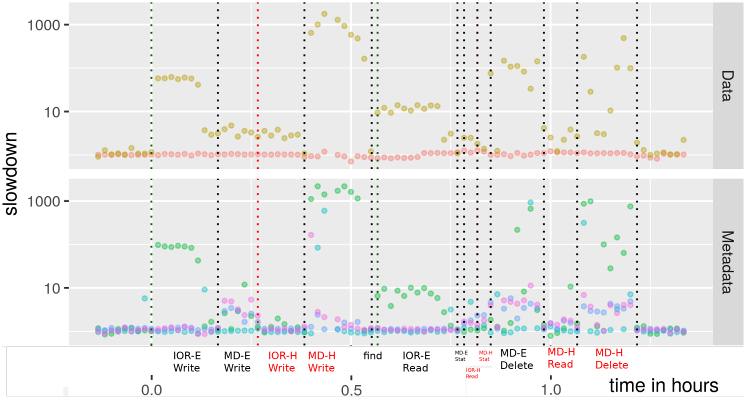


Figure: Slowdown (60s mean statistics)

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

Timelines of 4h Statistics

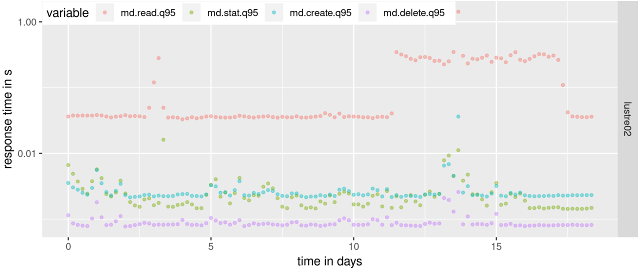


Figure: Mistral metadata timeline

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

  Reason: unknown

Slowdown for 4h Statistics

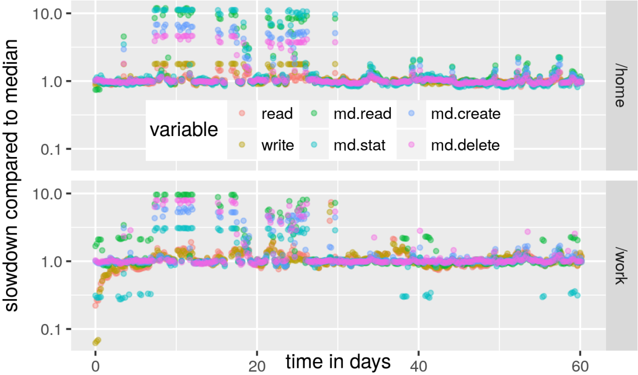


Figure: JASMIN, computed on 4 hour intervals

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