Perf Tools - PerfLine

Tushar Jain, CORTX Performance

Feb 10, 2021

AGENDA

01 Introduction

02 Configurations

03 ADDB Data Processing

04 Future Enhancements

What is PerfLine?

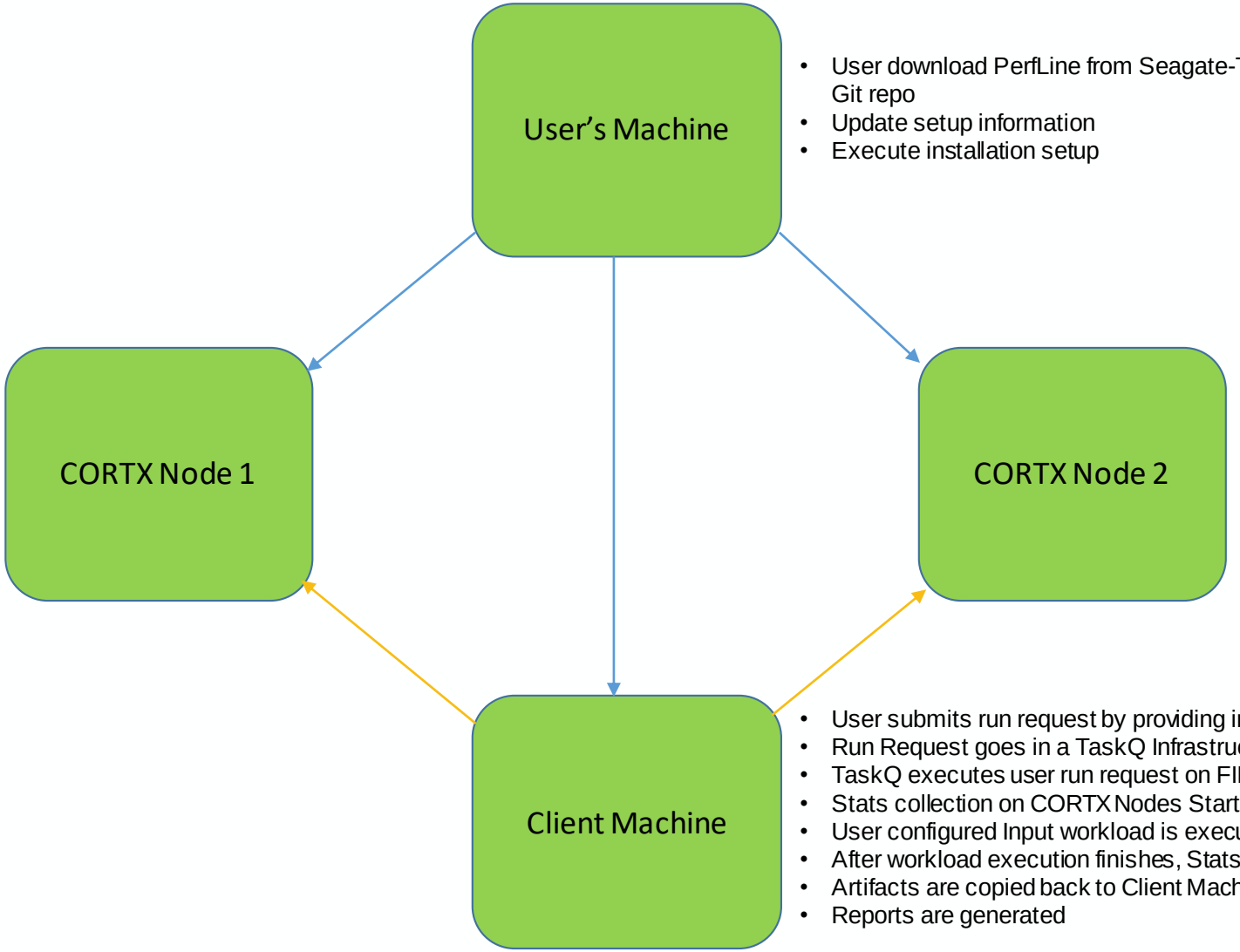
- PerfLine is a CORTX Profiling and Performance Analysis Tool

- **Can do**
  - Run Benchmark / Microbenchmark workloads on CortX Setup – S3Bench, m0crate
  - Make use of ADDB(Analytic and Diagnostic Data-Base) framework
  - Capture state of System, CortX, Storage
  - Process captured state and provide comprehensive data and report to user
- **Can't do**
  - Provision CORTX Infrastructure
  - Post Process Benchmark Report and Summary

Why PerfLine?

- CORTX is an object store software
  - Handles IO Requests Asynchronously
  - Difficult to relate various internal requests/calls generating as result of a single client request
- 3rd Party tools fail to capture this relation
  - rely on synchronous behavior
  - Focused on aggregated system resource utilization rather than per unit measurements

PerfLine Setup



- User download PerfLine from Seagate-Tools Git repo
- Update setup information
- Execute installation setup

- Nodes Handle Workload Requests
- Capture and Process ADDB Stats and other Artifacts

- User submits run request by providing inputs
- Run Request goes in a TaskQ Infrastructure
- TaskQ executes user run request on FIFO basis
- Stats collection on CORTX Nodes Starts
- User configured Input workload is executed
- After workload execution finishes, Stats collection is stopped
- Artifacts are copied back to Client Machine
- Reports are generated

Setup Summary

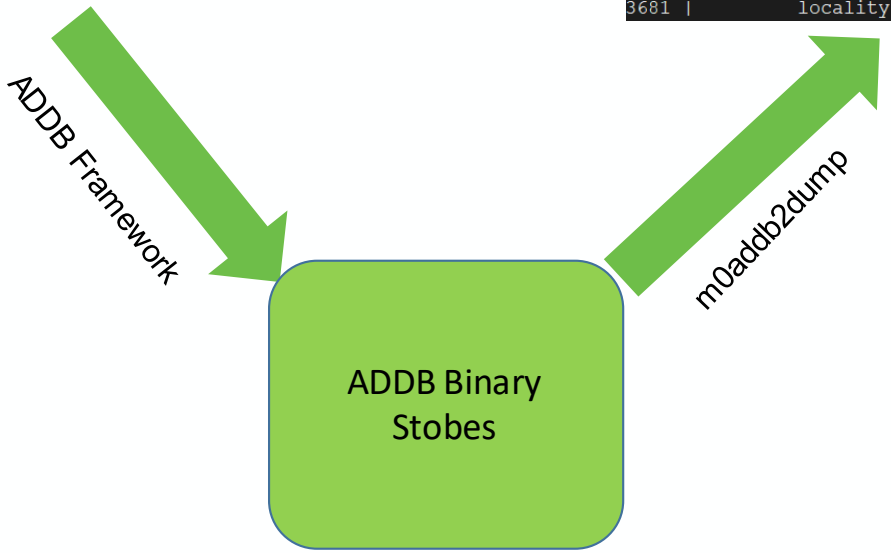
```
[2021-02-10 02:37:27] system | -- Play recap --
localhost                  : ok=19    changed=15    unreachable=0     failed=0
Wednesday 10 February 2021  02:37:27 -0700 (0:00:07.704)       0:10:26.347 ****
===============================================================================
perfline_setup : Install perfline system dependencies ------------------------------ 264.77s
perfline_setup : Install perfline python dependencies ------------------------------- 59.80s
perfline_setup : Copy perfline chronometry_v2 scripts ------------------------------- 47.67s
perfline_setup : Copy perfline chronometry scripts ---------------------------------- 47.09s
perfline_setup : Copy perfline wrapper scripts -------------------------------------- 46.93s
perfline_setup : Creating artifacts directory /var/perfline ------------------------- 34.18s
perfline_setup : Copying local.repo in "/etc/yum.repo.d/" location ------------------ 31.71s
perfline_setup : Creating perfline directory /root/perfline ------------------------- 24.27s
perfline_setup : Copy perfline webui scripts ---------------------------------------- 14.92s
perfline_setup : Copy webui systemd file -------------------------------------------- 11.82s
perfline_setup : Copy perfline systemd file ----------------------------------------- 11.28s
perfline_setup : Copy huey_consumer python bin -------------------------------------- 11.28s
perfline_setup : Start webui service ------------------------------------------------- 8.04s
perfline_setup : Start perfline service ---------------------------------------------- 7.70s
Gathering Facts ---------------------------------------------------------------------- 3.38s
perfline_setup : Compress directory files/wrapper into files/wrapper.tar.gz ---------- 0.50s
perfline_setup : Compress directory files/chronometry into files/chronometry.tar.gz -- 0.34s
perfline_setup : Compress directory files/webui into files/webui.tar.gz -------------- 0.31s
perfline_setup : Compress directory files/chronometry_v2 into files/chronometry_v2.tar.gz -- 0.30s
```

Run Submission Configuration

```
common:
  version: 1
  description: Perf benchmark - s3bench, size=256Kb, clients=10, num=100
  priority: 1
  batch_id: 'demo_run'
  user: tushar.1.jain@seagate.com
  send_email: true

workload:
  - cmd: sleep 1

benchmark:
  fio: false
  s3bench: True

parameter:
  BucketName: demobucket
  NumClients: 10
  NumSample: 100
  ObjSize: 256Kb

execution_options:
  mkfs: false
  no_m0trace_files: true
  no_m0trace_dumps: true
  no_addb_stobs: false
  no_addb_dumps: false
  no_m0play_db: false
~
~
```

ADDB Data Collection

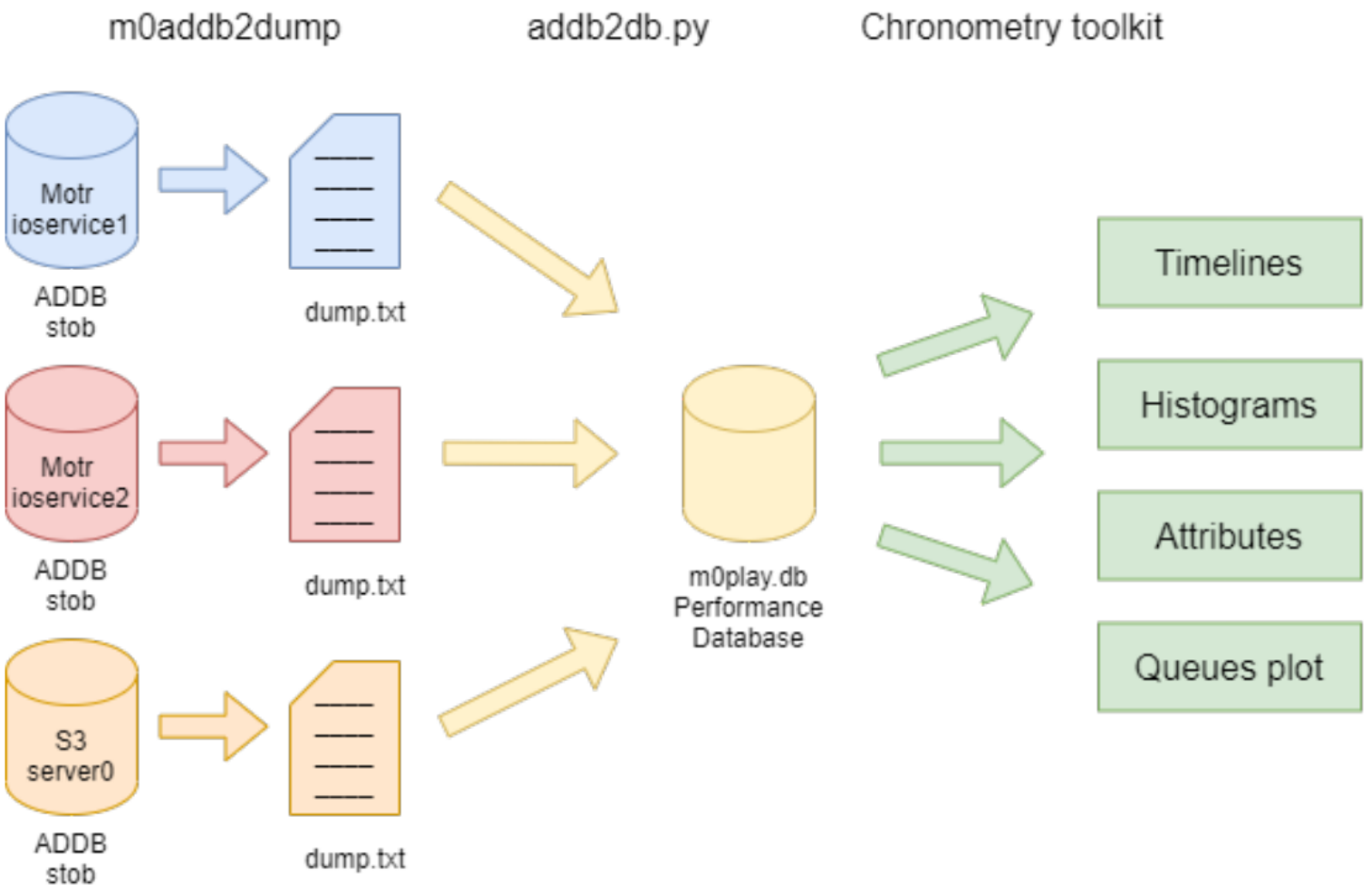
```
static void to_op_io_map(const struct m0_op *op,
                         struct m0_op_io *ioo)
{
        uint64_t oid  = m0_sm_id_get(&op->op_sm);
        uint64_t ioid = m0_sm_id_get(&ioo->ioo_sm);

        if (ioo->ioo_addb2_mapped++ == 0)
                M0_ADDB2_ADD(M0_AVI_CLIENT_TO_IOO, oid, ioid);
}
```




```
* 2021-02-04-06:50:38.277542944 client-to-ioo    client_id: 5571, ioo_id: 5572 |
    233681 |         locality        82 |        thread          7f6adf7fe700
* 2021-02-04-06:50:38.307873427 rpc-bulk-op      bulk_id: 5871, state: M0_RPC_BUL
   pid             233681 |         locality        82 |        thread
* 2021-02-04-06:50:38.308946211 attr             entity_id: 5871, M0_AVI_RPC_BULK
f41b80> |         pid             233681 |        locality        82 |
* 2021-02-04-06:50:38.308949283 attr             entity_id: 5871, M0_AVI_RPC_BULK
     pid             233681 |         locality        82 |        thread
* 2021-02-04-06:50:38.308950582 attr             entity_id: 5871, M0_AVI_RPC_BULK
 |         pid             233681 |         locality        82 |        thread
* 2021-02-04-06:50:38.308952011 attr             entity_id: 5871, M0_AVI_RPC_BULK
       pid             233681 |         locality        82 |        thread
* 2021-02-04-06:50:38.309652909 rpc-out-phase    sm_id: 5934 --> INITIALISED |
  233681 |         locality        82 |        thread          7f6adf7fe700 |
* 2021-02-04-06:50:38.309714531 rpc-out-phase    sm_id: 5934 --> URGENT |
3681 |         locality        82 |        thread          7f6adf7fe700 |
```

Credits : Maxim Malezhin

ADDB Data Collection



Credits : Maxim Malezhin

ADDB Data Demo

Enhancements Planned

- Per Task build and deploy based on git commit
- Restore Queue Length Plots
- Generate Aggregated Report Dashboard
- Restore and Update WebUI
- Add more benchmark workloads – Cosbench, Veeam

Talk to us!

- PerfLine Location
  - Github Repository : Seagate/Seagate-tools
  - Demo Recording and these slides : Available Here

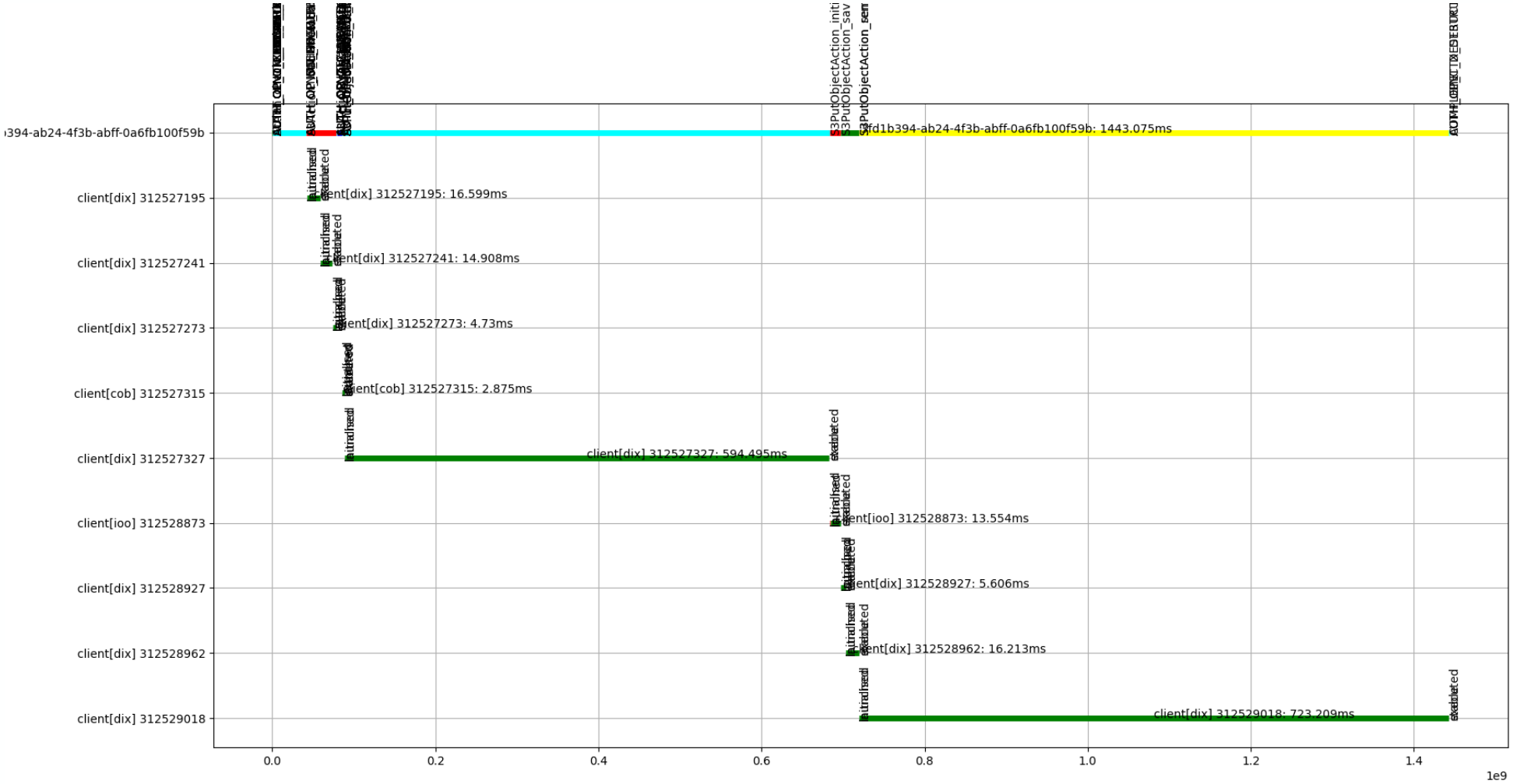
- Have query/suggestion/idea for PerfTools?
  - Email : cortx.perf@seagate.com
  - Jira : project = EOS AND component = Cortx-Perf

Q & A

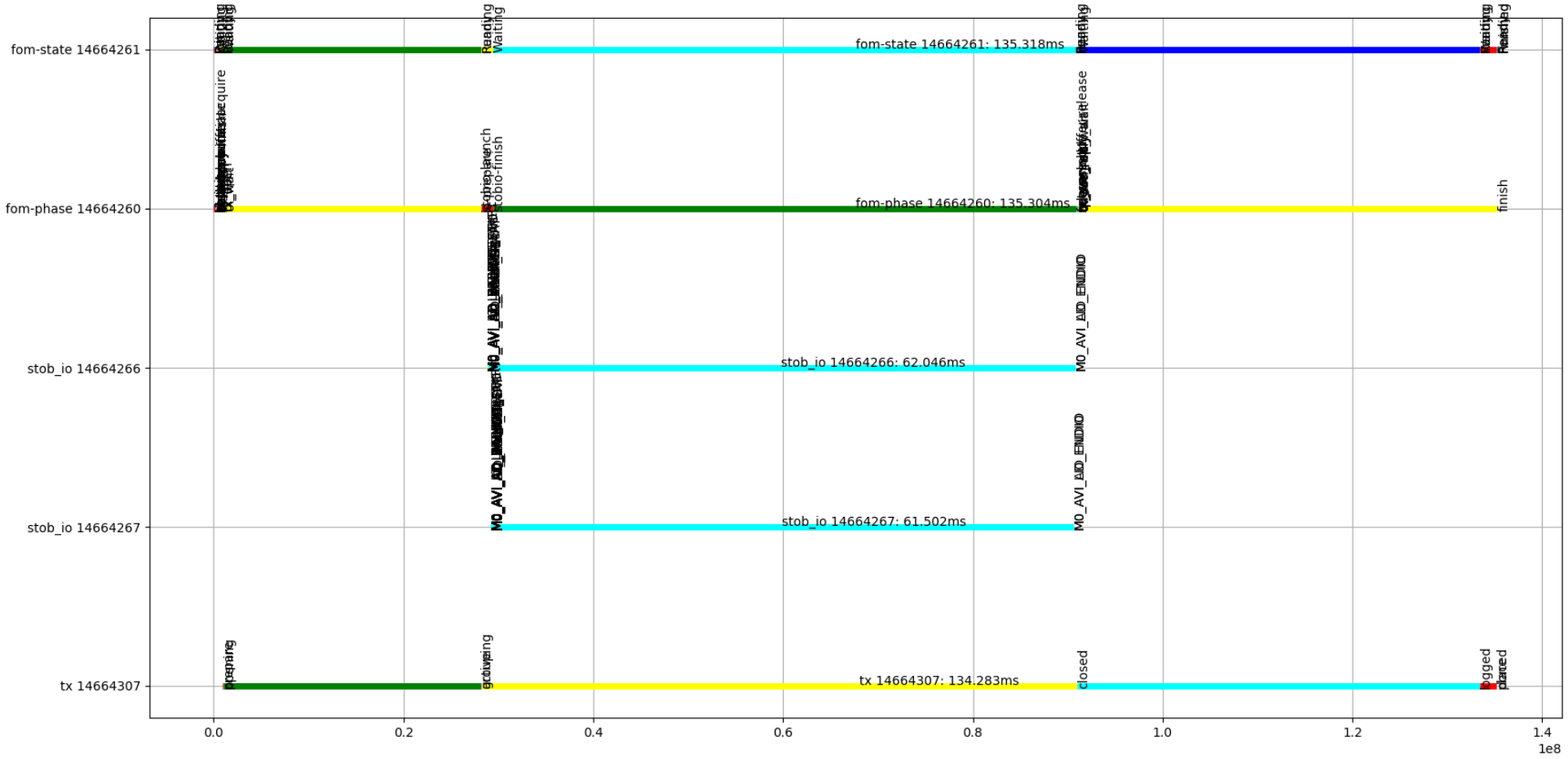
Backup Slides

Sequence diagrams / Timelines

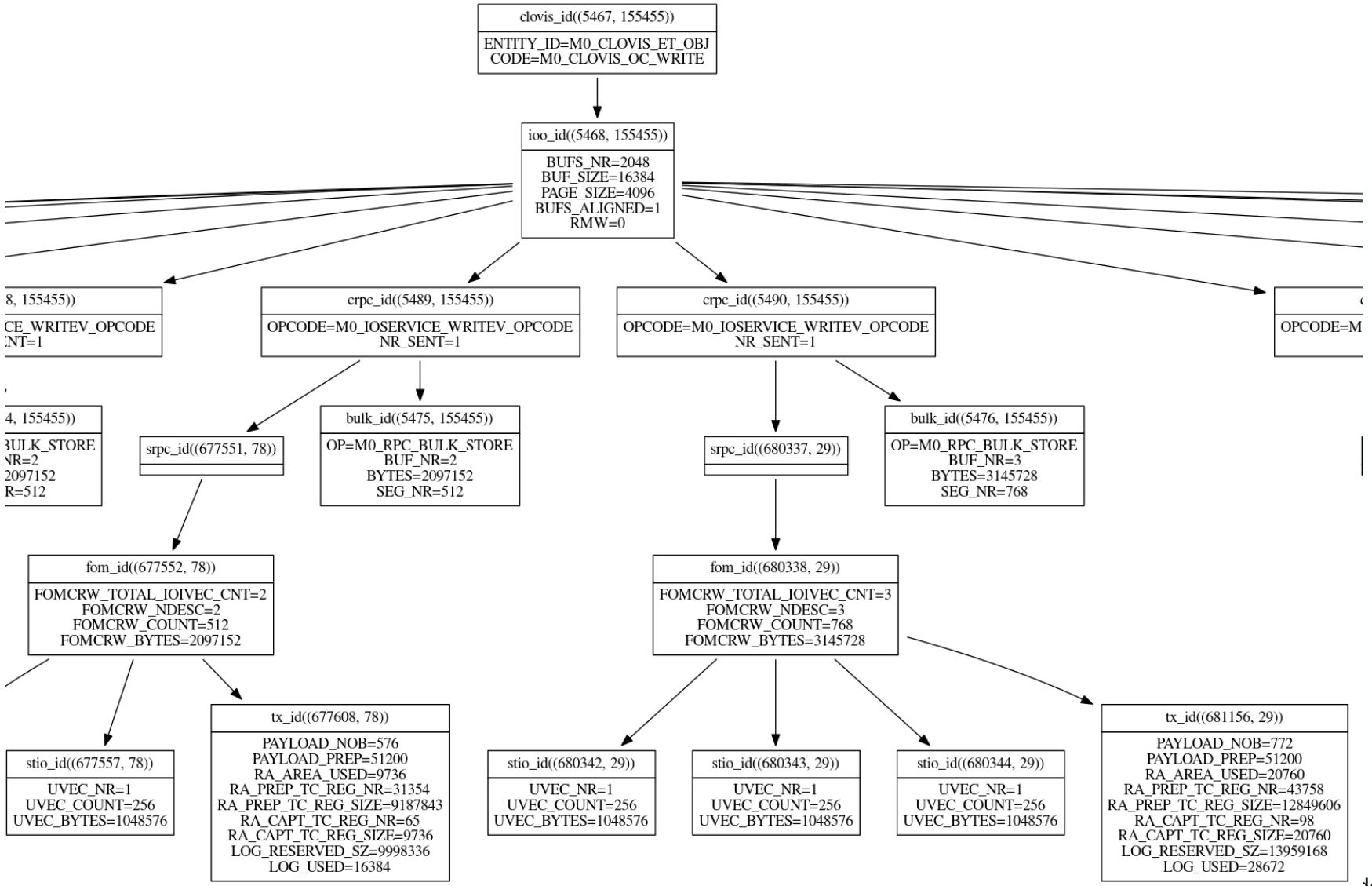
Timeline shows system evolution and time spent on each phase for a single request



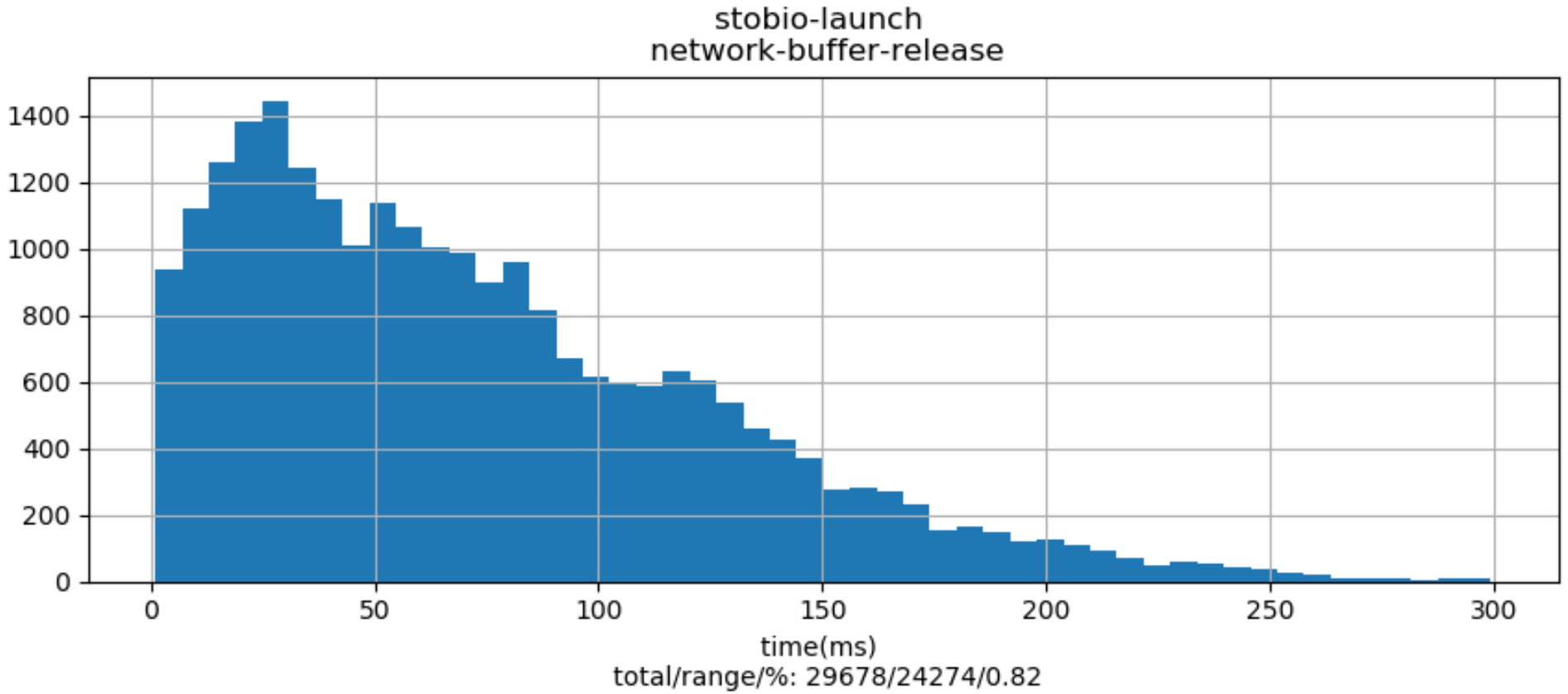
Sequence diagrams / Timelines (2)



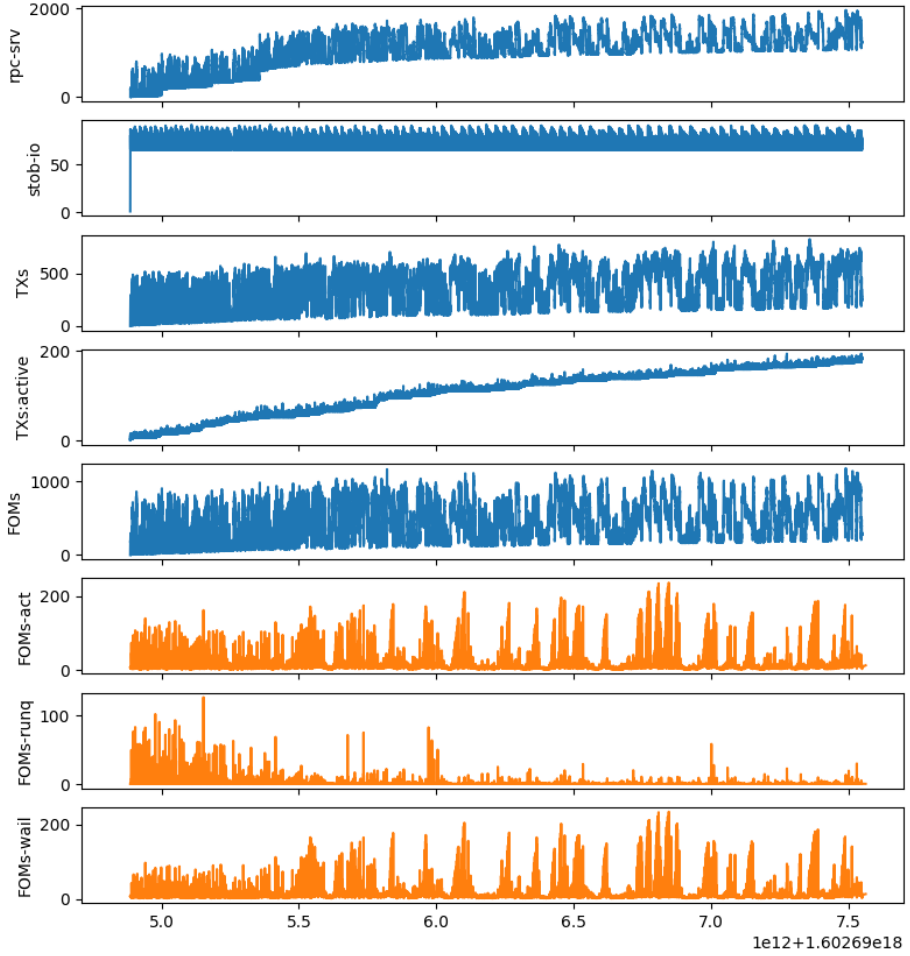
Graph of attributes

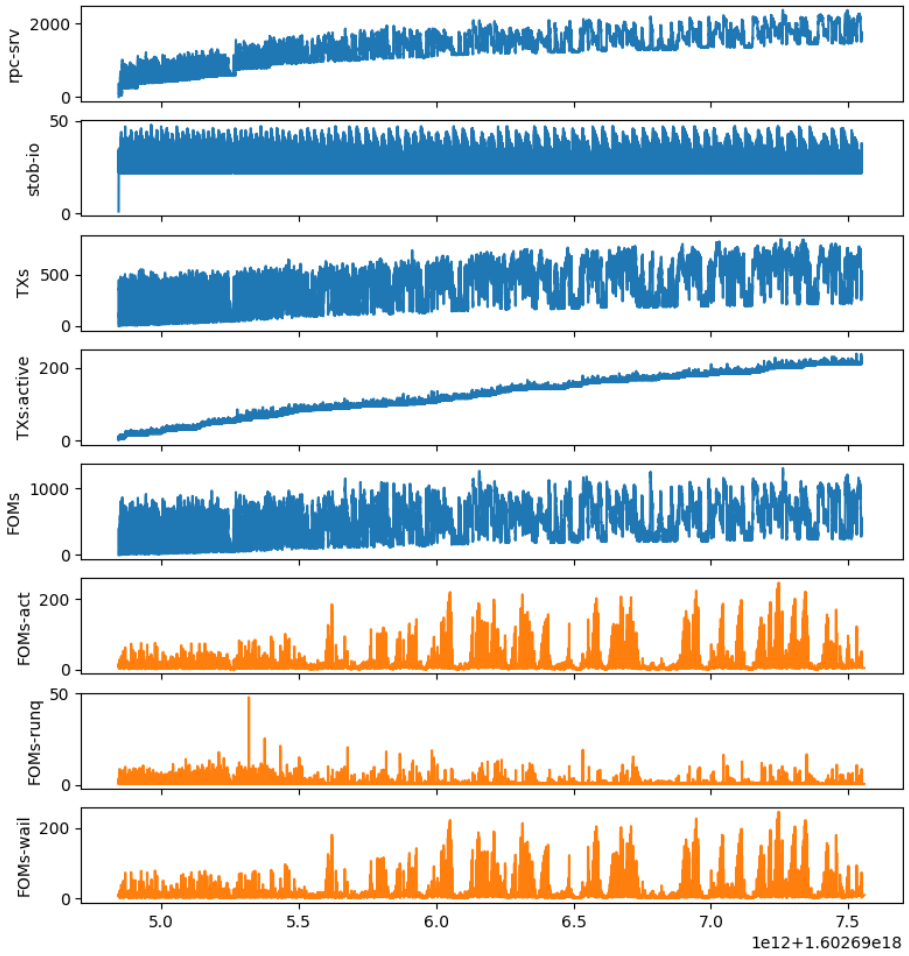


Histograms (2)



Queue length plot





Run Submission Configuration

```
common:
  version: 1
  description: Perf benchmark - s3bench, size=256Kb, clients=10, num=100
  priority: 1
  batch_id: 'demo_run'
  user: tushar.1.jain@seagate.com
  send_email: true

workload:
  - cmd: sleep 1

benchmark:
  fio: false
  s3bench: True

parameter:
  BucketName: demobucket
  NumClients: 10
  NumSample: 100
  ObjSize: 256Kb

execution_options:
  mkfs: false
  no_m0trace_files: true
  no_m0trace_dumps: true
  no_addb_stobs: false
  no_addb_dumps: false
  no_m0play_db: false
~
~
```