Metadata Issues

Julian M. Kunkel

Deutsches Klimarechenzentrum GmbH (DKRZ)

2017-06-25

Metadata Benchmarking Issues

- Latency + variability is relevant for interactive usage
  - Q(99): Waiting time of 1 vs. 100s with same throughput is different
- Unrealistic system internal optimization / cheating shall be prevented
  - Batch create is typically well optimized by file systems
- Often: long setup time to actually benchmark useful scenario
  - It would be nice to (nightly) run a short MD benchmark
  - Compare performance across measurements...

MD-REAL-IO

- An open source benchmark `https://github.com/JulianKunkel/md-real-io`
- Plugins for POSIX, MPI-IO, Postgres, MongoDB, S3
- Operates on shared datasets / objects
- Measures overall runtime but optionally individual operations (latency)
- Phases:
  - Precreate a working set (optional)
  - Benchmark
    - stat, open, read, close, unlink a single object from the working set
    - open, write, close a new object ⇒ the working set stays the same throughout the test
  - Cleanup (optional, one can run the test repeatedly over the working set)
- Interpretation:
  - Multiple FIFO producer/consumer systems processing small data
  - Interactive usage from many users on a HPC system

Example Output

Here: Local EXT4 on an SSD via NVME interface

MDTest

```
Operation                         Max
---------                         ---
File creation     :         31778.271
File stat         :        336661.620
File read         :        145469.906
File removal      :         51664.232
Tree creation     :         18551.167
Tree removal      :             1.983
```

MDRealIO

precreate 24.4s 40982.9 iops/s 1 dset 1000000 obj 0.041 dset/s 40982.9 obj/s 152.4 Mib/s (0 errs)

benchmark **0.7s 59312.1 iops/s** 10000 obj 14828.0 obj/s 110.3 Mib/s (0 errs)

benchmark **0.7s 55441.3 iops/s** 10000 obj 13860.3 obj/s 103.1 Mib/s (0 errs)

benchmark **0.8s 52715.3 iops/s** 10000 obj 13178.8 obj/s 98.0 Mib/s (0 errs)

cleanup 11.1s 90087.1 iops/s 1000000 obj 1 dset 90087.0 obj/s 0.090 dset/s (0 errs)

Metadata results on an USB-Stick

An external USB (1) stick connected to the laptop, 4 threads, 130k files

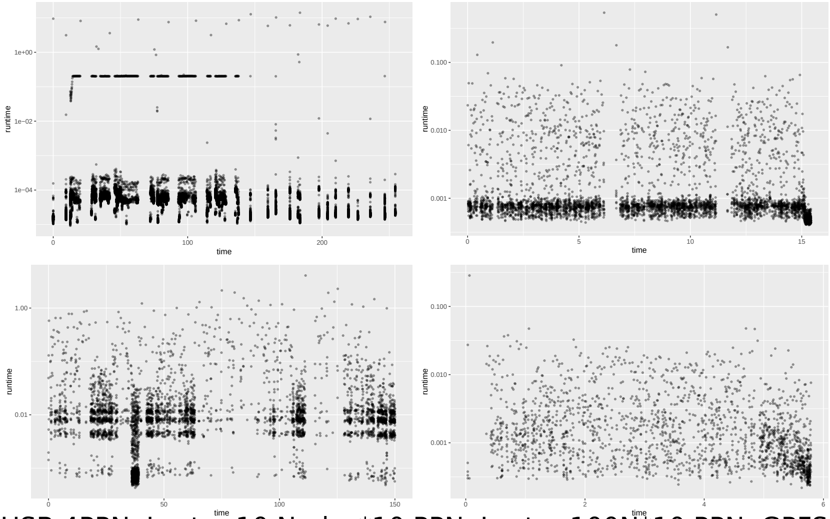
```
precreate 262.8s 380.5 iops/s 4 dset 100000 obj 0.015 dset/s 380.5 obj/s 1.4 Mib/s (0 errs)
benchmark 199.8s 200.2 iops/s 10000 obj 50.1 obj/s 0.4 Mib/s (0 errs)
benchmark 99.0s 404.0 iops/s 10000 obj 101.0 obj/s 0.8 Mib/s (0 errs)
benchmark 302.2s 132.4 iops/s 10000 obj 33.1 obj/s 0.2 Mib/s (0 errs)
cleanup 227.5s 439.6 iops/s 100000 obj 4 dset 439.5 obj/s 0.018 dset/s (0 errs)
```

Variability between runs indicates an optimization issue (write-back timer)

MDTest

```
Operation                    Max
---------                    ---
File creation     :        463.342
File stat         :    2778790.782
File read         :        928.824
File removal      :        296.219
Tree creation     :       3173.428
Tree removal      :          0.282
```

Latency Results: Precreate Timelines (1 process)



Throughput IOOPs per proc:

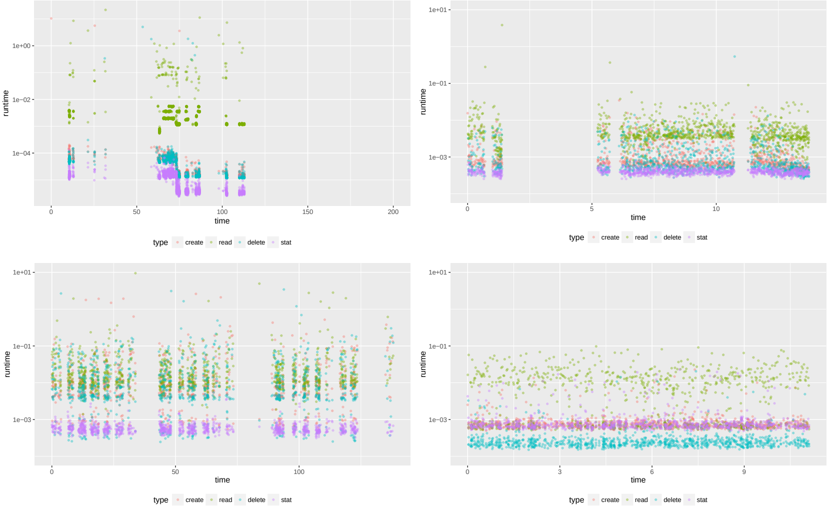
USB: 95

Lustre:
10 Nodes: 250
100 Nodes: 26

GPFS: 113

USB 4PPN, Lustre 10 Nodes*10 PPN, Lustre 100N*10 PPN, GPFS 10N*10 PPN

Latency Results: Benchmark Timelines (1 proc per experiment)



Throughput IOOPs per proc:

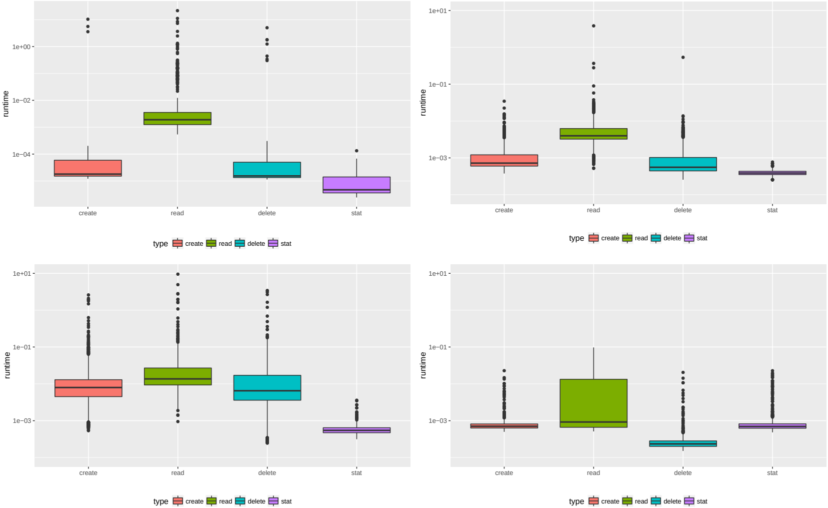
USB: 100

Lustre:
10 Nodes: 230
100 Nodes: 31

GPFS: 230

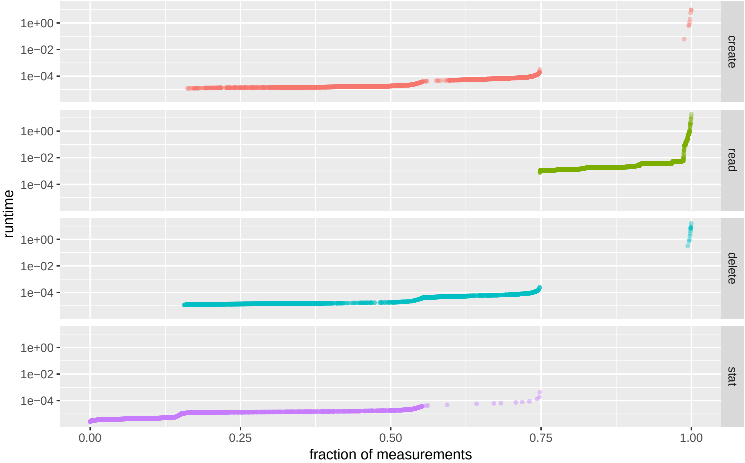
USB 4PPN, Lustre 10 Nodes*10 PPN, Lustre 100N*10 PPN, GPFS 10N*10 PPN

Latency Results: Benchmark Boxplots (1 process)



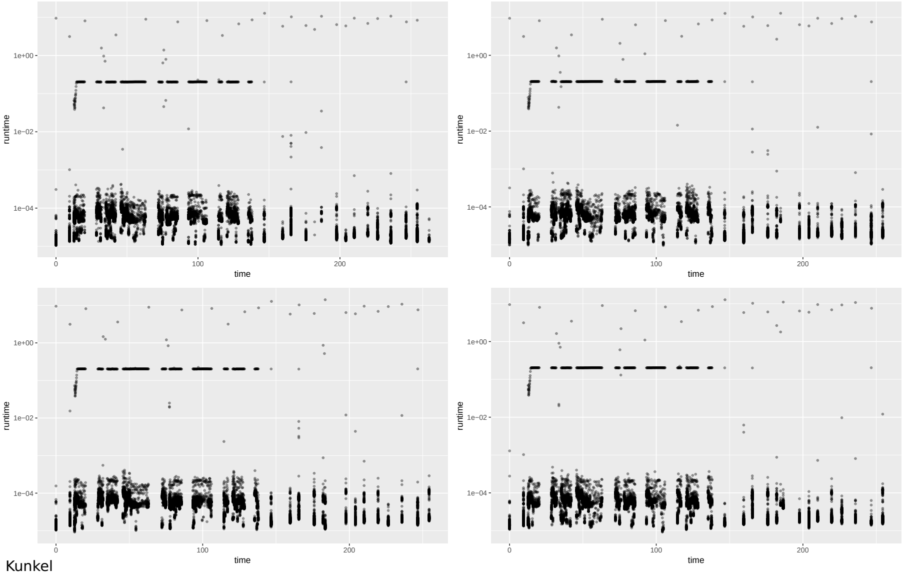
USB 4PPN, Lustre 10 Nodes*10 PPN, Lustre 100N*10 PPN, GPFS 10N*10 PPN

Characteristics of Latency (Here USB-Stick)

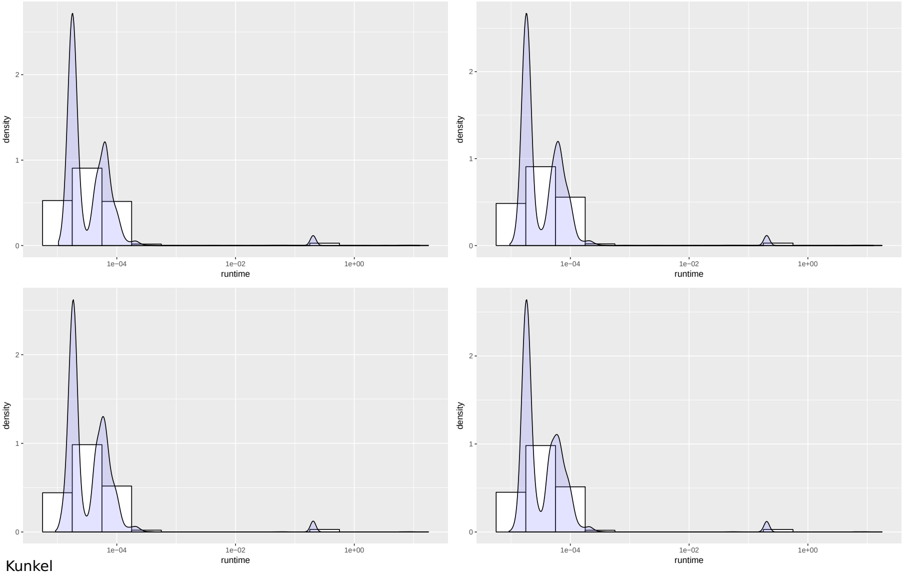


A MD benchmark should output (MD throughput) & latency(max)

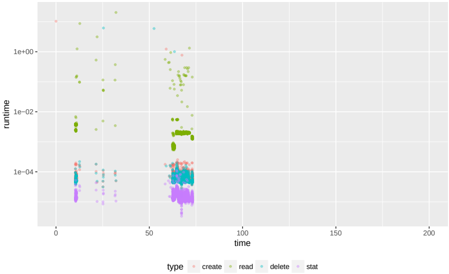
Latency Results: Precreate Timelines (all processes), similar speed

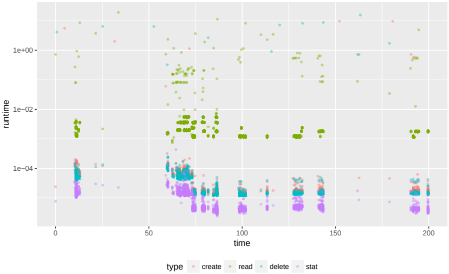


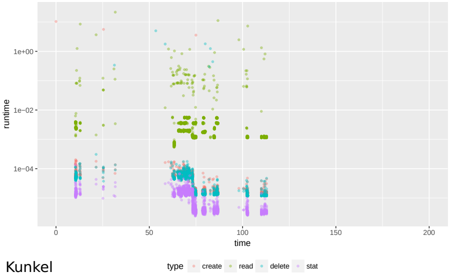
Latency Results: Precreate Histograms / Density

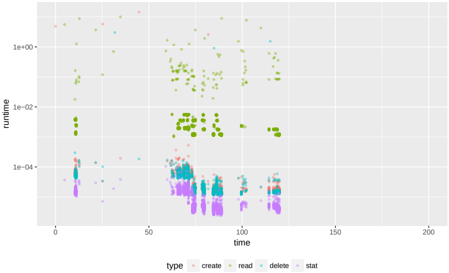


Latency Results: Benchmark Phase Timelines

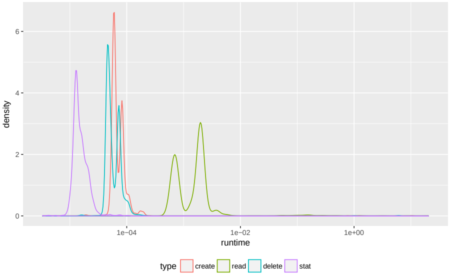


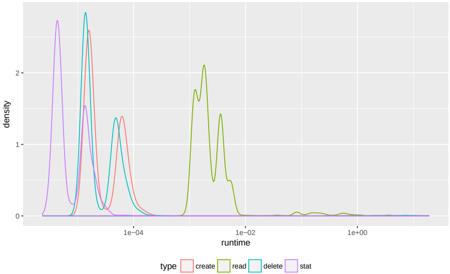


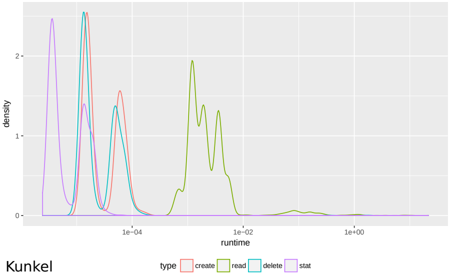


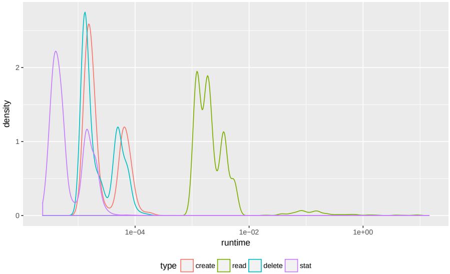


Latency Results: Benchmark Phase Histograms / Density

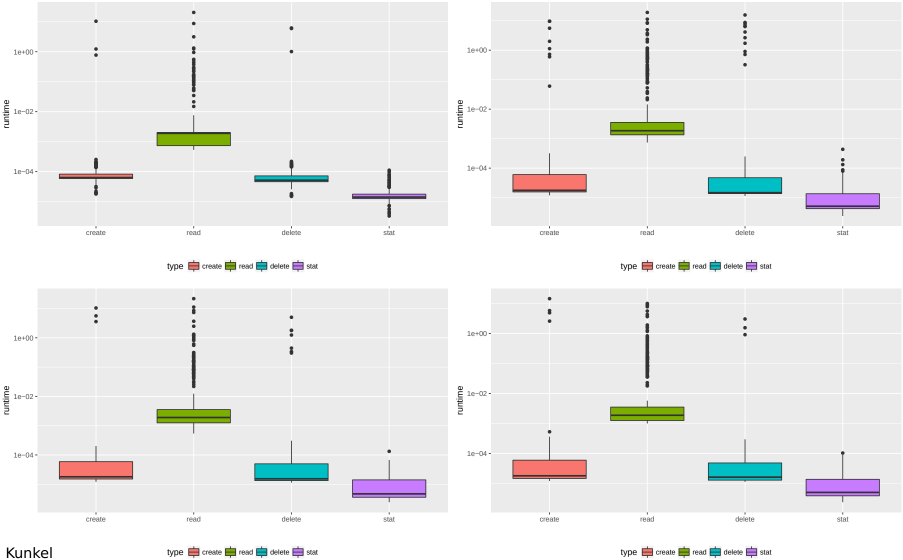




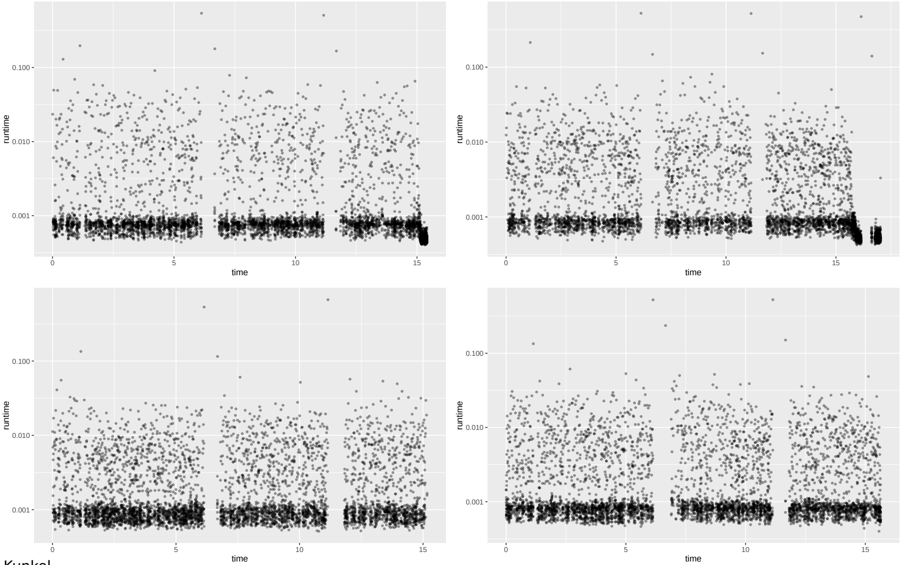




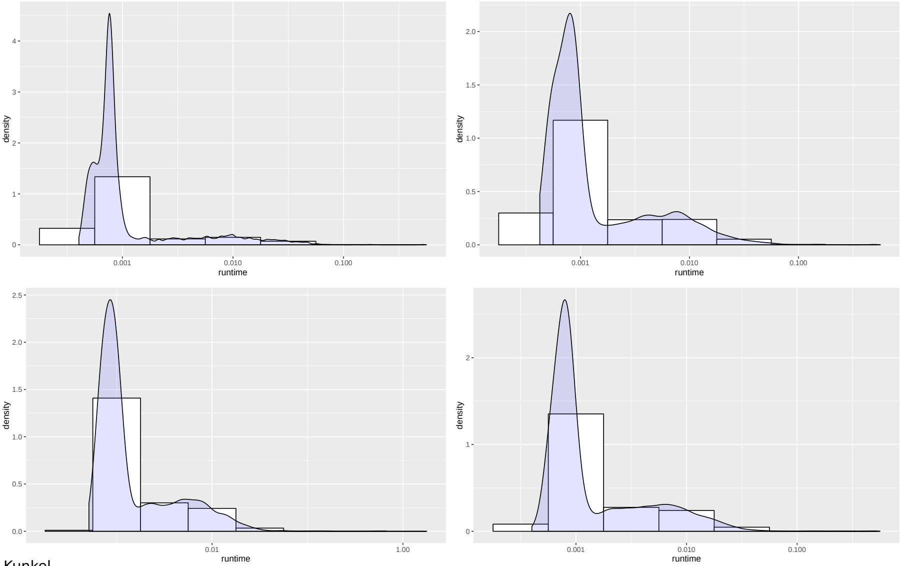
Latency Results: Benchmark Phase Boxplots



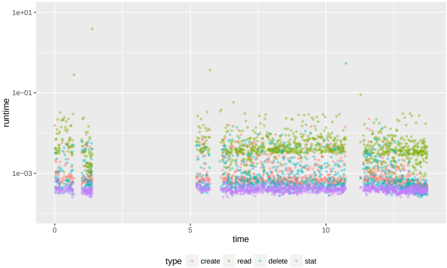
Latency Results: Mistral 10 Nodes, 10 PPN, Precreate 0-3

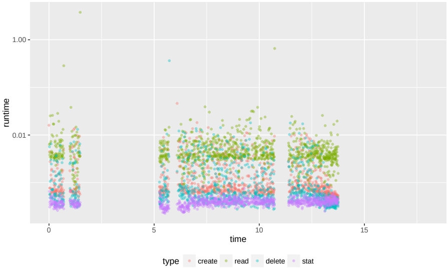


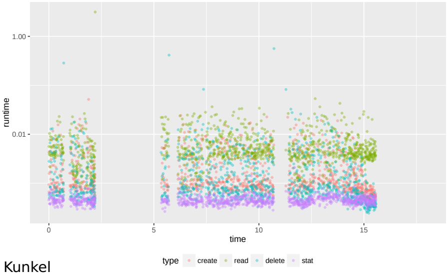
Latency Results: Precreate Histograms / Density

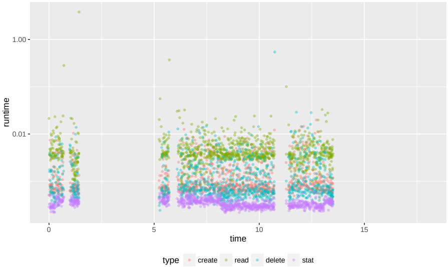


Latency Results: Benchmark Phase Timelines

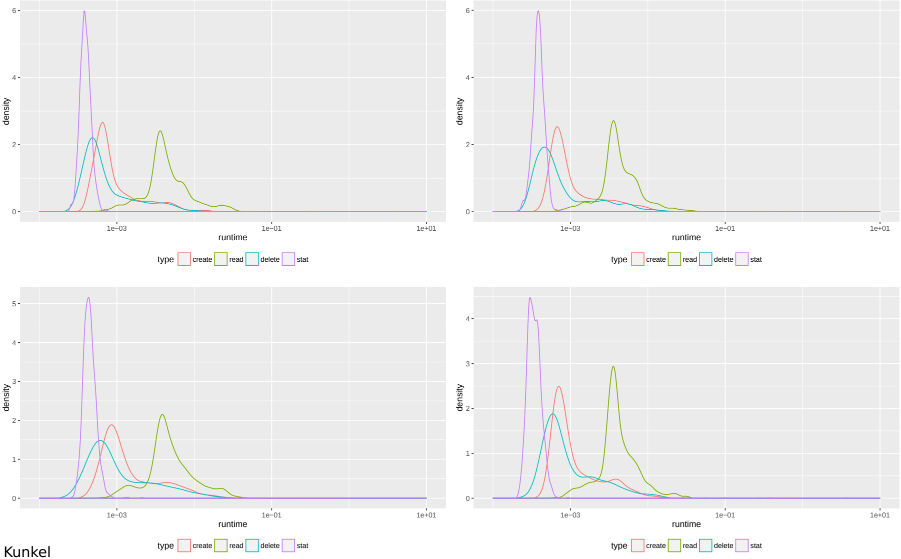




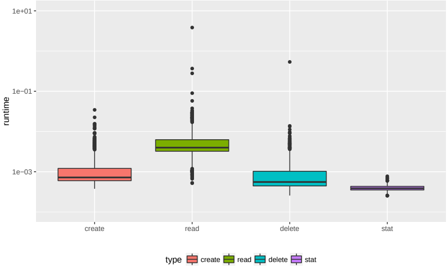


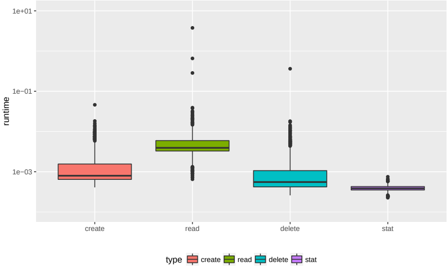


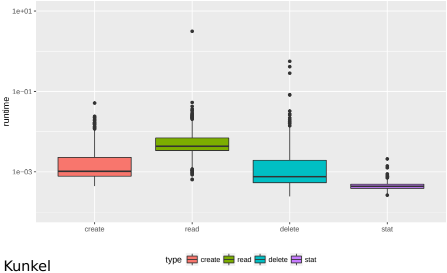
Latency Results: Benchmark Phase Histograms / Density

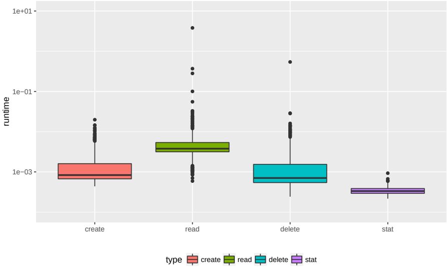


Latency Results: Benchmark Phase Boxplots

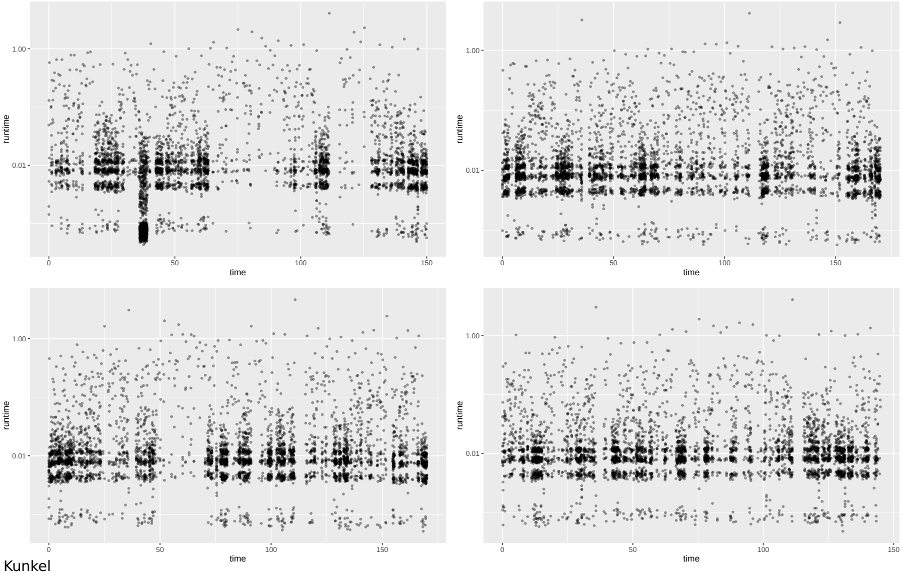




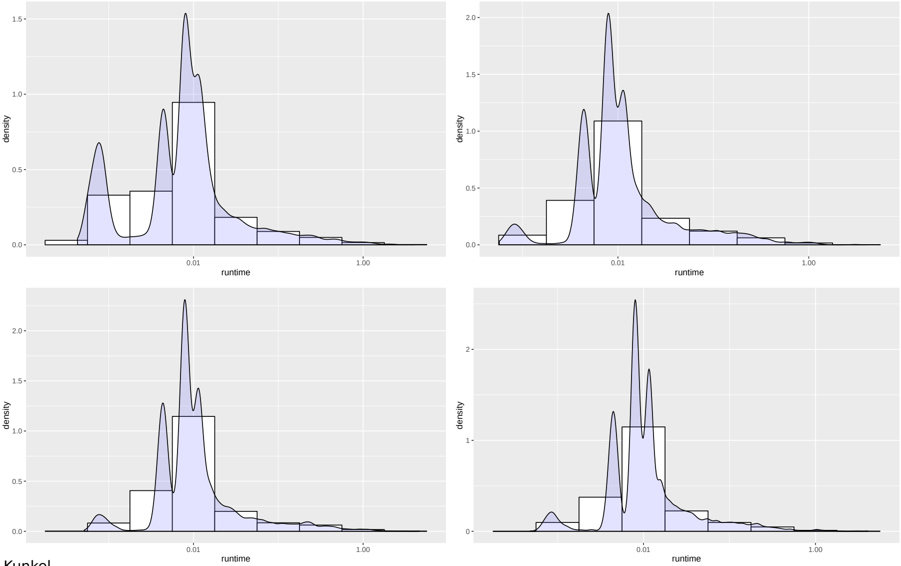




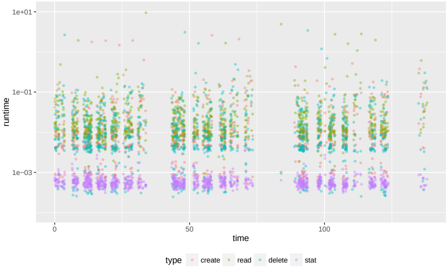
Latency Results: Mistral 100 Nodes, 10 PPN, Precreate 0-3

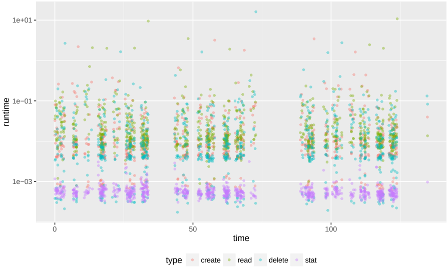


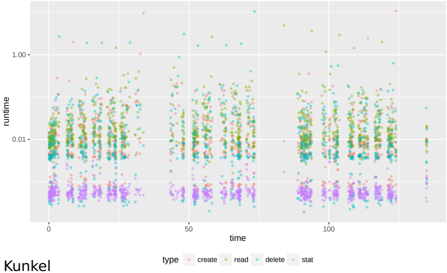
Latency Results: Precreate Histograms / Density

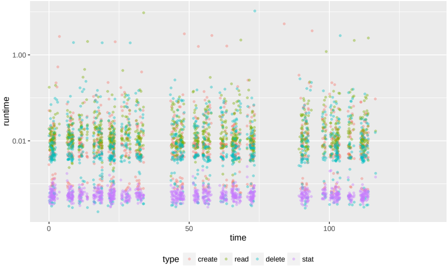


Latency Results: Benchmark Phase Timelines

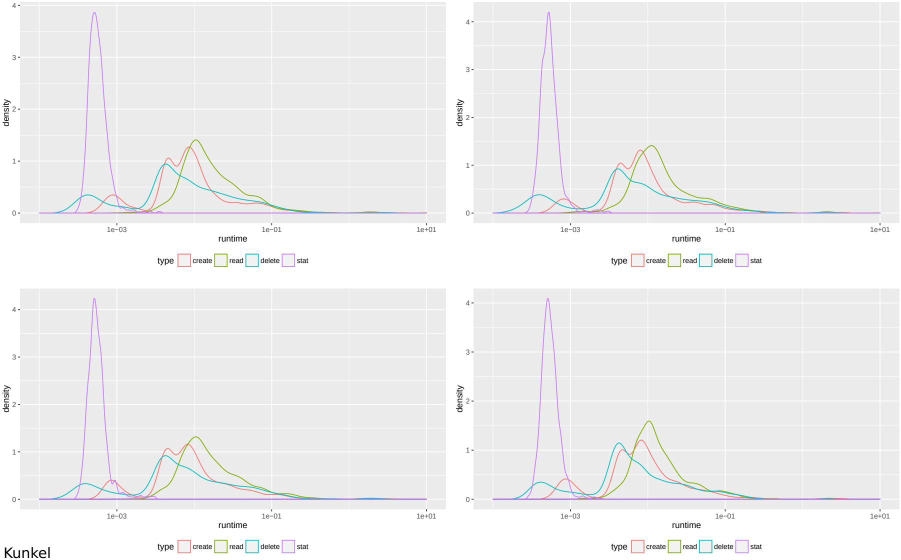




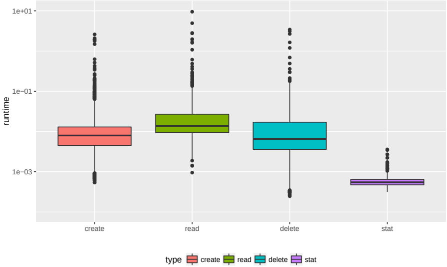


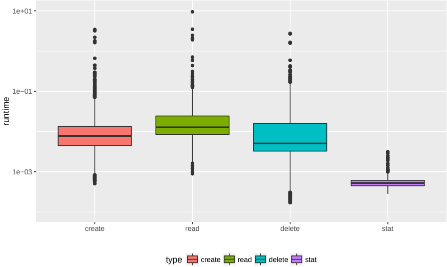


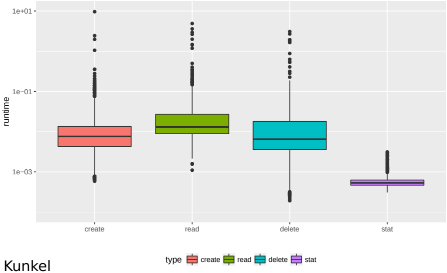
Latency Results: Benchmark Phase Histograms / Density

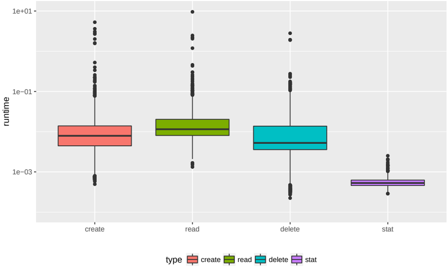


Latency Results: Benchmark Phase Boxplots









Comparing Results, 10 Nodes

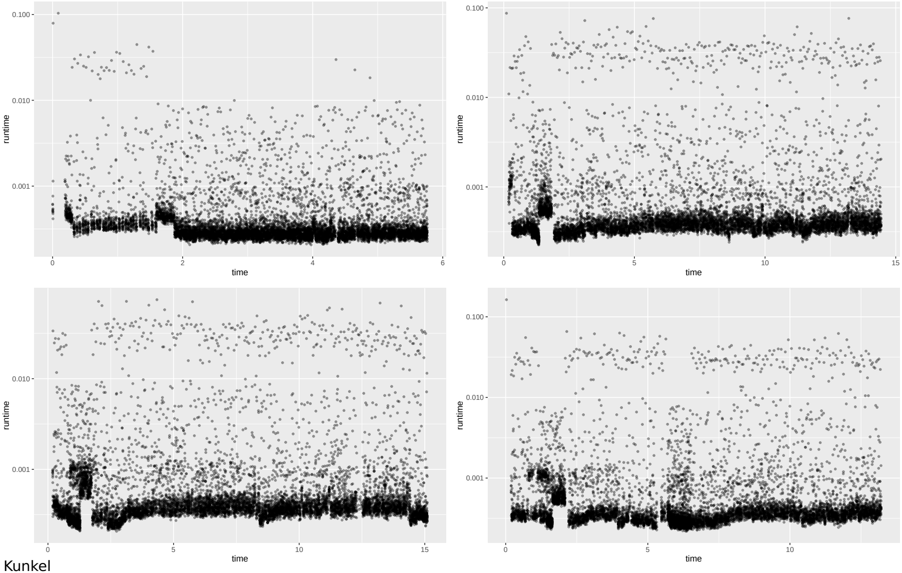
100 procs, Precreating 2000 files, accessing 1000 files

```
precreate 17.7s 11285.5 iops/s 100 dset 200000 obj 5.640 dset/s 11279.8 obj/s 42.0 Mib/s (0 errs)
benchmark 17.2s 23207.2 iops/s 100000 obj 5801.8 obj/s 43.2 Mib/s (0 errs)
benchmark 17.6s 22722.4 iops/s 100000 obj 5680.6 obj/s 42.3 Mib/s (0 errs)
benchmark 19.8s 20230.7 iops/s 100000 obj 5057.7 obj/s 37.6 Mib/s (0 errs)
cleanup 7.4s 27036.2 iops/s 200000 obj 100 dset 27022.7 obj/s 13.511 dset/s (0 errs)
```

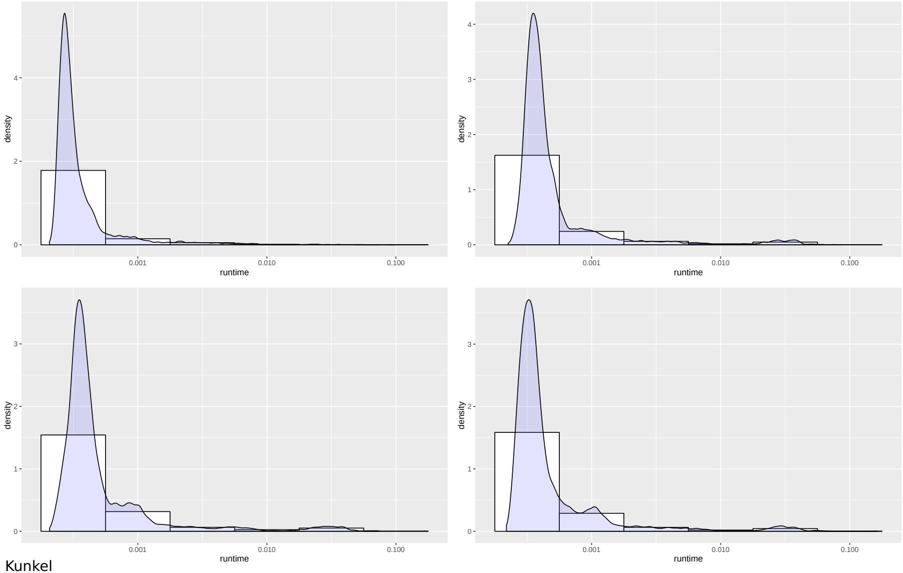
10 procs, Precreating 10000 files, accessing 10000 files

```
precreate 15.5s 6469.9 iops/s 10 dset 100000 obj 0.647 dset/s 6469.2 obj/s 24.1 Mib/s (0 errs)
benchmark 64.4s 6206.5 iops/s 100000 obj 1551.6 obj/s 11.5 Mib/s (0 errs)
benchmark 65.5s 6103.4 iops/s 100000 obj 1525.9 obj/s 11.4 Mib/s (0 errs)
benchmark 63.9s 6259.7 iops/s 100000 obj 1564.9 obj/s 11.6 Mib/s (0 errs)
cleanup 8.7s 11542.5 iops/s 100000 obj 10 dset 11541.3 obj/s 1.154 dset/s (0 errs)
```

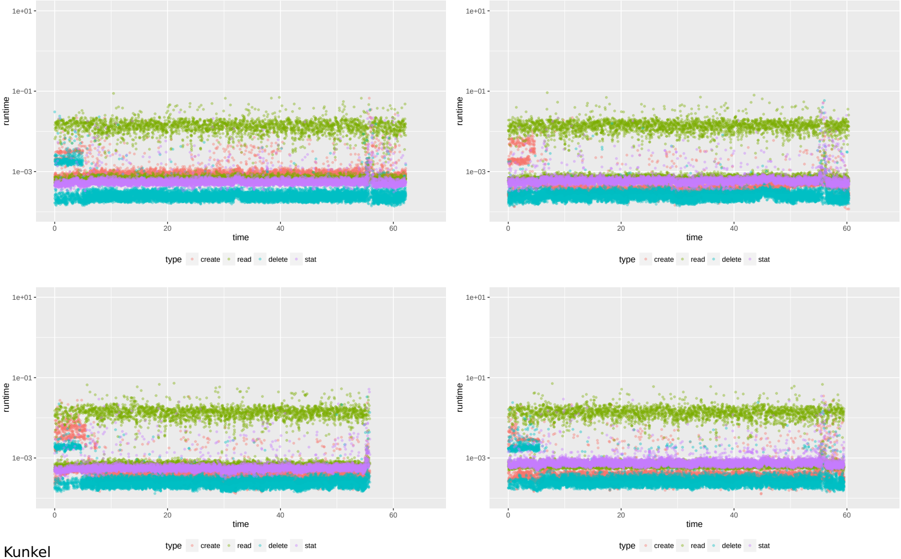
Latency Results: Cooley 10 Nodes, 1 PPN, Precreate 0-3



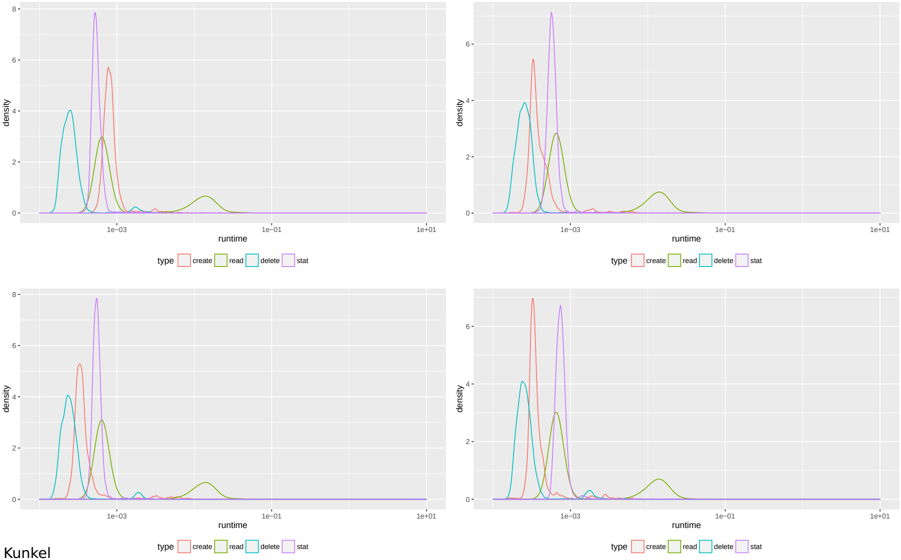
Latency Results: Precreate Histograms / Density



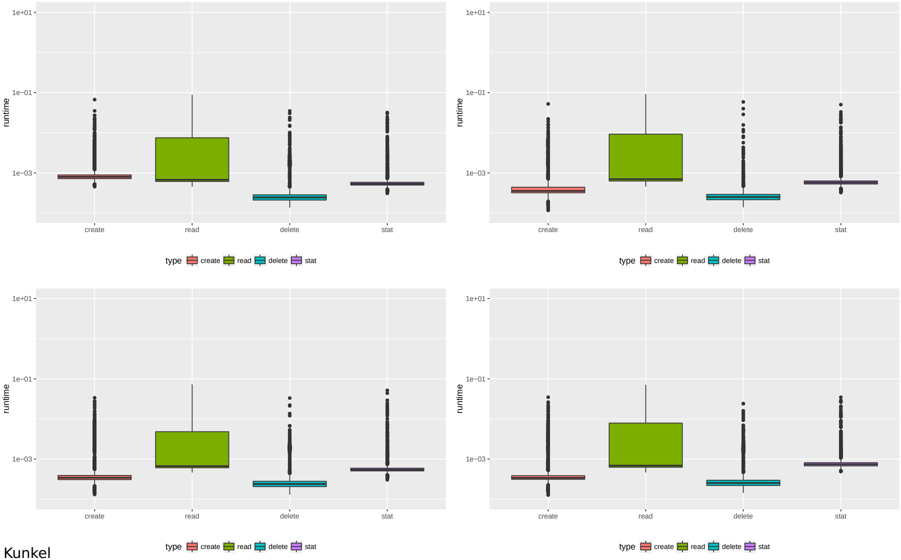
Latency Results: Benchmark Phase Timelines



Latency Results: Benchmark Phase Histograms / Density



Latency Results: Benchmark Phase Boxplots



Results

- The mixed workload shown before uses MD-REAL-IO
- Realistic working set: runtime on Mistral 12 minutes
- Creating a working set can take more time but a small set yields nearly same performance results
- The working set is 3,000,000 objects, 11 GiB
- Performance on our last-generation Blizzard supercomputer: 250 objects/s (x 8 ops/iteration)
- Mistral using a single metadata server (we have 5+7 servers)
  - Phase 1 (in production): 1200 iter/s, 9 MiB/s
  - Phase 2 (nearly empty): 7000 iter/s, 53 MiB/s
- Earth-Simulator: 1880 iter/s, 14 MiB/s