SDC

STORAGE DEVELOPER CONFERENCE

SNIA ■ SANTA CLARA, 2016

Performance Testing Ceph with CBT

Logan Blyth
Aquari
Logan.blyth@concurrent.com

Agenda

- ❒ Overview of Ceph I/O path
- ❒ Motivation Behind CBT
- ❒ Which Benchmarks can it run
- ❒ CBT Setup
- ❒ Running CBT
- ❒ CBT Results

CONCURRENT

MEDIA & TELECOMMUNICATIONS

Time Warner Cable

COX

Charter COMMUNICATIONS

vodafone

LIBERTY GLOBAL

J:COM

AUTOMOTIVE & TRANSPORTATION

Ford

DAIMLER

dallara

GE Transportation

AEROSPACE & DEFENSE

BOEING

AIRBUS

LOCKHEED MARTIN

U.S. AIR FORCE

NASA

MANUFACTURING & ENERGY

TOSHIBA

MITSUBISHI HEAVY INDUSTRIES, LTD.

CLEMSON UNIVERSITY SCE&G ENERGY INNOVATION CENTER

SIEMENS

NASDAQ: CCUR

We are a Global Software Company
Our Heritage is in Mission-Critical Solutions

~260 Employees Worldwide
Headquartered in Atlanta, GA

Why Aquari?

- ✓ **Flexibility**
  - ✓ Multiple Workload Types
  - ✓ Object, File & Block
  - ✓ Scalable to Exabytes
- ✓ **Manageability**
  - ✓ Ease of Installation
  - ✓ Ease of Operation
  - ✓ Ease of Expansion
- ✓ **Expertise**
  - ✓ Video and Simulations
- ✓ **Global Support**
  - ✓ NA, EMEA, APAC

RCN *“Aquari is a huge step forward for RCN.”*

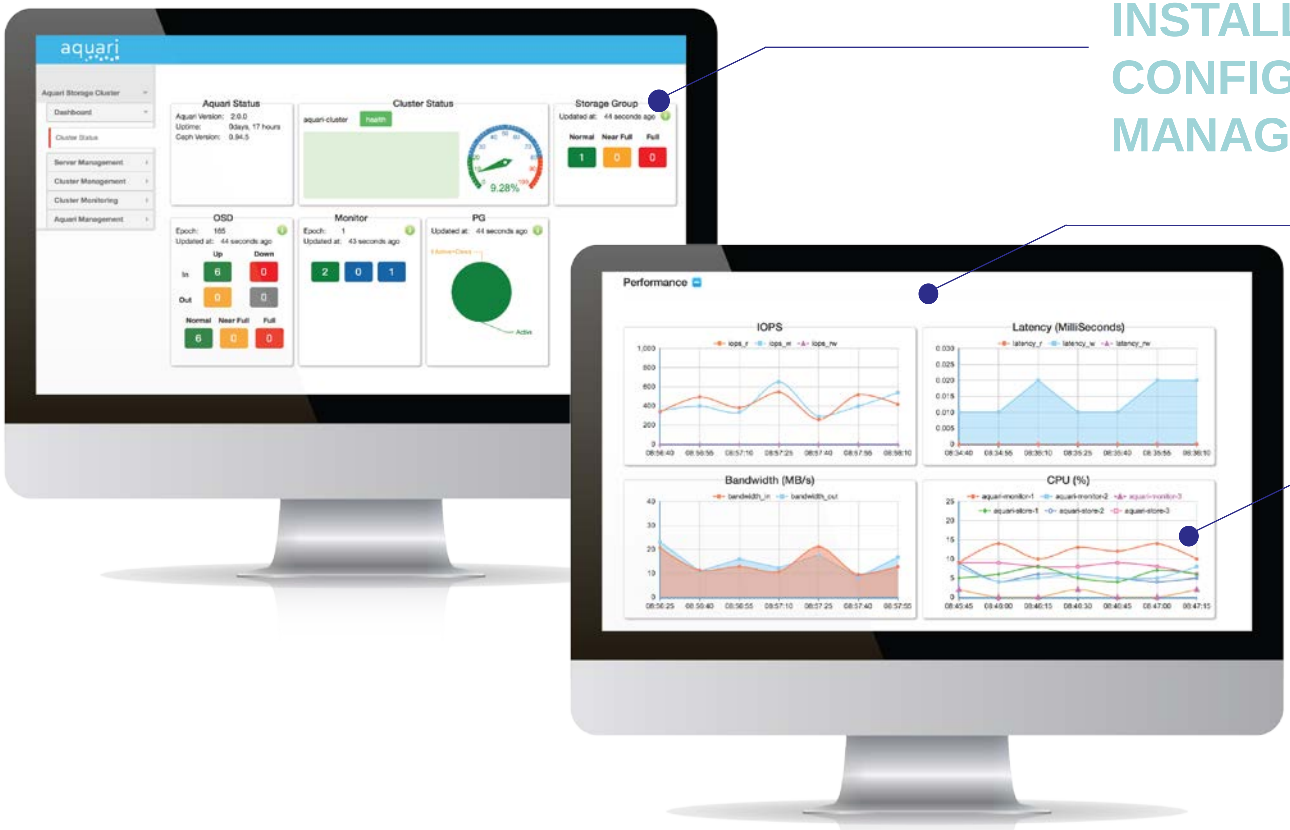
COX *“We are bought into the vision of where you are heading with Aquari.”*

VERISTOR *“You guys are 3x faster than SwiftStack.”*

redhat *“No one is doing Ceph Management like you guys.”*

MIRANTIS *“We deploy Ceph, but what you are doing is goes beyond what we do.”*

Aquari Storage OS UI



Aquari Version: 2.0.0
Uptime: 0days, 17 hours
Ceph Version: 0.94.5

Storage Group — Updated at: 44 seconds ago

OSD — Epoch: 165 — Updated at: 44 seconds ago

Monitor — Epoch: 1 — Updated at: 43 seconds ago

PG — Updated at: 44 seconds ago

INSTALL, CONFIGURE & MANAGE

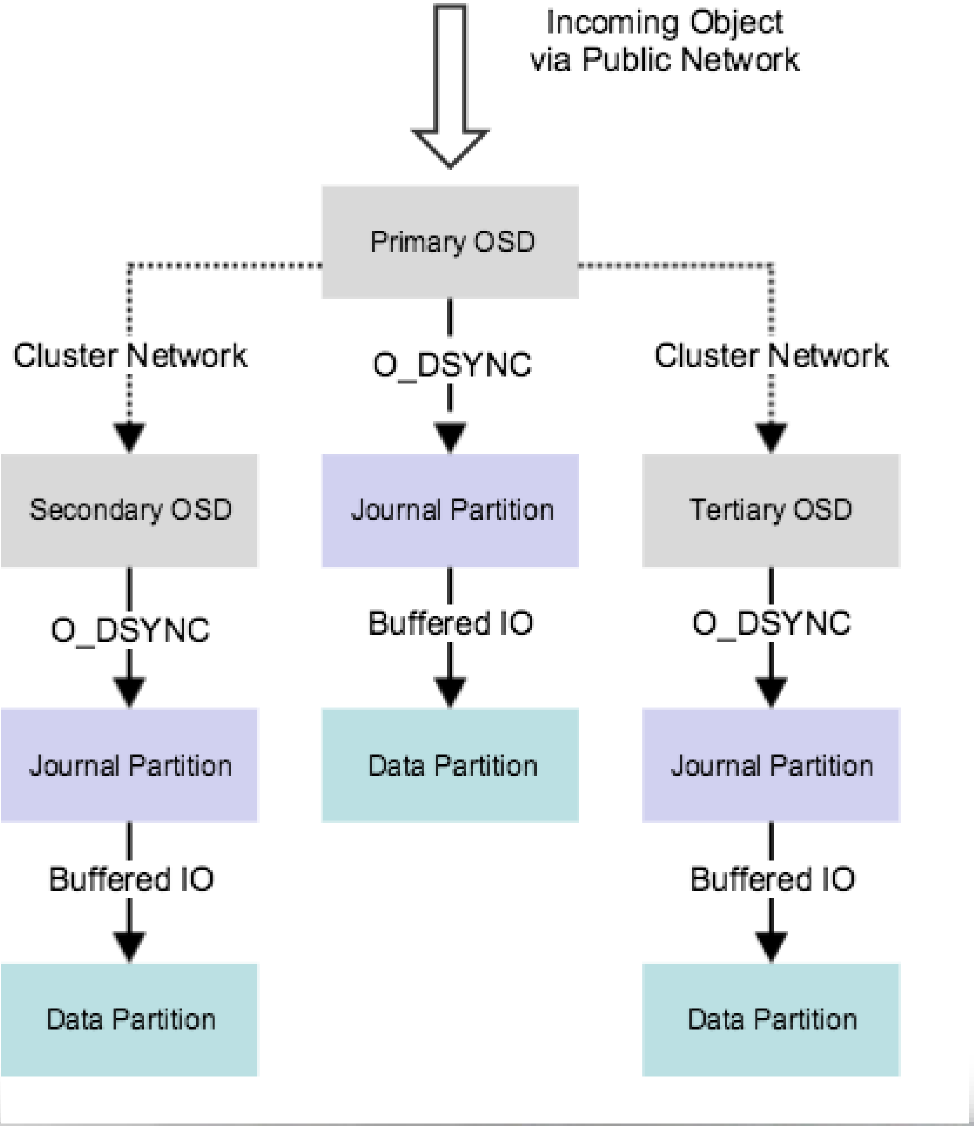
REAL-TIME PERFORMANCE INFORMATION

HTML5 GUI REST API BACKEND

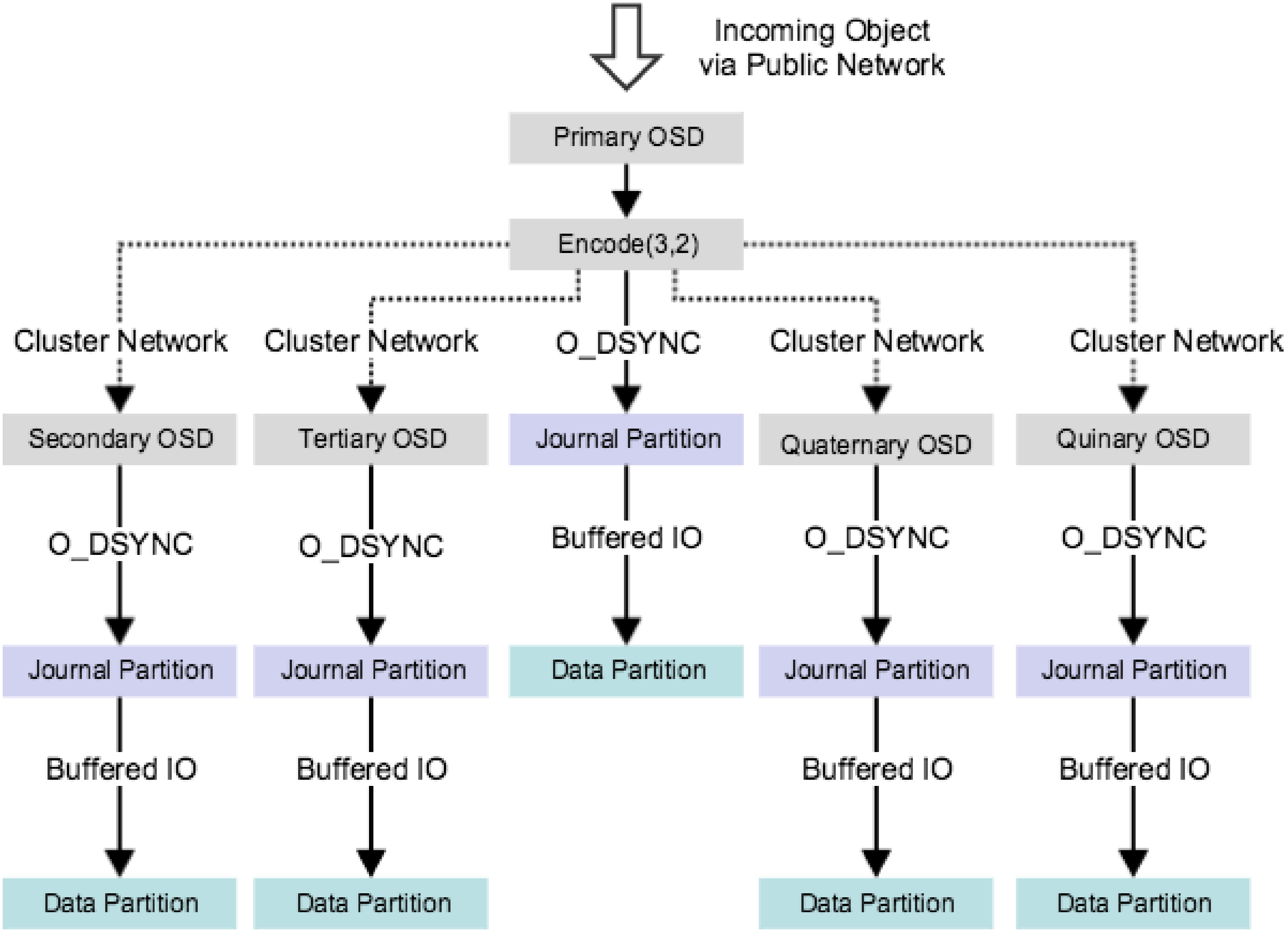
Quick Ceph IO Path Overview

Thanks unsplash

Ceph v0.94 Replicated Pool I/O Path



Ceph v.94 Erasure Code Pool I/O Path



CBT

unsplash

Motivation behind CBT

- Ceph Benchmarking Tool
  - Originally developed for Ceph regression testing – Valgrind integration
  - Now also used for cluster benchmark / comparison
  - Teuthology – Ceph Nightly testing and community lab
  - Useful for recovery / backfill testing

Why Use CBT?

- File based testing setup
  - Able to sweep through array of parameters in yml file
  - Built in metric collection with collectl
  - Able to rebuild a cluster
  - Able to supply different ceph.conf files
  - Option to run the same test multiple times for larger sample size
- Used by Industry
  - Inktank / Redhat
  - Intel
  - Concurrent
  - Cisco

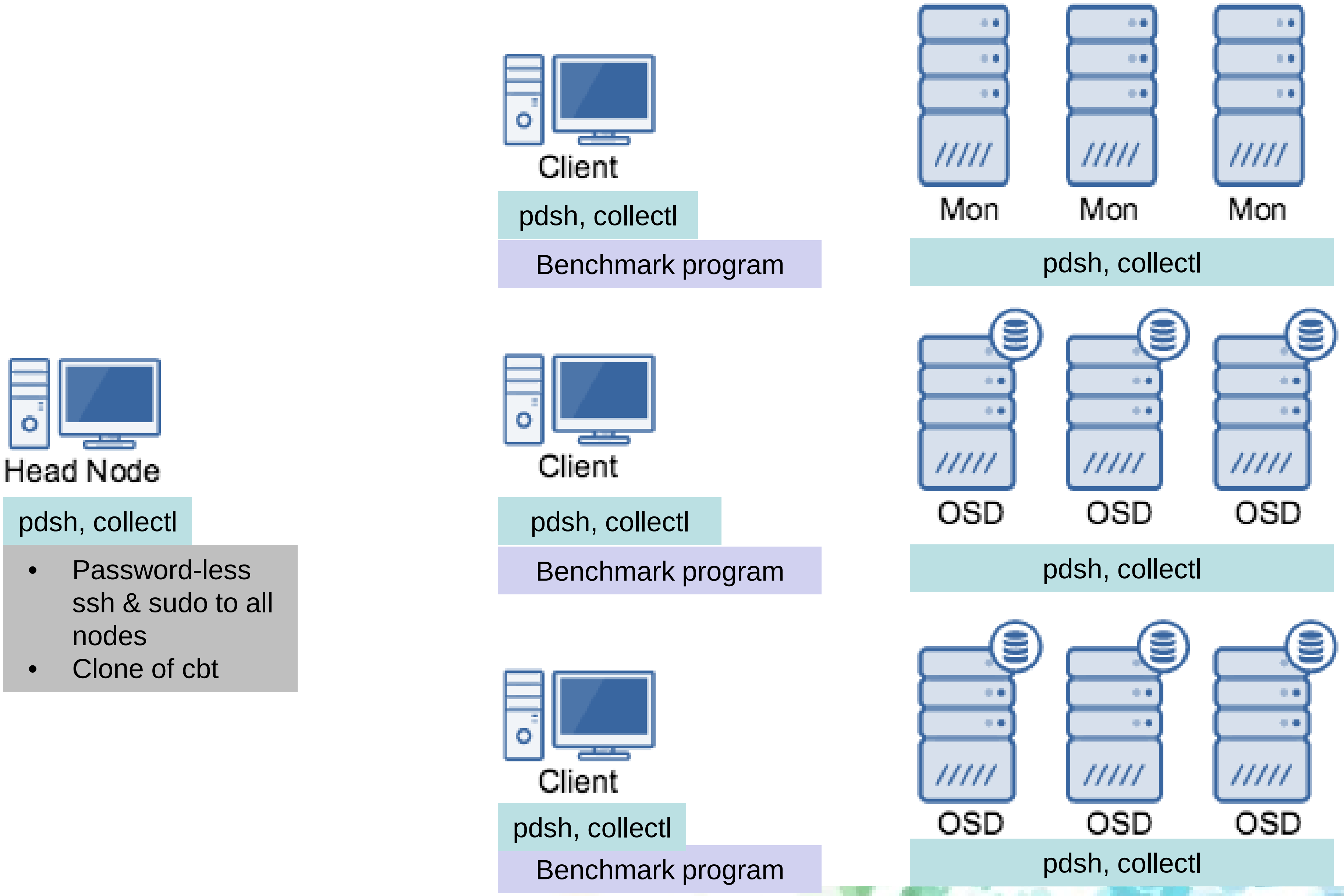
CBT Benchmarks

- Testing Harness around:
  - CosBench – cloud object storage, for S3/ Swift
  - Kvmrbdfio – RBD vol attached to KVM instance
  - Librbdfio – userspace librbd ioengine
  - Rbdfio – uses kernel rbd driver, /dev/rbd0
  - Rados bench – object based, asynchronous
  - ceph_test_rados – used by Redhat to stress test rados

CBT Setup

Thanks Again Unsplash

CBT Setup – Installation and Configuration



pdsh, collectl

- Password-less ssh & sudo to all nodes
- Clone of cbt

pdsh, collectl

Benchmark program

pdsh, collectl

Benchmark program

pdsh, collectl

Benchmark program

pdsh, collectl

pdsh, collectl

pdsh, collectl

CBT – Yml file – Cluster

```
---
cluster:
  user: 'aquari'
  head: "mon-01"
  clients: ["client1","client2","client3"]
  osds: ["data-01","data-02","data-03","data-04","data-05","data-06"]
  mons: ["mon-01","mon-02","mon-03"]
  osds_per_node: 10
  fs: 'xfs'
  mkfs_opts: '-f -i size=2048 -n size=8k'
  mount_opts: 'noatime,nodiratime,attr2,logbufs=8,logbsize=256k,largeio,inode64,swalloc'
  conf_file: '/etc/ceph/ceph.conf'
  iterations: 1
  use_existing: True
  rebuild_every_test: True
  clusterid: ”cluster_name"
  tmp_dir: "/tmp/cbt"
```

CBT – Yml file– Pool Profiles

```
pool_profiles:
  rbd3rep:
   pg_size: 4096
   pgp_size: 4096
   replication: 3
  erasure4_2:
   pg_size: 4096
   pgp_size: 4096
   replication: 'erasure'
   erasure_profile: 'ec42'
 erasure_profiles:
  ec42:
   erasure_k: 4
   erasure_m: 2
  ec32:
   erasure_k: 3
   erasure_m: 2
```

CBT – Yml file– Benchmark

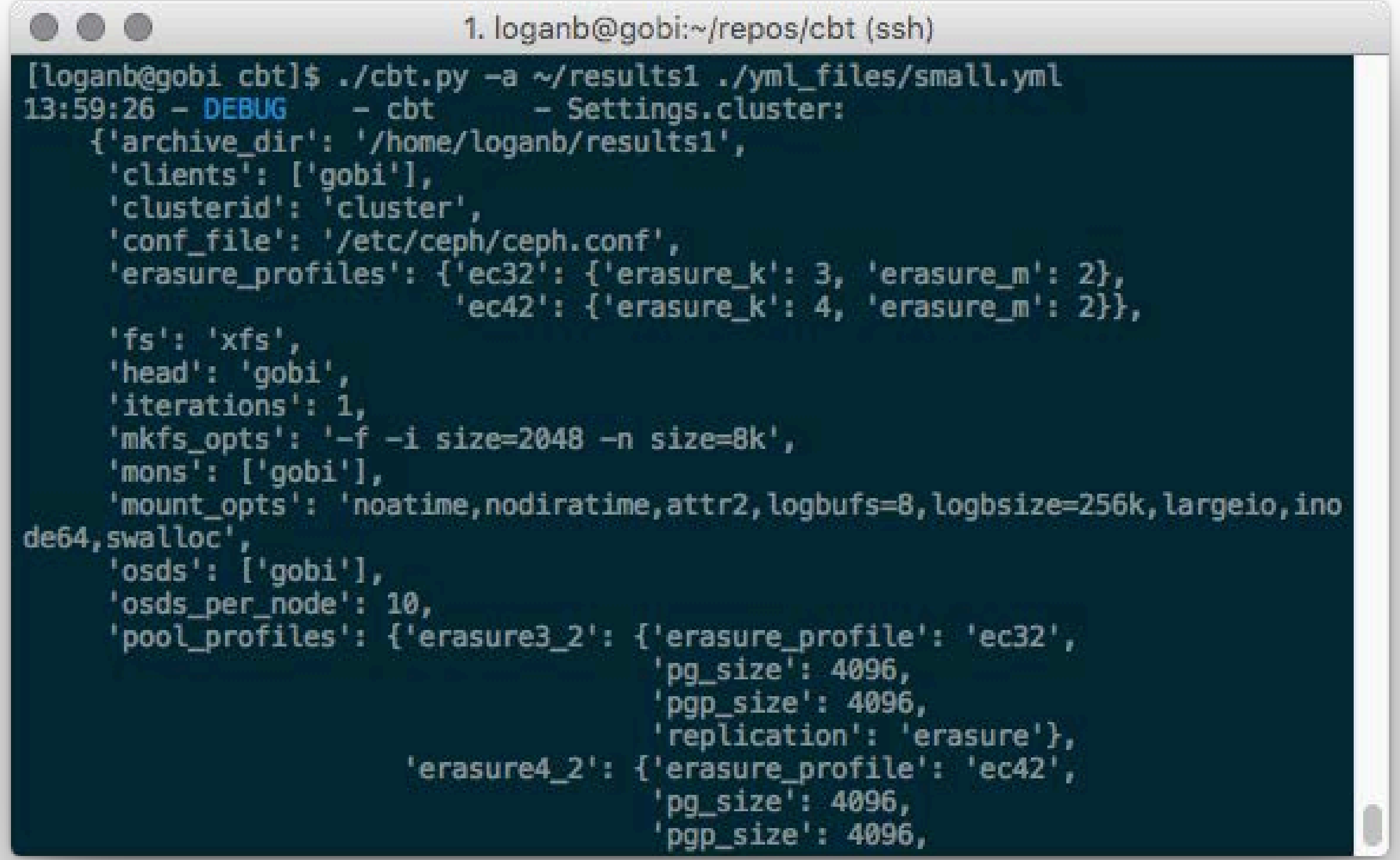
```
benchmarks:
  radosbench:
    time: 600 #seconds
    write_only: False
    readmode: ‘seq’
    pool_per_proc: False
    #Object size
    op_size: [4194304,1048576]
    # Number of rados bench processes generating concurrent_ops
    concurrent_procs: 1
    # Number of outstanding IO that rados bench keeps open
    concurrent_ops: 64
    osd_ra: [0]
    pool_profile: ['erasure4_2','rbd2rep','rbd3rep']
```

CBT – Before you Run

- Make sure the collectl invocation is to your liking in monitoring.py
- Check disk space on head node
  - Output can be 200-500MiB depending on collectl settings
- Command:

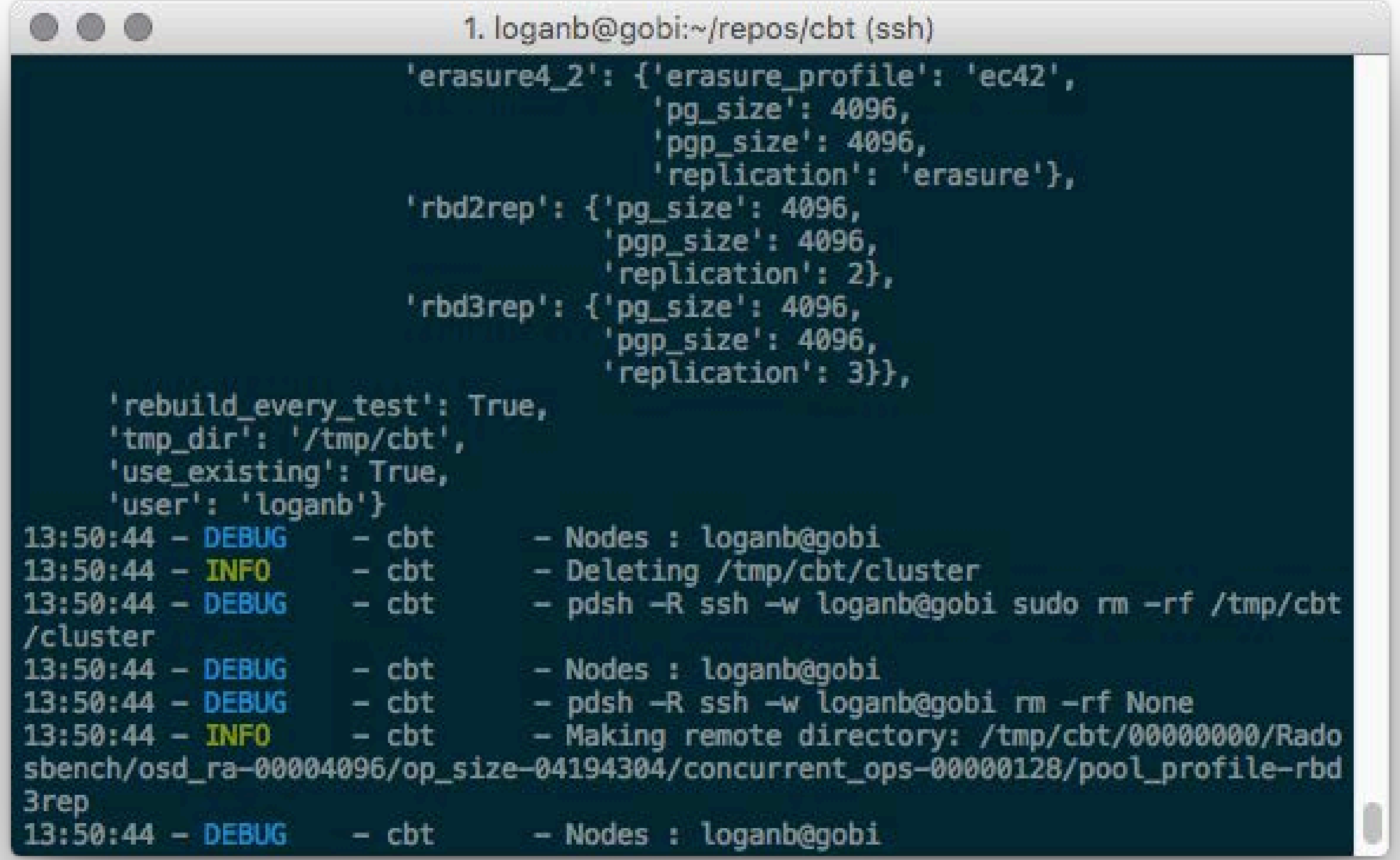
```
[loganb@head_node cbt]$ ./cbt.py –a
~/results/rados_bench ./yml_files/rados_bench.yml
```

CBT – Output – screenshot



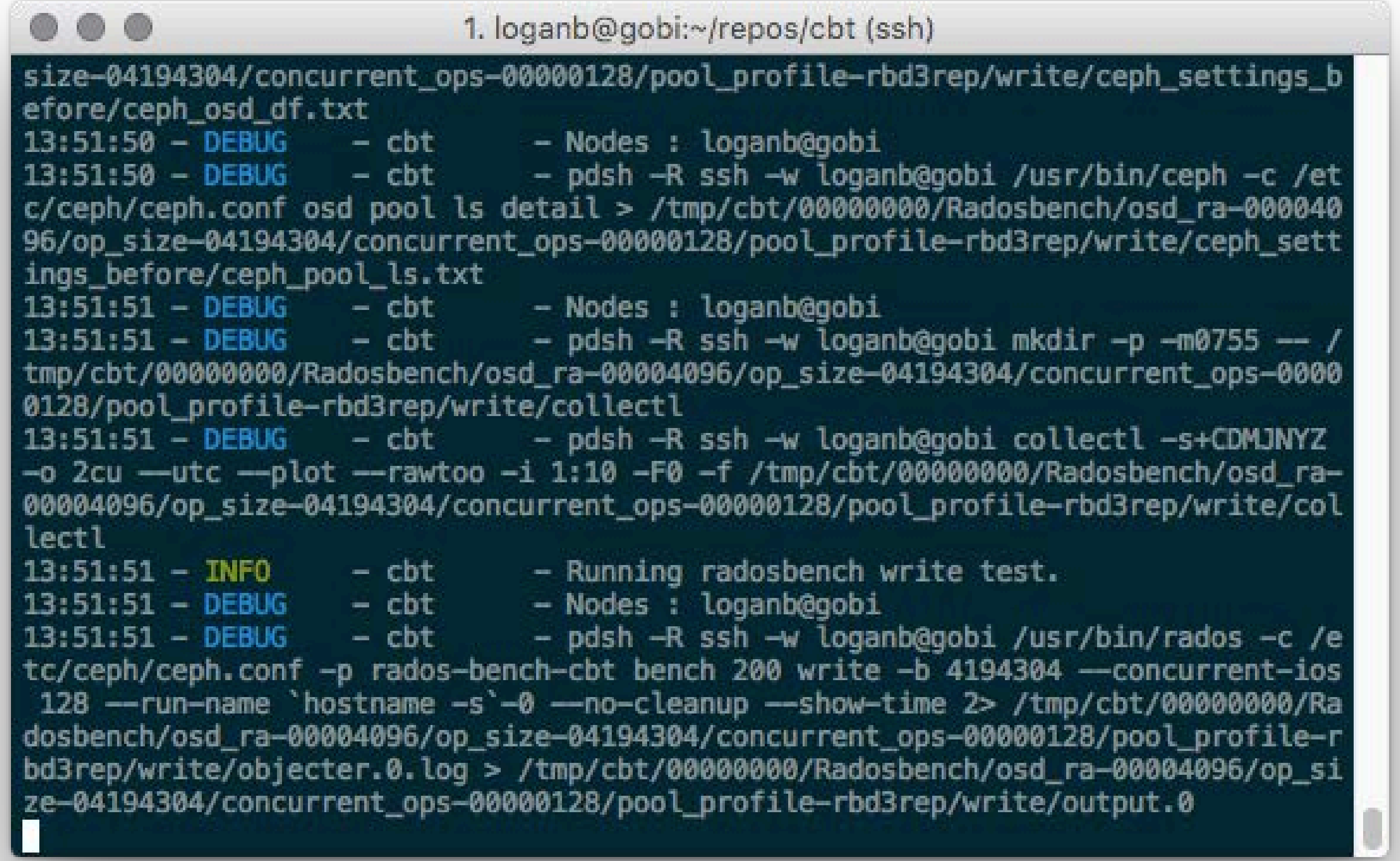
```
[loganb@gobi cbt]$ ./cbt.py -a ~/results1 ./yml_files/small.yml
13:59:26 - DEBUG    - cbt       - Settings.cluster:
    {'archive_dir': '/home/loganb/results1',
     'clients': ['gobi'],
     'clusterid': 'cluster',
     'conf_file': '/etc/ceph/ceph.conf',
     'erasure_profiles': {'ec32': {'erasure_k': 3, 'erasure_m': 2},
                          'ec42': {'erasure_k': 4, 'erasure_m': 2}},
     'fs': 'xfs',
     'head': 'gobi',
     'iterations': 1,
     'mkfs_opts': '-f -i size=2048 -n size=8k',
     'mons': ['gobi'],
     'mount_opts': 'noatime,nodiratime,attr2,logbufs=8,logbsize=256k,largeio,ino
de64,swalloc',
     'osds': ['gobi'],
     'osds_per_node': 10,
     'pool_profiles': {'erasure3_2': {'erasure_profile': 'ec32',
                                      'pg_size': 4096,
                                      'pgp_size': 4096,
                                      'replication': 'erasure'},
                       'erasure4_2': {'erasure_profile': 'ec42',
                                      'pg_size': 4096,
                                      'pgp_size': 4096,
```

CBT – Output – screenshot



```
                       'erasure4_2': {'erasure_profile': 'ec42',
                                      'pg_size': 4096,
                                      'pgp_size': 4096,
                                      'replication': 'erasure'},
                       'rbd2rep': {'pg_size': 4096,
                                   'pgp_size': 4096,
                                   'replication': 2},
                       'rbd3rep': {'pg_size': 4096,
                                   'pgp_size': 4096,
                                   'replication': 3}},
     'rebuild_every_test': True,
     'tmp_dir': '/tmp/cbt',
     'use_existing': True,
     'user': 'loganb'}
13:50:44 - DEBUG    - cbt        - Nodes : loganb@gobi
13:50:44 - INFO     - cbt        - Deleting /tmp/cbt/cluster
13:50:44 - DEBUG    - cbt        - pdsh -R ssh -w loganb@gobi sudo rm -rf /tmp/cbt
/cluster
13:50:44 - DEBUG    - cbt        - Nodes : loganb@gobi
13:50:44 - DEBUG    - cbt        - pdsh -R ssh -w loganb@gobi rm -rf None
13:50:44 - INFO     - cbt        - Making remote directory: /tmp/cbt/00000000/Rado
sbench/osd_ra-00004096/op_size-04194304/concurrent_ops-00000128/pool_profile-rbd
3rep
13:50:44 - DEBUG    - cbt        - Nodes : loganb@gobi
```

CBT – Output – screenshot

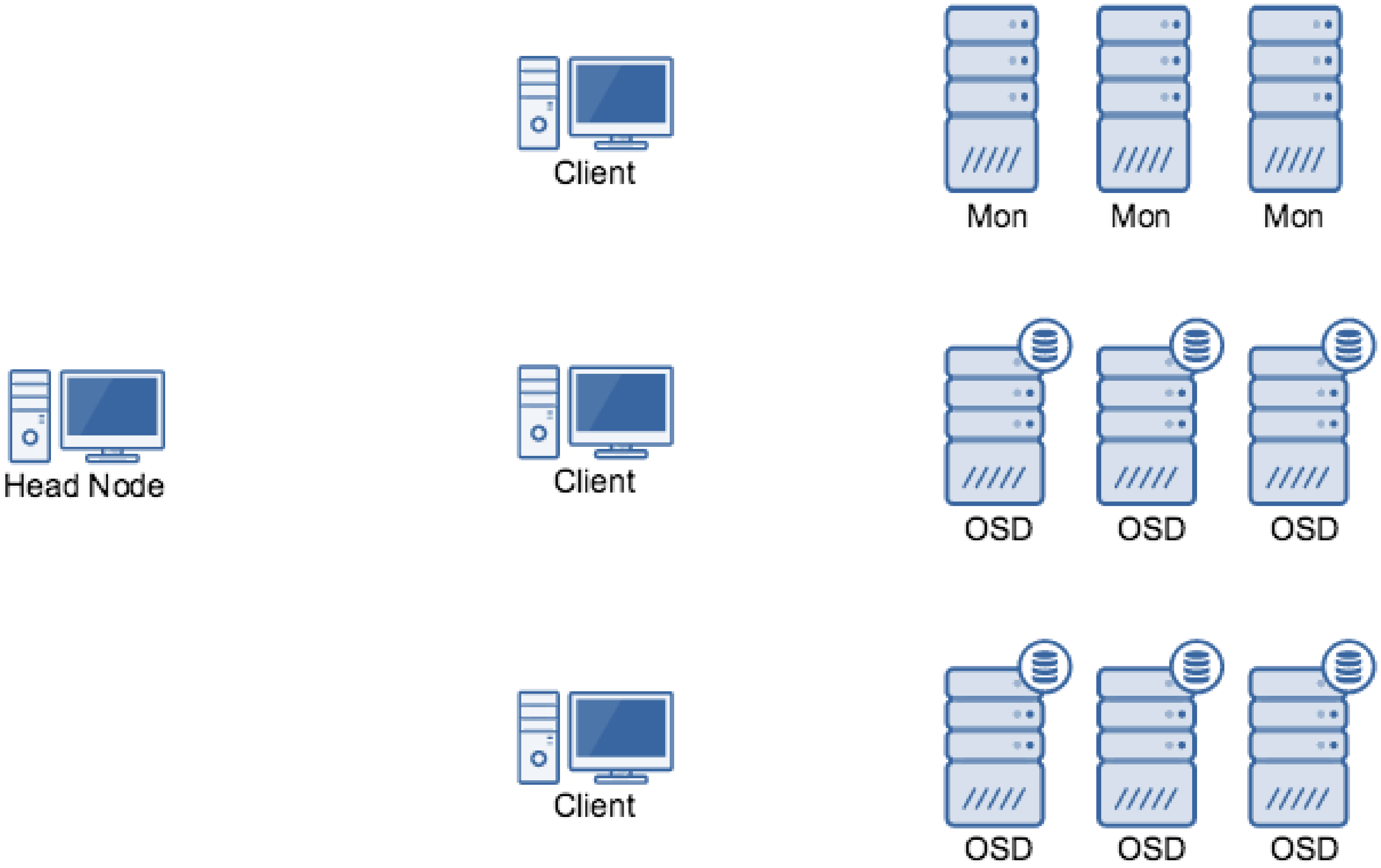


```
size-04194304/concurrent_ops-00000128/pool_profile-rbd3rep/write/ceph_settings_b
efore/ceph_osd_df.txt
13:51:50 - DEBUG    - cbt      - Nodes : loganb@gobi
13:51:50 - DEBUG    - cbt      - pdsh -R ssh -w loganb@gobi /usr/bin/ceph -c /et
c/ceph/ceph.conf osd pool ls detail > /tmp/cbt/00000000/Radosbench/osd_ra-000040
96/op_size-04194304/concurrent_ops-00000128/pool_profile-rbd3rep/write/ceph_sett
ings_before/ceph_pool_ls.txt
13:51:51 - DEBUG    - cbt      - Nodes : loganb@gobi
13:51:51 - DEBUG    - cbt      - pdsh -R ssh -w loganb@gobi mkdir -p -m0755 -- /
tmp/cbt/00000000/Radosbench/osd_ra-00004096/op_size-04194304/concurrent_ops-0000
0128/pool_profile-rbd3rep/write/collectl
13:51:51 - DEBUG    - cbt      - pdsh -R ssh -w loganb@gobi collectl -s+CDMJNYZ
-o 2cu --utc --plot --rawtoo -i 1:10 -F0 -f /tmp/cbt/00000000/Radosbench/osd_ra-
00004096/op_size-04194304/concurrent_ops-00000128/pool_profile-rbd3rep/write/col
lectl
13:51:51 - INFO     - cbt      - Running radosbench write test.
13:51:51 - DEBUG    - cbt      - Nodes : loganb@gobi
13:51:51 - DEBUG    - cbt      - pdsh -R ssh -w loganb@gobi /usr/bin/rados -c /e
tc/ceph/ceph.conf -p rados-bench-cbt bench 200 write -b 4194304 --concurrent-ios
 128 --run-name `hostname -s`-0 --no-cleanup --show-time 2> /tmp/cbt/00000000/Ra
dosbench/osd_ra-00004096/op_size-04194304/concurrent_ops-00000128/pool_profile-r
bd3rep/write/objecter.0.log > /tmp/cbt/00000000/Radosbench/osd_ra-00004096/op_si
ze-04194304/concurrent_ops-00000128/pool_profile-rbd3rep/write/output.0
```

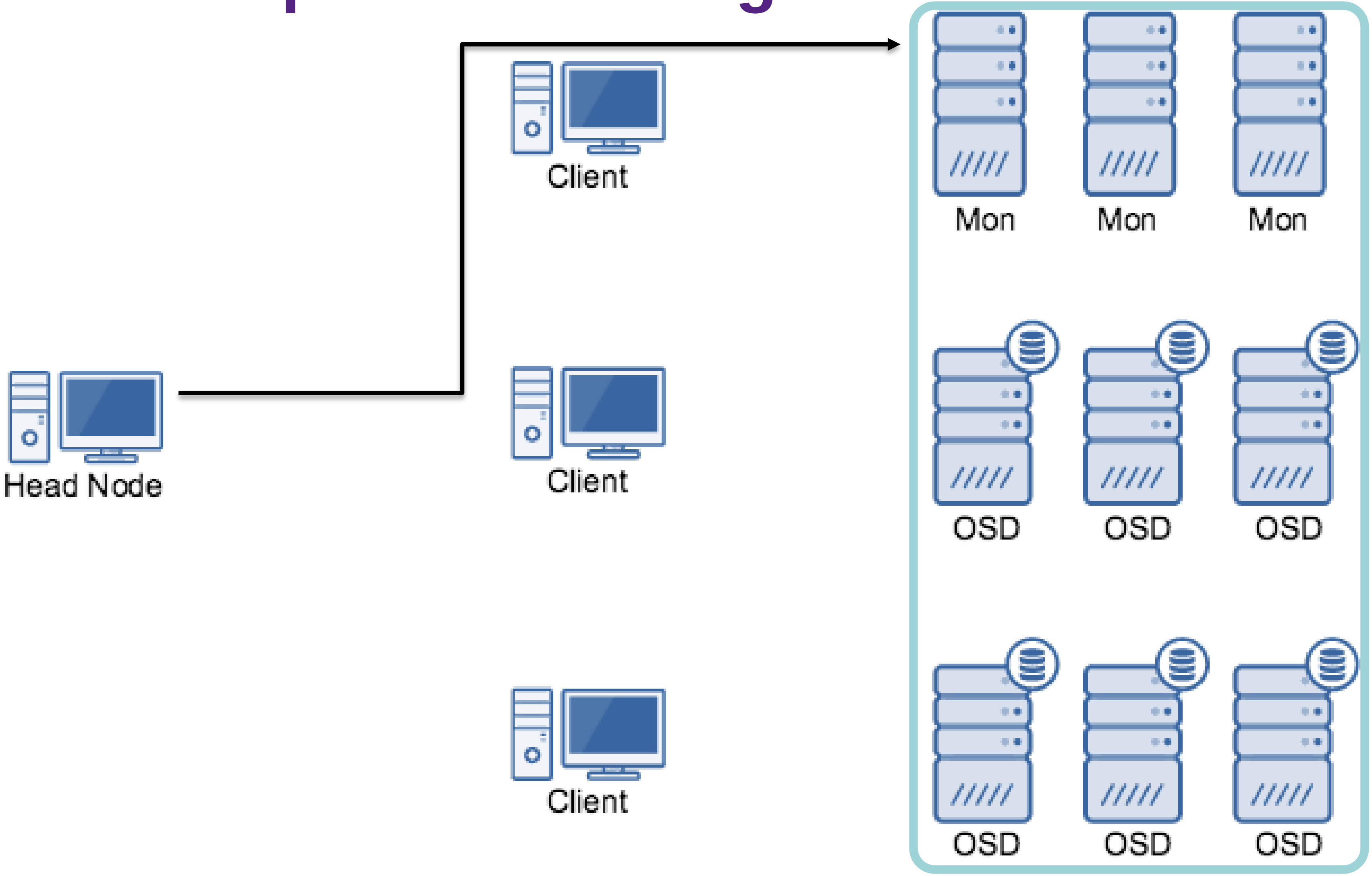
What CBT is actually doing

more unsplash

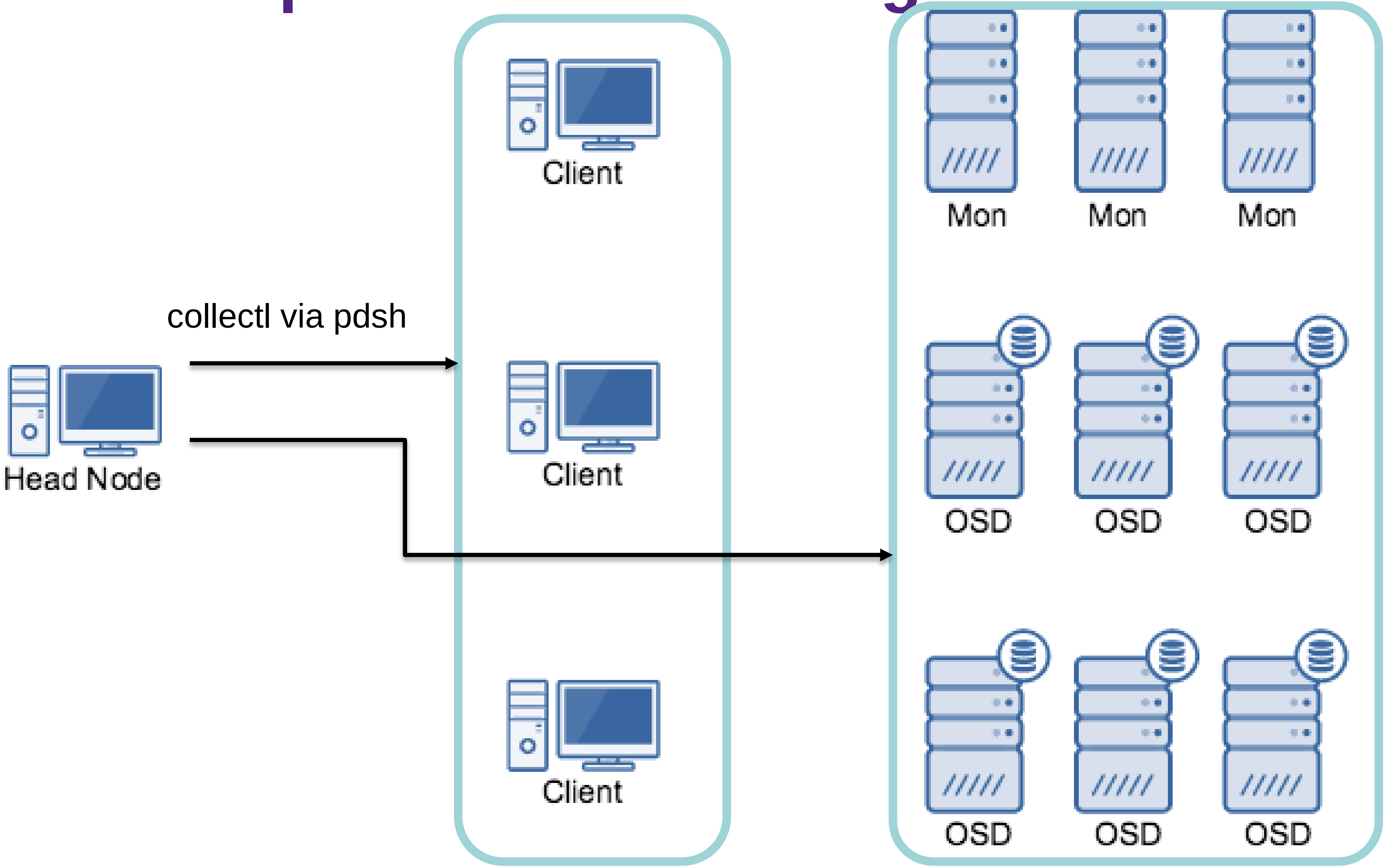
CBT


Client
Mon
Mon
Mon
Head Node
Client
OSD
OSD
OSD
Client
OSD
OSD
OSD

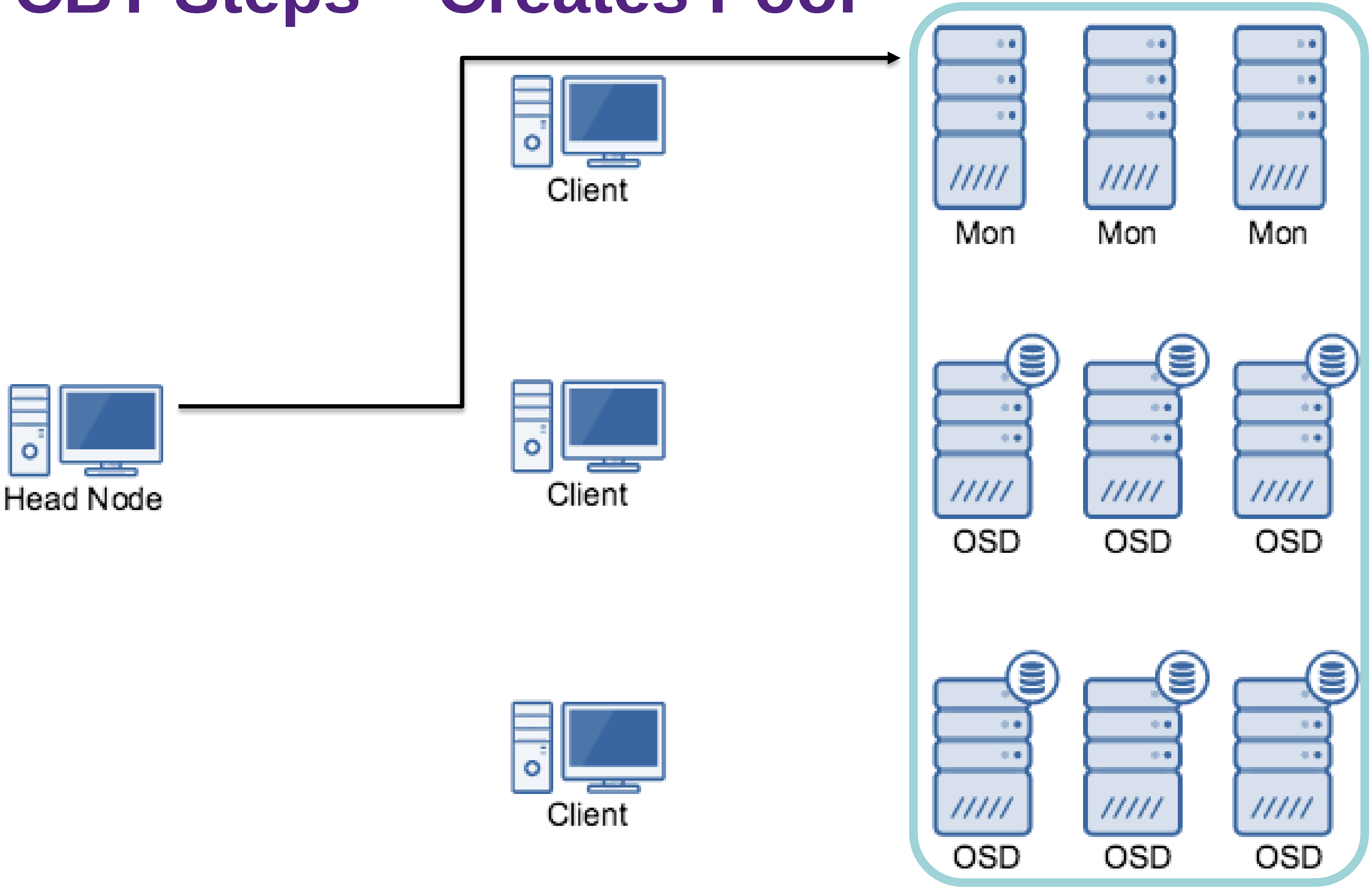
CBT Steps – Scrubbing check



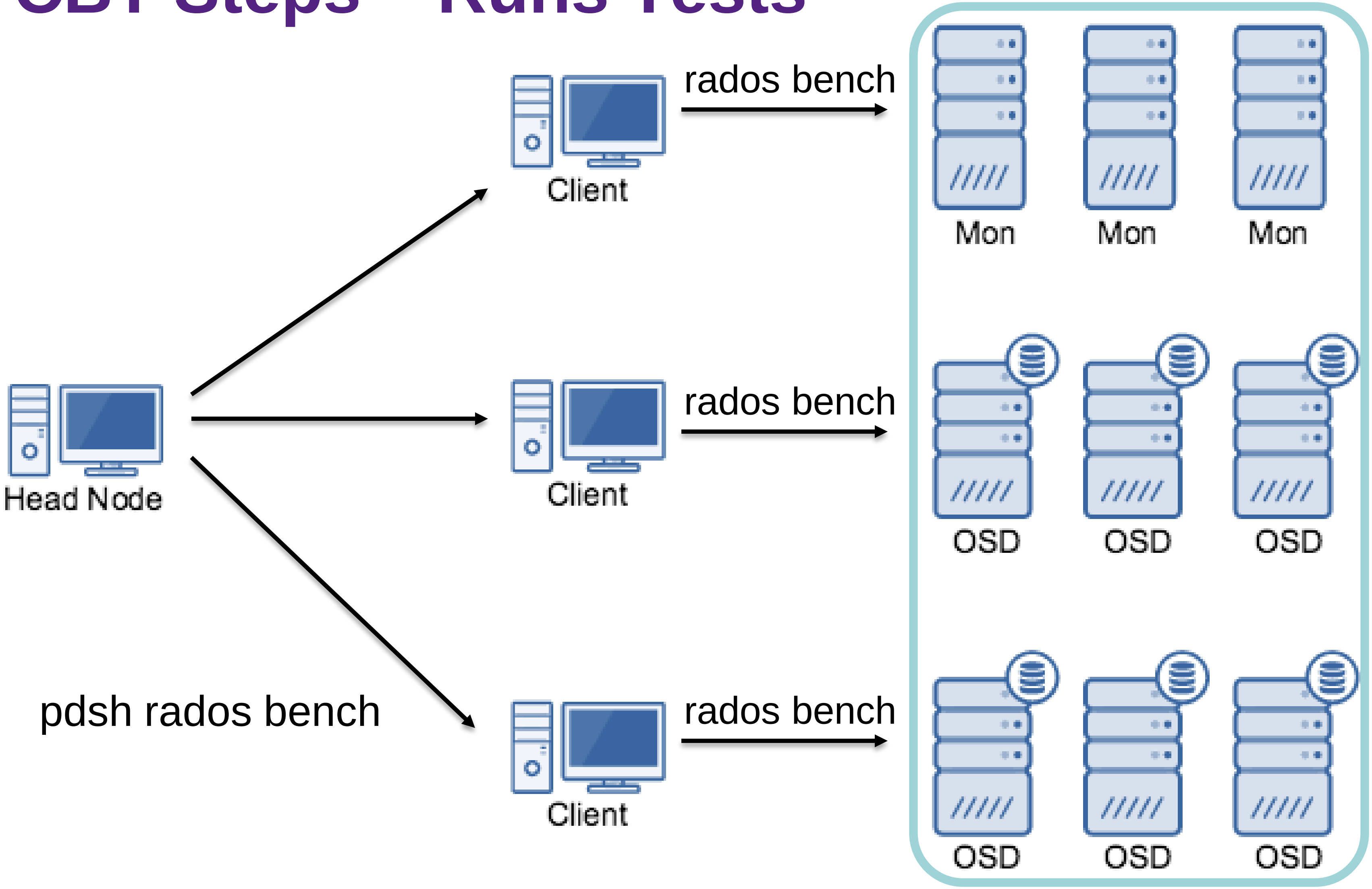
CBT Steps – Idle Monitoring - 60s



CBT Steps – Creates Pool

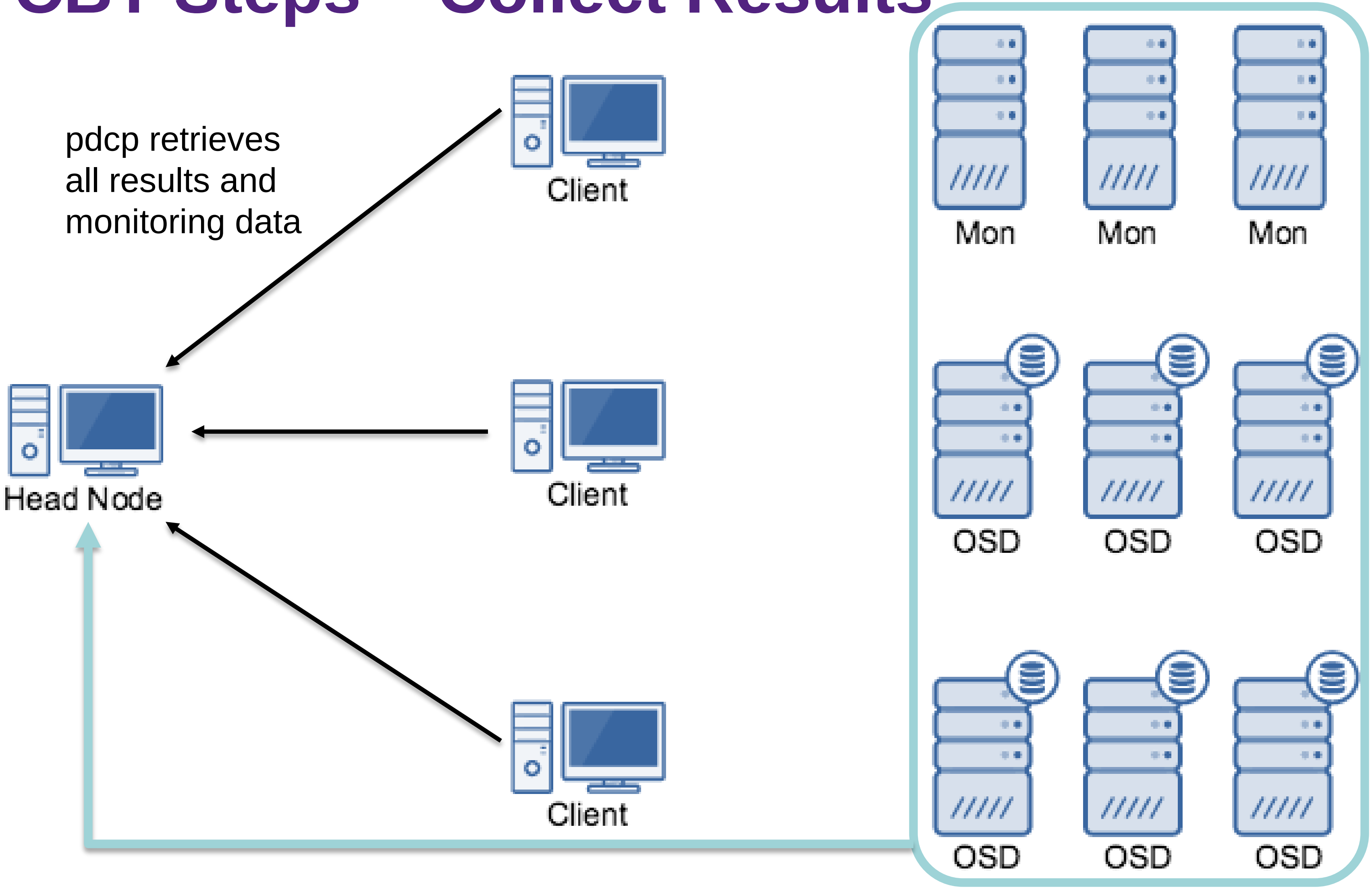


CBT Steps – Runs Tests

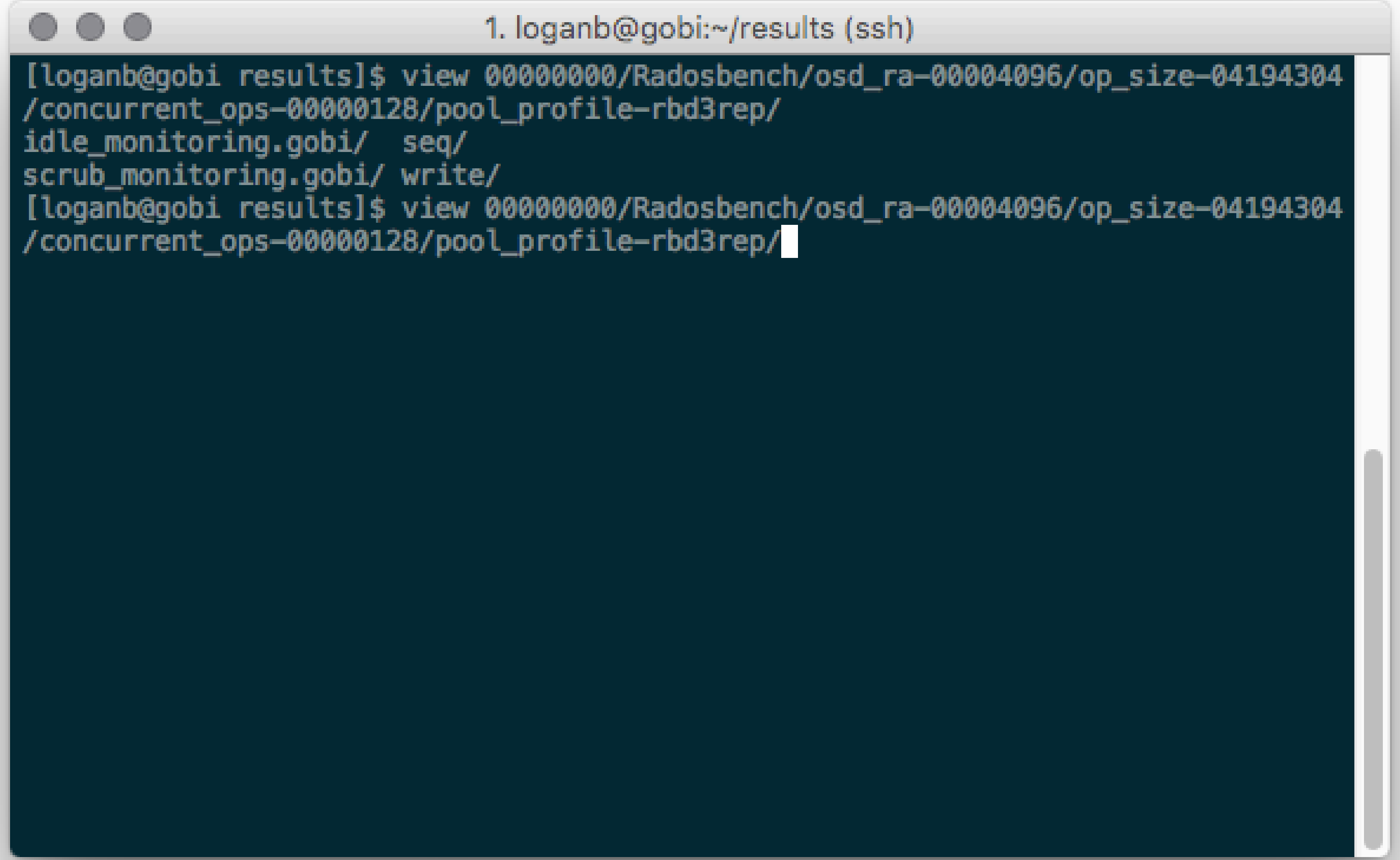


CBT Steps – Collect Results

pdcp retrieves all results and monitoring data



CBT Results – On head node



```
[loganb@gobi results]$ view 00000000/Radosbench/osd_ra-00004096/op_size-04194304
/concurrent_ops-00000128/pool_profile-rbd3rep/
idle_monitoring.gobi/  seq/
scrub_monitoring.gobi/ write/
[loganb@gobi results]$ view 00000000/Radosbench/osd_ra-00004096/op_size-04194304
/concurrent_ops-00000128/pool_profile-rbd3rep/
```

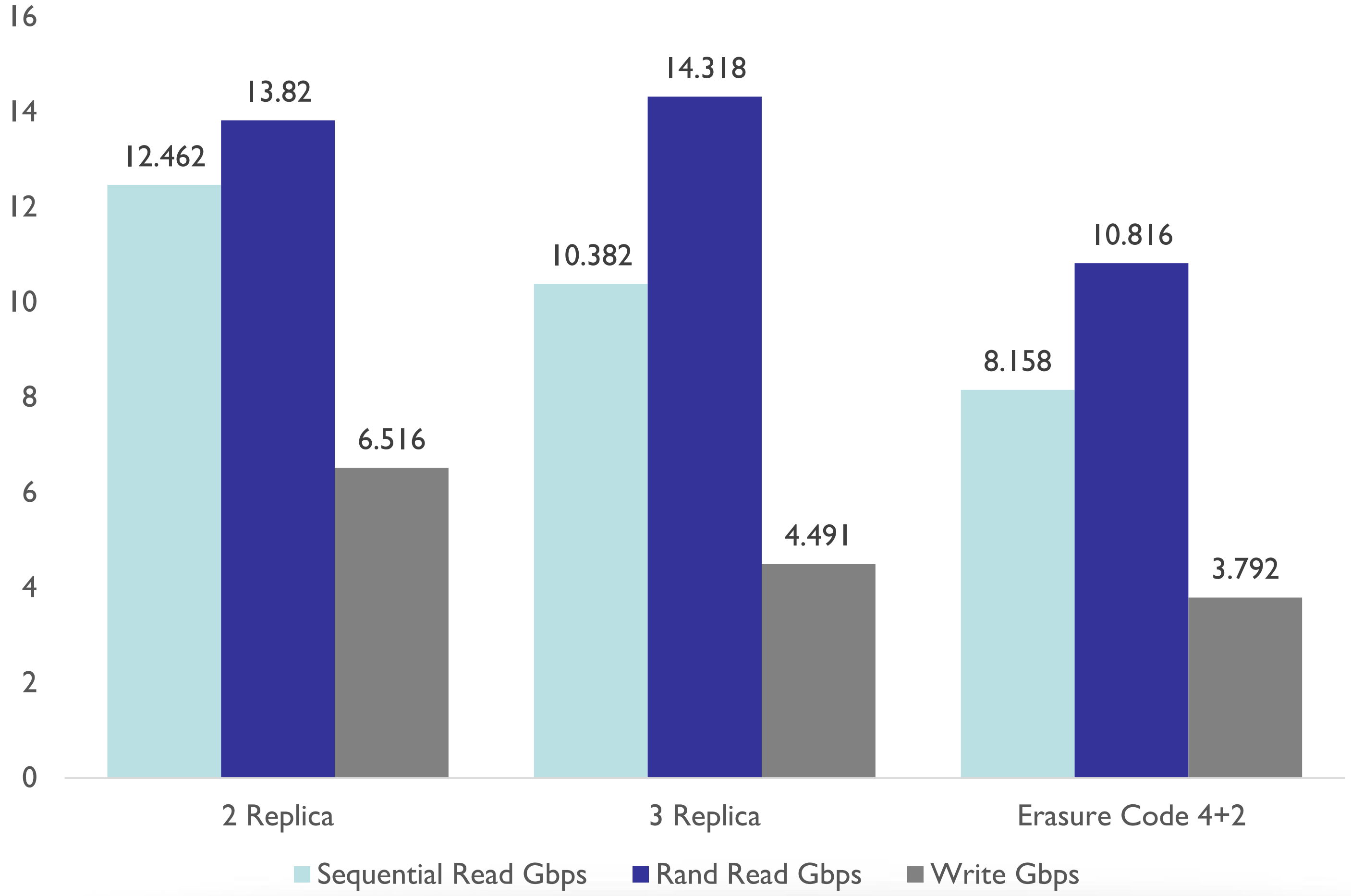
CBT Results – Location

- 00000000 – Iteration of test
- radosbench, osd_ra, op_size, concurrent_ops, pool_profile – all from the yml file
- write – IO type, could be write, seq, or rand
- output.0.gobi – benchmark output.Instance.hostname

```
[loganb@gobi ~]$ view
results/00000000/Radosbench/osd_ra-
00004096/op_size-04194304/concurrent_ops-
00000128/pool_profile-rbd3rep/write/output.0.gobi
```

Rados Bench 4MiB Object Size

Sum of client throughput from rados bench / number of data nodes = per data node throughput



Data Nodes – 4RU

- Drives – 30 7.2k SATA, 6 SATA SSDs for Journals
- 2 Intel E5-2630 v3
- 128GiB RAM
- Mellanox ConnectX3-Pro – in MLAG for public
- Mellanox ConnectX3-Pro – in MLAG for cluster
- Test was 100% reads, or 100% writes, no mixing
- 8 Clients
- Host level failure domain

CBT Direction and Extension

- Adding your own benchmark is pretty simple, inherit a python base class
  - Beware of run_dir and out_dir
- Looking to add uncore monitoring
- Pull request for a plugin style architecture for monitoring
- Sysctl setting compare tool

Relevant Sysctl Settings

- Check your NIC vendors recommendations
- sched_{min|wakeup}granularity_ns
- kernel.pid_max
- vm.vfs_cache_pressure
- vm.swappiness
- pcie_bus_perf <-actually a kernel boot option

Ceph Settings

- To change runtime settings
  - ceph tell osd.* injectargs –param-name val
- max_filestore_sync_interval
- osd_op_threads

Thank you!

Thanks to Mark Nelson mnelson@redhat.com for the feedback on the slides

Logan.blyth@ccur.com

Aquaristorage.com