FUJITSU

shaping tomorrow with you

Igor Podoski / Zofia Domaradzka

Ghost Cluster and Throwing Fireballs

Agenda

- Ghost Cluster
- Throwing Fireballs
- Questions??

ETERNUS CD10000

- A Fujitsu software-defined storage system based on Ceph and RHEL7
- Appliance fully integrated with and automated on Fujitsu Primergy Servers
- Provides custom tools for installation, configuration, monitoring, diagnostics etc.

ETERNUS CD10000 - monitoring

- Custom monitoring system using CD10000 snmp agents
- Monitoring of PGs, OSDs, monitors and overall cluster state
- Active polling and traps
- Responsiveness for a large cluster must be tested (e.g. cluster with 224 nodes)

Testing monitoring system responsiveness

Challenges:

Testing for different ceph configurations and cluster sizes, e.g.:

- Large number of PGs
- Cluster at full/near full state
- Change of state of specific MOs at given moment or several states at once

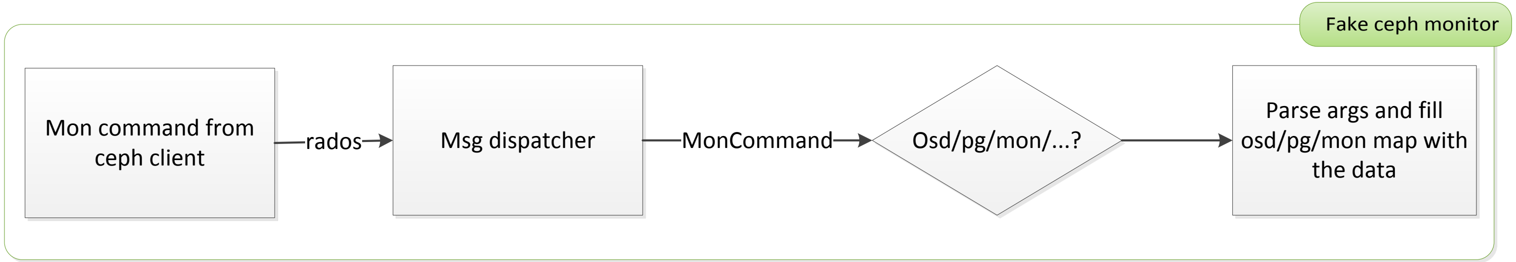
Solution

GHOST CLUSTER

How does it work?

- Work in progress stand alone process simulating ceph monitor
- Based on the fragments of ceph monitor code
- Uses RADOS protocol, so it is compatible with all Ceph clients (/usr/bin/ceph etc...)
- Uses MON / OSD / PG maps for storing fake objects in the process memory

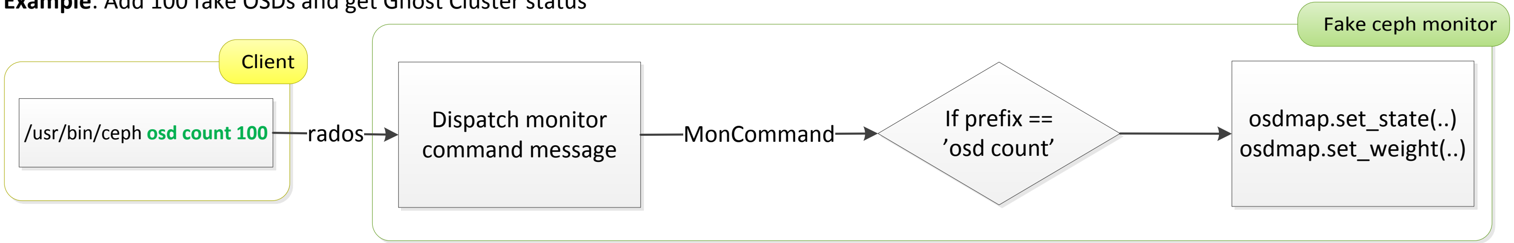
How does it work?



Fake ceph monitor

Mon command from ceph client → rados → Msg dispatcher → MonCommand → Osd/pg/mon/...? → Parse args and fill osd/pg/mon map with the data

Example: Add 100 fake OSDs and get Ghost Cluster status

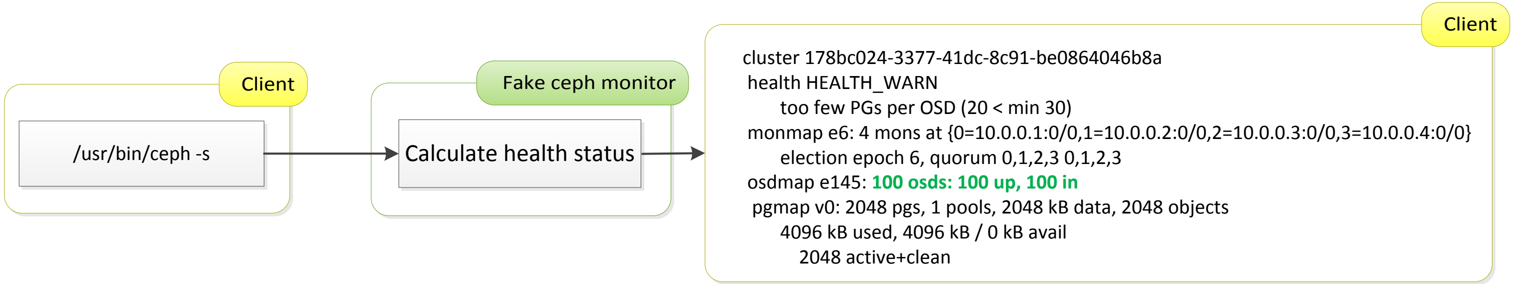


Client

/usr/bin/ceph **osd count 100** → rados →

Fake ceph monitor

Dispatch monitor command message → MonCommand → If prefix == 'osd count' → osdmap.set_state(..) osdmap.set_weight(..)



Client

/usr/bin/ceph -s →

Fake ceph monitor

Calculate health status →

Client

```
cluster 178bc024-3377-41dc-8c91-be0864046b8a
 health HEALTH_WARN
       too few PGs per OSD (20 < min 30)
 monmap e6: 4 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0,3=10.0.0.4:0/0}
       election epoch 6, quorum 0,1,2,3 0,1,2,3
 osdmap e145: 100 osds: 100 up, 100 in
  pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
       4096 kB used, 4096 kB / 0 kB avail
          2048 active+clean
```

Current functionality

- MONmap manipulations:
  - add/remove monitors
  - change quorum status
- OSDmap manipulations:
  - change number of OSDs
  - change state of each OSD (full, nearfull, etc.)
- PGmap manipulations:
  - change numer of PGs
  - change state of each PG (active+clean, degraded, etc.)
- Overall cluster status manipulations:
  - change cluster flags (full, etc.)

Benefits

- Time and resource saver for:
  - reconfiguring and filling up real Ceph cluster
  - no need to use physical cluster, single process can be run on any virtual environment
- Flexible configuration and resposiveness:
  - change of parameters on the fly
  - immediate cluser state response
- Easier automation of test scenarios:
  - predefined configuration profiles can be used
  - state transision can be also emulated e.g. long PGs recovery time

Demo

- Fake Ceph monitor was started on localhost
- Currently PG / MON / OSD maps are empty:

```
[root@localhost build]# ceph -s
    cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_ERR
            no osds
     monmap e0: 0 mons at {}
            election epoch 0, quorum
     osdmap e1: 0 osds: 0 up, 0 in
      pgmap v0: 0 pgs, 0 pools, 0 bytes data, 0 objects
            0 kB used, 0 kB / 0 kB avail
```

Demo

- Let’s have a look at allowed options:

```
[root@localhost build]# ceph -h

...

mon add <name> <IPaddr[:port]>                   add new monitor with <name> ip <ip:[port]>
mon quorum <quorum> [<quorum>...]                set quorum <0 1 2>
mon rm <name>                                    remove monitor <name>
mon skew <int[0-]>                               set skew <seconds>
osd <int[0-]> <state>                            set <state> on osd <num>
osd count <int[0-]>                              set <num> osds
osd set <flag>                                   set <flag> on osdmap
osd unset <flag>                                 unset <flag> on osdmap
osd usage <int> <int[0-]> <int[0-]> <int[0-]>    add usage to <num> osd with <kb> <kb_used> <kb_avail>
pg count <int[0-]> <int[0-]> <int[0-]> <int[0-]> set <pool> <size> <obj> <num> pgs
status                                           show cluster status
```

Demo

FUJITSU

Set predefined profile:

```
#!/bin/bash
ceph mon add 0 10.0.0.1
ceph mon add 1 10.0.0.2
ceph mon add 2 10.0.0.3
ceph mon quorum 0 1 2

ceph osd count 20

ceph pg count 0 1024 1 2048

ceph osd usage 0 0 1024 1024
ceph osd usage 1 0 1024 1024
ceph osd usage 2 0 1024 1024
ceph osd usage 3 0 1024 1024
```

Get ghost cluster status:

```
[root@localhost build]# ceph -s
    cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_OK
     monmap e4: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 4, quorum 0,1,2 0,1,2
     osdmap e21: 20 osds: 20 up, 20 in
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Demo

- Reduce quorum to mon.0 and mon.1

```
[root@localhost build]# ceph mon quorum 0 1

[root@localhost build]# ceph -s
    cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_WARN
            1 mons down, quorum 0,1 0,1
     monmap e5: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 5, quorum 0,1 0,1
     osdmap e21: 20 osds: 20 up, 20 in
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Demo

- Emulate OSD.0 down

```
[root@localhost build]# ceph osd 0 down

[root@localhost build]# ceph -s
    cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_WARN
            1 mons down, quorum 0,1 0,1
            1/20 in osds are down
     monmap e5: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 5, quorum 0,1 0,1
     osdmap e23: 20 osds: 19 up, 20 in
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Demo

Add monitor clock skew

```
[root@localhost build]# ceph mon skew 2

[root@localhost build]# ceph -s
cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_WARN
            clock skew detected on mon.0
            1 mons down, quorum 0,1 0,1
            1/20 in osds are down
            Monitor clock skew detected
     monmap e5: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 5, quorum 0,1 0,1
     osdmap e23: 20 osds: 19 up, 20 in
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Demo

- Emulate nearfull OSD.1

```
[root@localhost build]# ceph osd 1 nearfull

[root@localhost build]# ceph -s
cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_WARN
            clock skew detected on mon.0
            1 near full osd(s)
            1 mons down, quorum 0,1 0,1
            1/20 in osds are down
            Monitor clock skew detected
     monmap e5: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 5, quorum 0,1 0,1
     osdmap e24: 20 osds: 19 up, 20 in
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Demo

Set cluster full flag

```
[root@localhost build]# ceph osd set full

[root@localhost build]# ceph -s
cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_WARN
            clock skew detected on mon.0
            1 near full osd(s)
            1 mons down, quorum 0,1 0,1
            1/20 in osds are down
            Monitor clock skew detected
     monmap e5: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 5, quorum 0,1 0,1
     osdmap e24: 20 osds: 19 up, 20 in
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Demo

- Emulate OSD.2 full

```
[root@localhost build]# ceph osd 2 full
[root@localhost build]# ceph -s
    cluster 1bb821e7-4550-4f1b-baec-259e2809261a
     health HEALTH_ERR
            clock skew detected on mon.0
            1 full osd(s)
            1 near full osd(s)
            1 mons down, quorum 0,1 0,1
            1/20 in osds are down
            full flag(s) set
            Monitor clock skew detected
     monmap e9: 3 mons at {0=10.0.0.1:0/0,1=10.0.0.2:0/0,2=10.0.0.3:0/0}
            election epoch 9, quorum 0,1 0,1
     osdmap e113: 20 osds: 19 up, 20 in
            flags full
      pgmap v0: 2048 pgs, 1 pools, 2048 kB data, 2048 objects
            4096 kB used, 4096 kB / 0 kB avail
                2048 active+clean
```

Throwing fireballs

What it is?

Throwing fireballs into Ceph means:

- Break stuff e.g.
  - Add 10% packet drop to public interface for node with mon0
  - Add 100ms network delay to cluster interface on different node
  - Pin all ceph-osd processes from node with mon1 to one logical CPU core
  - Move all ms_dispatch threads from all ceph-osds on node without monitors to one logical CPU core
  - Misconfigure OSD parameters in resobanble way
  - Filling up OSD partitions with non PG stuff (e.g. using dd)
- See Ceph reaction:
  - When / where / how it breaks
- Create a cure for newly created dissease:
  - Analyse ceph logs and potential core dumps
  - Deduce probablity of newly created conditions and prepare a solution

Tools for throwing fireballs

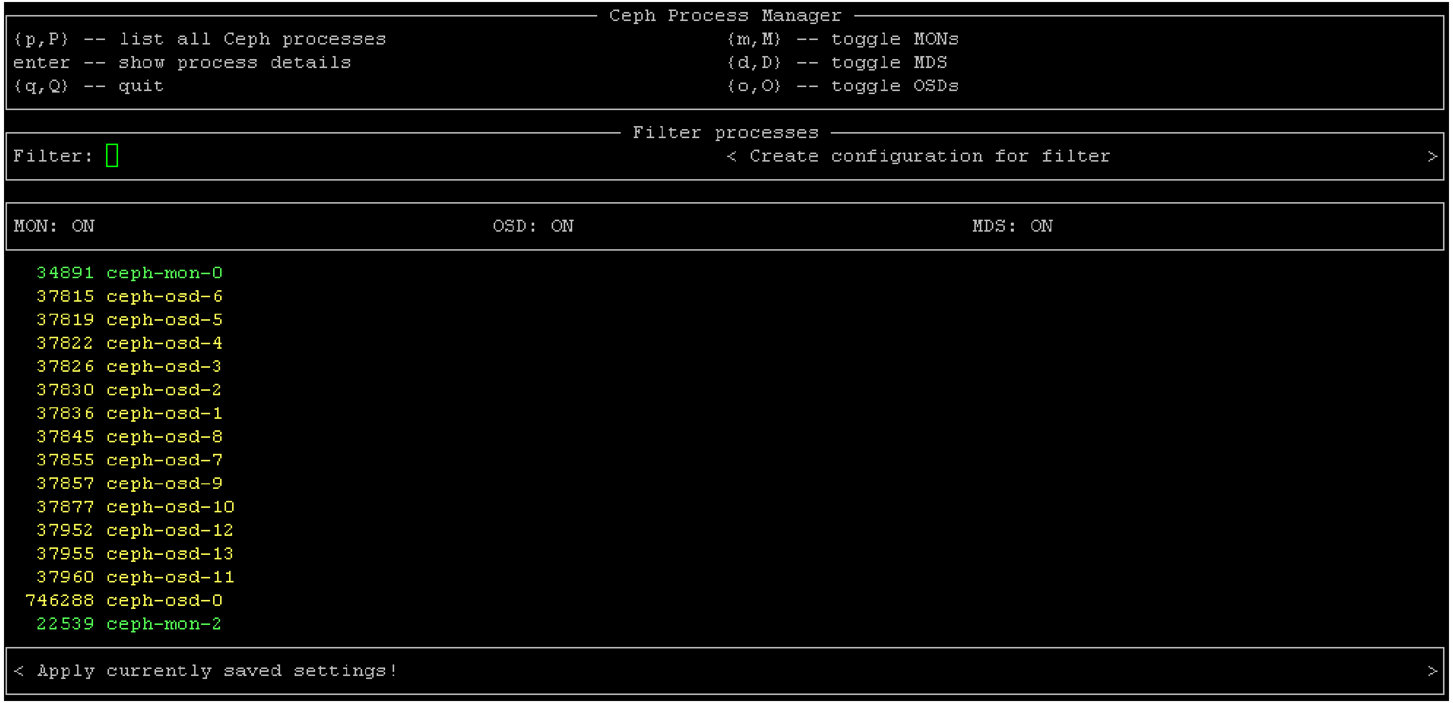
- Ceph configuration poisoning:
  - Injecting args at runtime
  - Permanent changes in ceph.conf
- System tools:
  - tuned, tc, /proc files, iptables, changing XFS properties, etc.
- Dedicated tools:
  - Newly created CPM (Ceph Process Manager)
  - Dedicated scripts and code snippets

CPM - Ceph Process Manager

- Uses python and salt to interact with Ceph cluster
- Manages Ceph processes at higher level:
  - Doesn’t matter on which node ceph-* are running
  - Keeps configuration in a flat JSON file
  - Uses regular expressions to match process and thread names
- Can tune several things (for processes and individual threads)
  - Set any logical CPUs on which can run
  - Change nice priority of processes and threads
  - Change scheduling and real-time priority
  - Change I/O scheduling policy and priority
- Uses python and custom salt module
  - It will be released soon

CPM demo

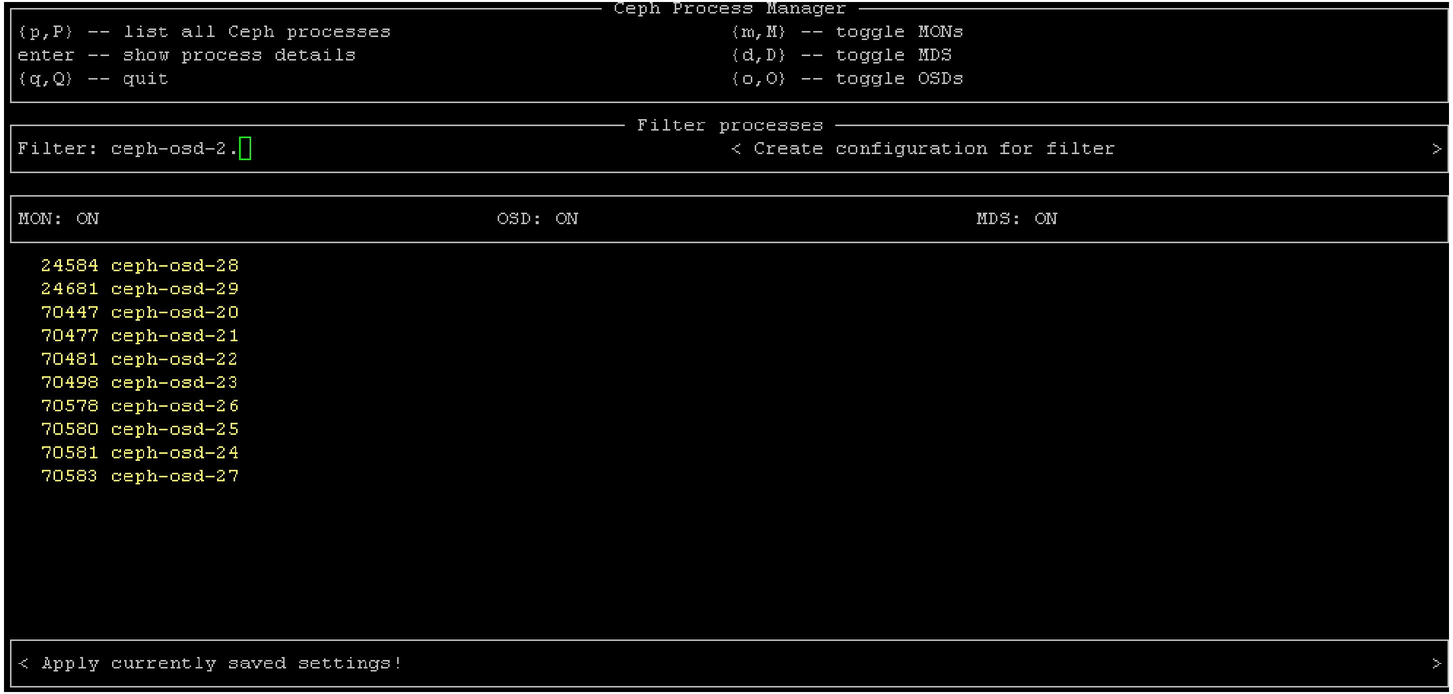
CPM can be started with GUI or in batch mode only.



```
                                              Ceph Process Manager
{p,P} -- list all Ceph processes                              {m,M} -- toggle MONs
enter -- show process details                                 {d,D} -- toggle MDS
{q,Q} -- quit                                                 {o,O} -- toggle OSDs

                                                Filter processes
Filter:                                                       < Create configuration for filter                   >

MON: ON                                 OSD: ON                                 MDS: ON

  34891 ceph-mon-0
  37815 ceph-osd-6
  37819 ceph-osd-5
  37822 ceph-osd-4
  37826 ceph-osd-3
  37830 ceph-osd-2
  37836 ceph-osd-1
  37845 ceph-osd-8
  37855 ceph-osd-7
  37857 ceph-osd-9
  37877 ceph-osd-10
  37952 ceph-osd-12
  37955 ceph-osd-13
  37960 ceph-osd-11
 746288 ceph-osd-0
  22539 ceph-mon-2

< Apply currently saved settings!                                                                                 >
```

CPM demo

Processes can be filtred by reglar expressions.

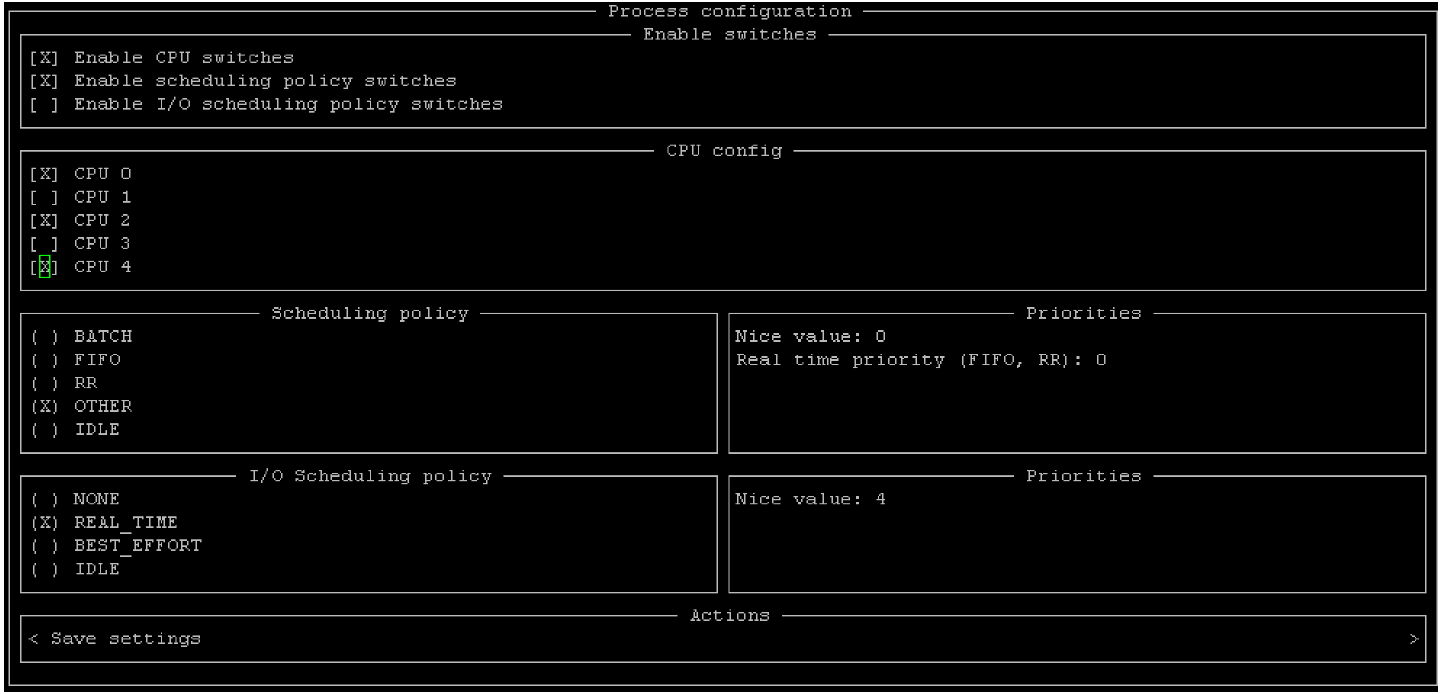


```
Ceph Process Manager
{p,P} -- list all Ceph processes                          {m,M} -- toggle MONs
enter -- show process details                             {d,D} -- toggle MDS
{q,Q} -- quit                                             {o,O} -- toggle OSDs

Filter processes
Filter: ceph-osd-2.                                       < Create configuration for filter          >

MON: ON                          OSD: ON                          MDS: ON

  24584 ceph-osd-28
  24681 ceph-osd-29
  70447 ceph-osd-20
  70477 ceph-osd-21
  70481 ceph-osd-22
  70498 ceph-osd-23
  70578 ceph-osd-26
  70580 ceph-osd-25
  70581 ceph-osd-24
  70583 ceph-osd-27

< Apply currently saved settings!                                                                    >
```

CPM demo

In process view serveral options can be chosen.

Settings will be saved in JSON format.



```
                              Process configuration
                                Enable switches
[X]  Enable CPU switches
[X]  Enable scheduling policy switches
[ ]  Enable I/O scheduling policy switches

                                  CPU config
[X]  CPU 0
[ ]  CPU 1
[X]  CPU 2
[ ]  CPU 3
[X]  CPU 4

           Scheduling policy                               Priorities
( )  BATCH                                  Nice value: 0
( )  FIFO                                   Real time priority (FIFO, RR): 0
( )  RR
(X)  OTHER
( )  IDLE

          I/O Scheduling policy                            Priorities
( )  NONE                                   Nice value: 4
(X)  REAL_TIME
( )  BEST_EFFORT
( )  IDLE

                                   Actions
<  Save settings                                                              >
```

CPM demo

- JSON config example:
  - Pin every osd process on the whole cluster to logical cpu core 0 and 1
  - Change will be made on all nodes in the cluster

```
{
    "ceph-osd-*": {
        "scheduling": {
            "policy": "OTHER",
            "priority": 0
        },
        "ionice": {
            "policy": "REAL_TIME",
            "priority": 4
        },
        "enable": {
            "io_sched": false,
            "sched": false,
            "cpu": true
        },
        "taskset": [0,1],
        "thread_name": "",
        "nice": 0
    }
}
```

CPM demo

- JSON config example:
  - Move ms_dispatch thread for every ceph-osd process to logical cpu cores: 3,4,6,18
  - Change will be made on all nodes in the cluster

```
{
    "ceph-osd-*": {
        "scheduling": {
            "policy": "OTHER",
            "priority": 0
        },
        "ionice": {
            "policy": "REAL_TIME",
            "priority": 4
        },
        "enable": {
            "io_sched": false,
            "sched": false,
            "cpu": true
        },
        "taskset": [3,4,6,18],
        "thread_name": "ms_dispatch",
        "nice": 0
    }
}
```

- To apply JSON profile:

```
> python cpm.py --apply profile.json
```

Thowing fireballs in the wild

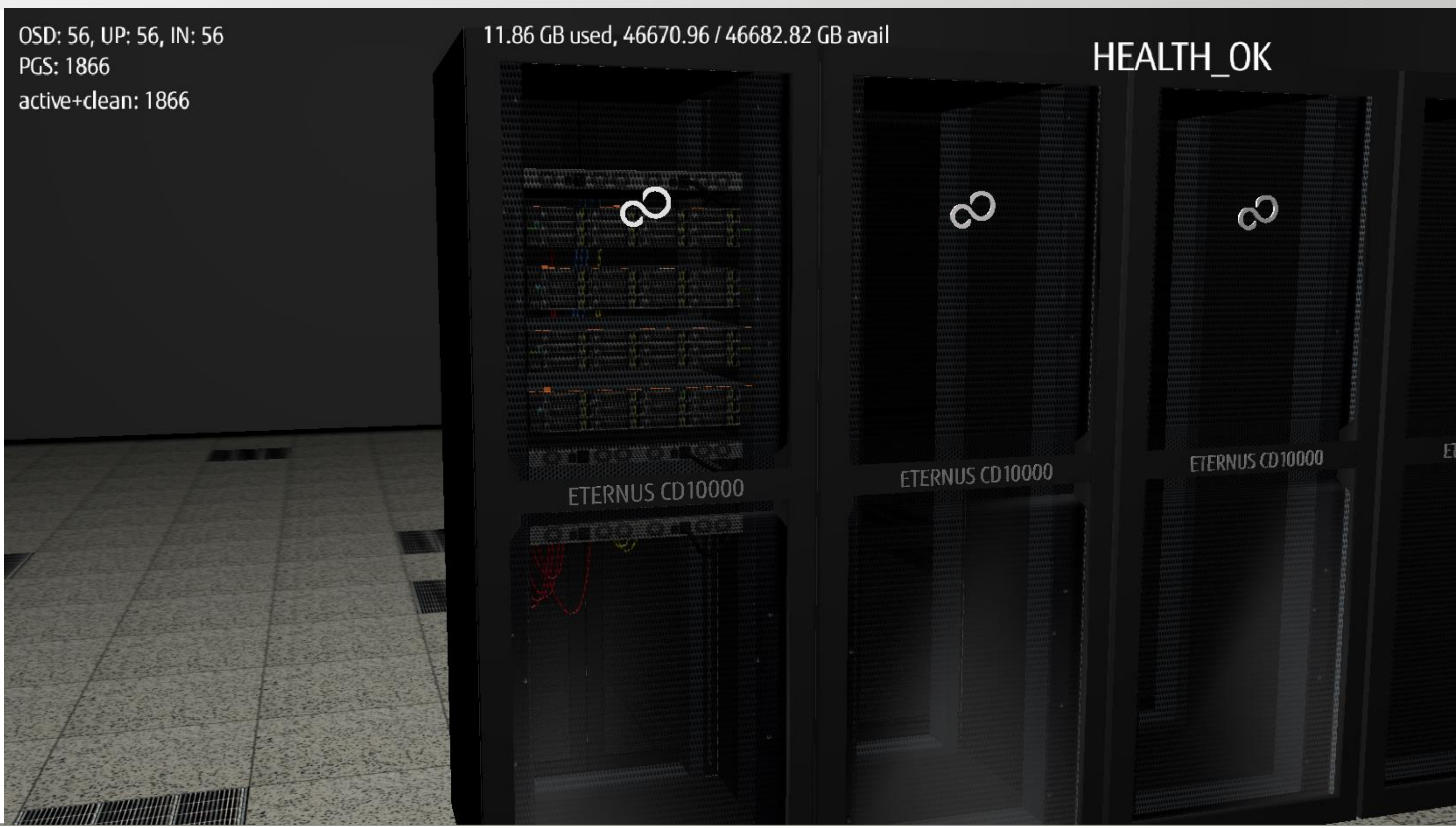
- How to present this technique?
- Is there a way to:
  - Make it more real than just flat files and terminal commands?
  - Move it to different level of abstraction?
  - Make it more fun?
- Blender comes for the rescue!
  - Game-like interface for throwing fireballs
  - Realtime logs and Ceph status on HUD display
  - True interaction with physical servers
  - Interaction through librados and salt

Let’s play!

- Controls:
  - Mouse look
  - W S A D keyboard for movement
- Graphics:
  - 3D models of ETERNUS CD10000 appliance

Heads-Up Display

- Left
  - OSD count, UP vs IN
  - PG count
  - PG states
- Center
  - Cluster usage in GB
- Right
  - Health status
  - Health summary



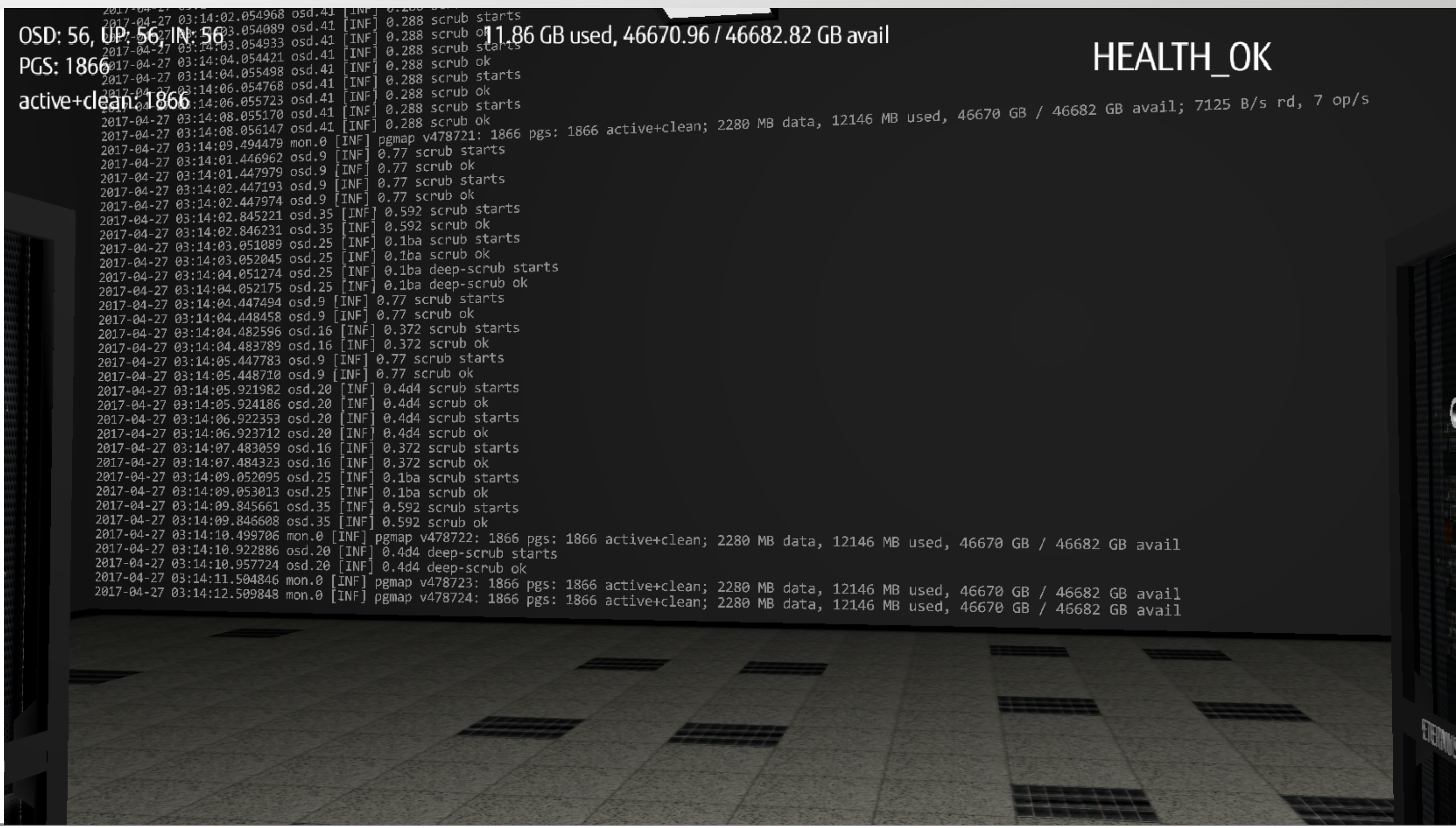
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

11.86 GB used, 46670.96 / 46682.82 GB avail

HEALTH_OK

Monitor log wall

Realtime update from monitor log callback (python)



OSD: 56, UP: 56, IN: 56

PGS: 1866

active+clean: 1866

11.86 GB used, 46670.96 / 46682.82 GB avail

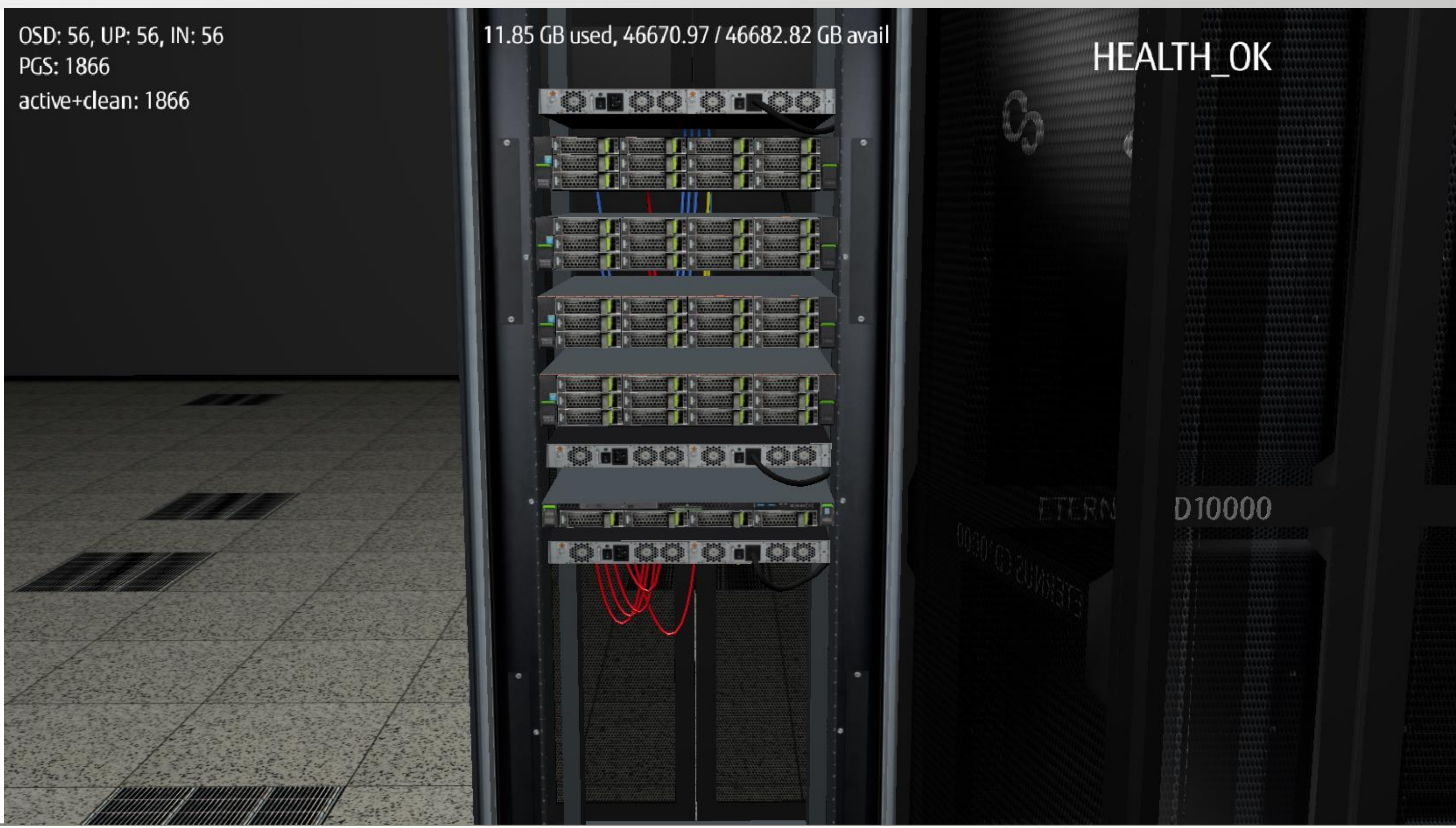
HEALTH_OK

```
2017-04-27 03:14:02.054968 osd.41 [INF] 0.288 scrub starts
2017-04-27 03:14:03.054089 osd.41 [INF] 0.288 scrub ok
2017-04-27 03:14:03.054933 osd.41 [INF] 0.288 scrub starts
2017-04-27 03:14:04.054421 osd.41 [INF] 0.288 scrub ok
2017-04-27 03:14:04.055498 osd.41 [INF] 0.288 scrub starts
2017-04-27 03:14:06.054768 osd.41 [INF] 0.288 scrub ok
2017-04-27 03:14:06.055723 osd.41 [INF] 0.288 scrub starts
2017-04-27 03:14:08.055170 osd.41 [INF] 0.288 scrub ok
2017-04-27 03:14:08.056147 osd.41 [INF] 0.288 scrub ok
2017-04-27 03:14:09.494479 mon.0 [INF] pgmap v478721: 1866 pgs: 1866 active+clean; 2280 MB data, 12146 MB used, 46670 GB / 46682 GB avail; 7125 B/s rd, 7 op/s
2017-04-27 03:14:01.446962 osd.9 [INF] 0.77 scrub starts
2017-04-27 03:14:01.447979 osd.9 [INF] 0.77 scrub ok
2017-04-27 03:14:02.447193 osd.9 [INF] 0.77 scrub starts
2017-04-27 03:14:02.447974 osd.9 [INF] 0.77 scrub ok
2017-04-27 03:14:02.845221 osd.35 [INF] 0.592 scrub starts
2017-04-27 03:14:02.846231 osd.35 [INF] 0.592 scrub ok
2017-04-27 03:14:03.051089 osd.25 [INF] 0.1ba scrub starts
2017-04-27 03:14:03.052045 osd.25 [INF] 0.1ba scrub ok
2017-04-27 03:14:04.051274 osd.25 [INF] 0.1ba deep-scrub starts
2017-04-27 03:14:04.052175 osd.25 [INF] 0.1ba deep-scrub ok
2017-04-27 03:14:04.447494 osd.9 [INF] 0.77 scrub starts
2017-04-27 03:14:04.448458 osd.9 [INF] 0.77 scrub ok
2017-04-27 03:14:04.482596 osd.16 [INF] 0.372 scrub starts
2017-04-27 03:14:04.483789 osd.16 [INF] 0.372 scrub ok
2017-04-27 03:14:05.447783 osd.9 [INF] 0.77 scrub starts
2017-04-27 03:14:05.448710 osd.9 [INF] 0.77 scrub ok
2017-04-27 03:14:05.921982 osd.20 [INF] 0.4d4 scrub starts
2017-04-27 03:14:05.924186 osd.20 [INF] 0.4d4 scrub ok
2017-04-27 03:14:06.922353 osd.20 [INF] 0.4d4 scrub starts
2017-04-27 03:14:06.923712 osd.20 [INF] 0.4d4 scrub ok
2017-04-27 03:14:07.483059 osd.16 [INF] 0.372 scrub starts
2017-04-27 03:14:07.484323 osd.16 [INF] 0.372 scrub ok
2017-04-27 03:14:09.052095 osd.25 [INF] 0.1ba scrub starts
2017-04-27 03:14:09.053013 osd.25 [INF] 0.1ba scrub ok
2017-04-27 03:14:09.845661 osd.35 [INF] 0.592 scrub starts
2017-04-27 03:14:09.846608 osd.35 [INF] 0.592 scrub ok
2017-04-27 03:14:10.499706 mon.0 [INF] pgmap v478722: 1866 pgs: 1866 active+clean; 2280 MB data, 12146 MB used, 46670 GB / 46682 GB avail
2017-04-27 03:14:10.922886 osd.20 [INF] 0.4d4 deep-scrub starts
2017-04-27 03:14:10.957724 osd.20 [INF] 0.4d4 deep-scrub ok
2017-04-27 03:14:11.504846 mon.0 [INF] pgmap v478723: 1866 pgs: 1866 active+clean; 2280 MB data, 12146 MB used, 46670 GB / 46682 GB avail
2017-04-27 03:14:12.509848 mon.0 [INF] pgmap v478724: 1866 pgs: 1866 active+clean; 2280 MB data, 12146 MB used, 46670 GB / 46682 GB avail
```

What is inisde?

Starting from top:
- public network switch
- node4
- node3
- node2
- node1
- cluster network switch
- *management node
- *admin network switch

* Management node and admin network is an additional part of ETERNUS CD10000 appliance.



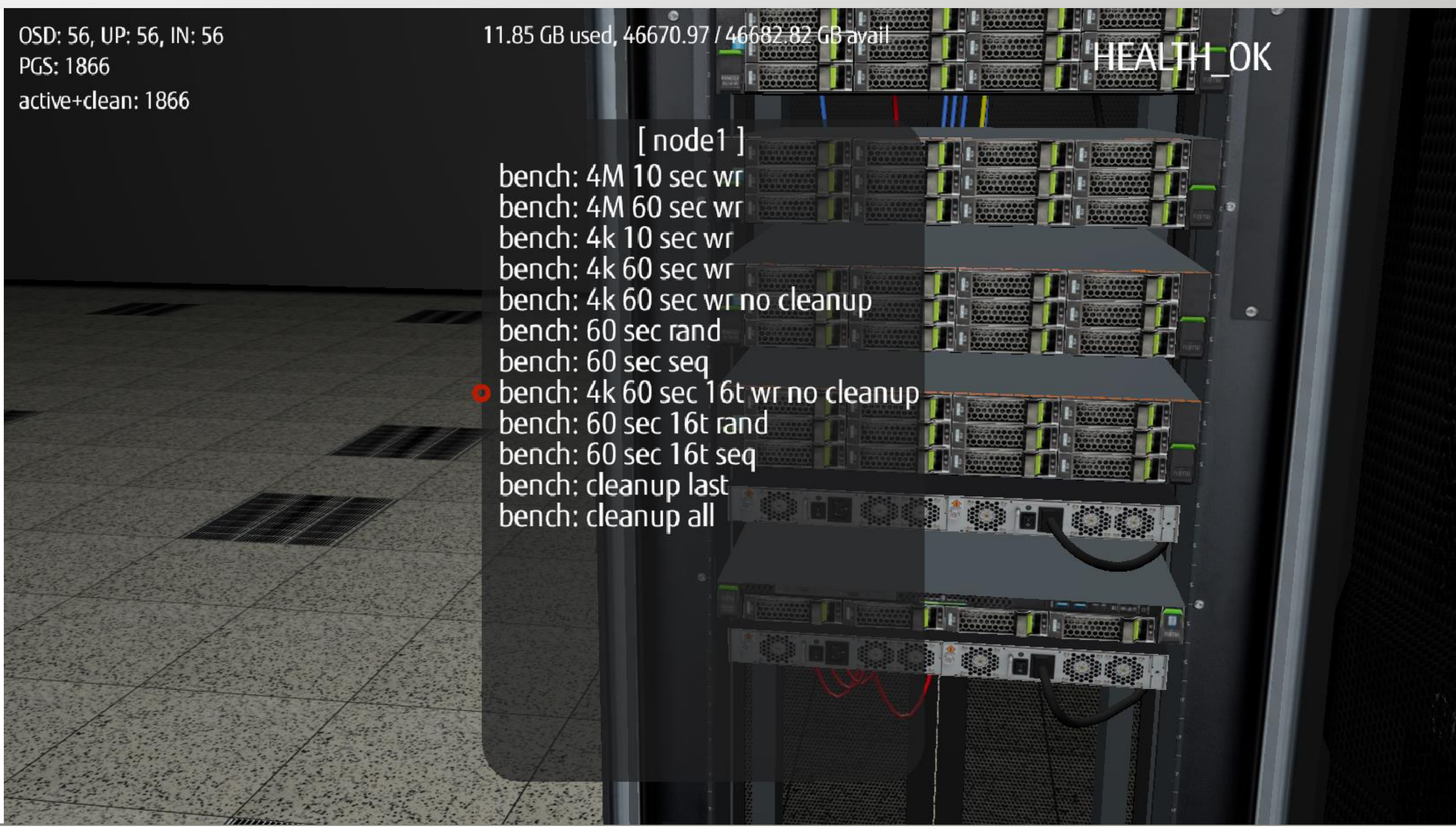
```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
11.85 GB used, 46670.97 / 46682.82 GB avail
```

```
HEALTH_OK
```

Every object has its own menu

Let’s start rados bench from node1.



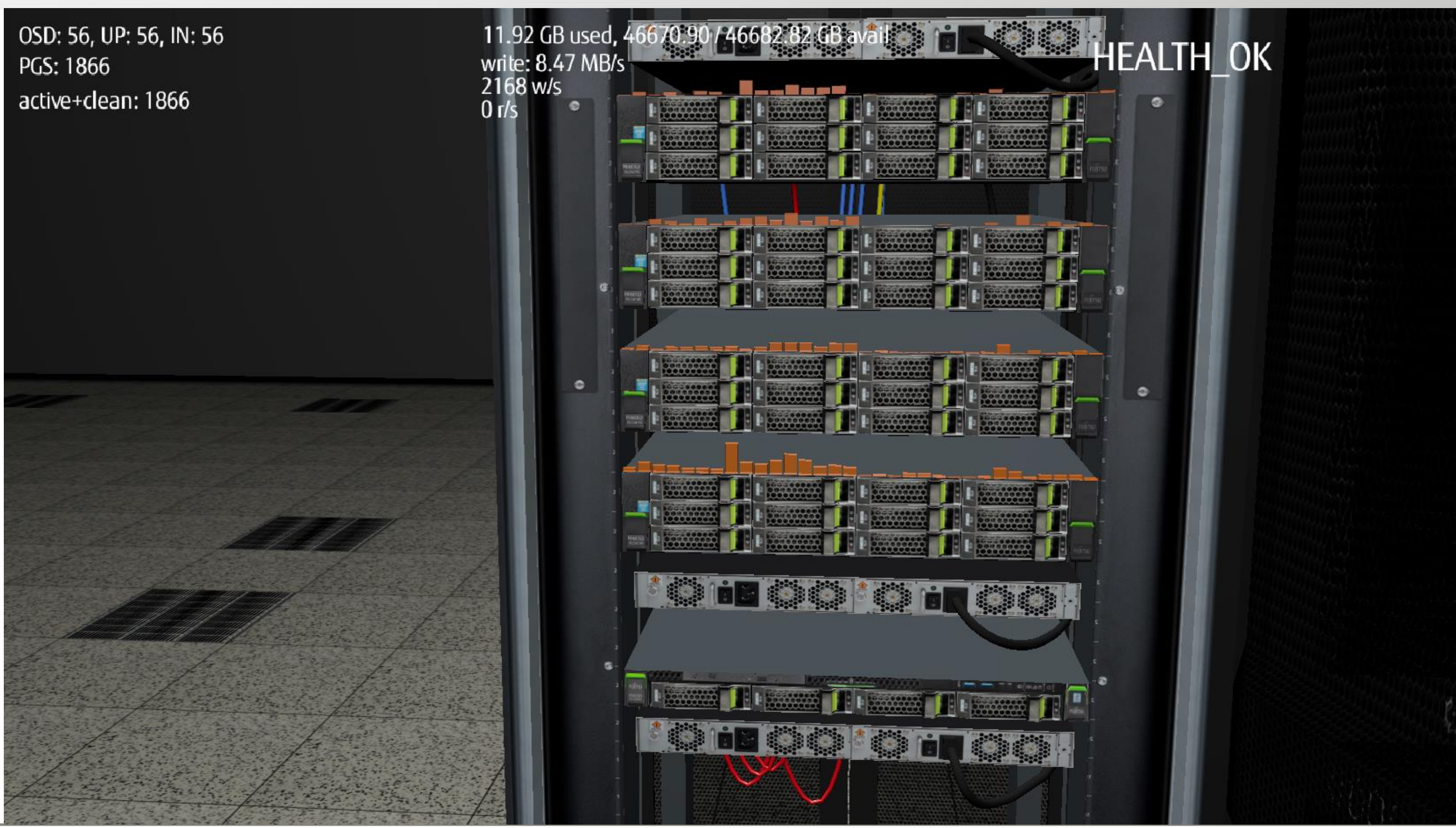
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

11.85 GB used, 46670.97 / 46682.82 GB avail

HEALTH_OK

Logical CPU usage 0-31

Orange blocks are scaling from 0 to 100% just as logical core usage on server after starting rados bench test.



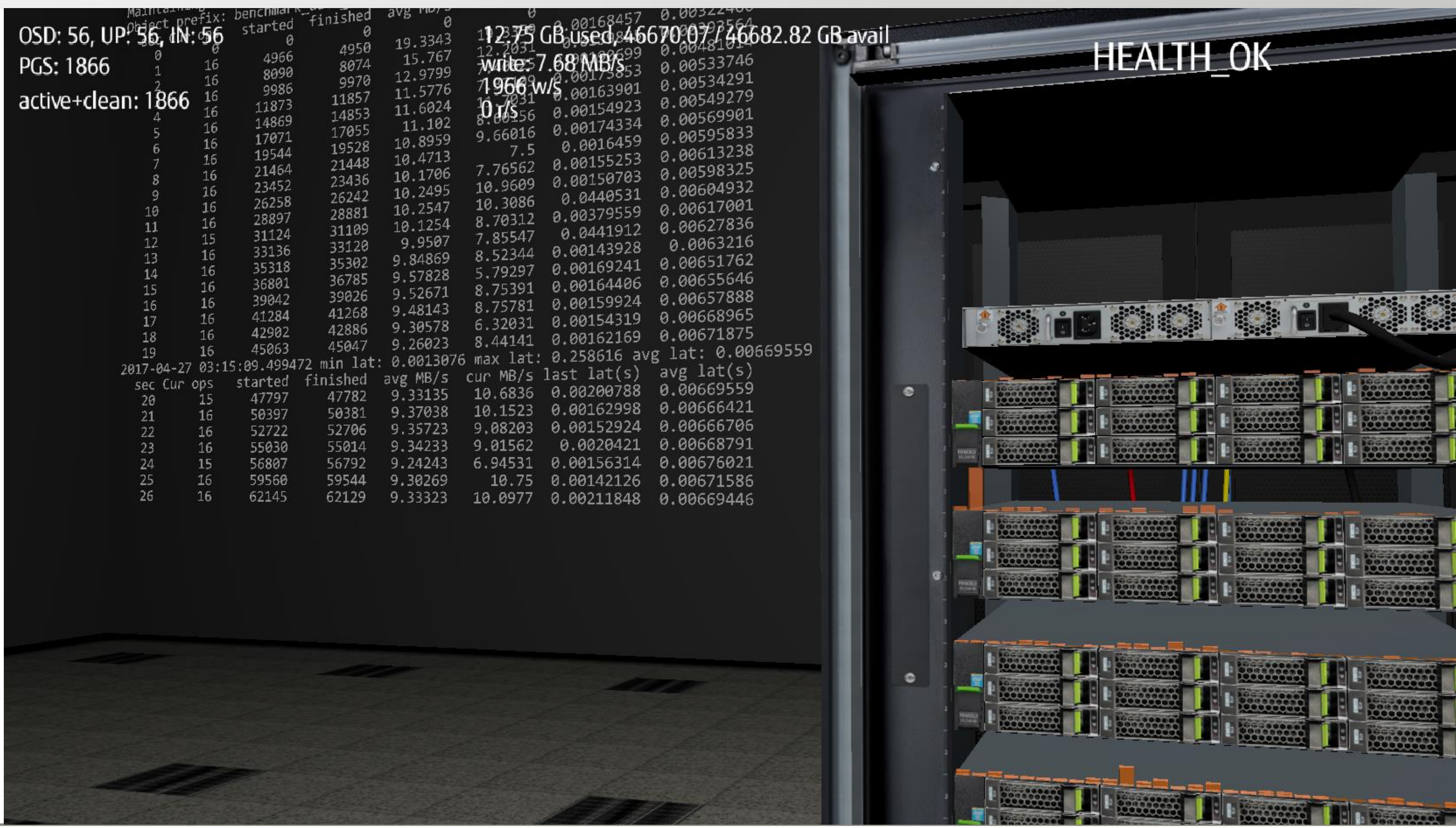
```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
11.92 GB used, 46670.90 / 46682.82 GB avail
write: 8.47 MB/s
2168 w/s
0 r/s
```

HEALTH_OK

Rados bench wall

Current rados bench results are displayed on another wall of 3D server room.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

12.75 GB used, 46670.07 / 46682.82 GB avail
write: 7.68 MB/s
1966 w/s
0 r/s

HEALTH_OK

```
Maintaining[illegible]
Object prefix: benchmark_[illegible]
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
    0       0         0         0         0         0           -           0
    1      16      4966      4950   19.3343 [illegible]  0.00168457  0.00322400
    2      16      8090      8074    15.767   12.2031 [illegible]  0.00302564
    3      16      9986      9970   12.9799 [illegible] [illegible]699  0.00481014
    4      16     11873     11857   11.5776 [illegible]  0.00175853  0.00533746
    5      16     14869     14853   11.6024   11.7031  0.00163901  0.00534291
    6      16     17071     17055    11.102   8.00156  0.00154923  0.00549279
    7      16     19544     19528   10.8959   9.66016  0.00174334  0.00569901
    8      16     21464     21448   10.4713       7.5   0.0016459  0.00595833
    9      16     23452     23436   10.1706   7.76562  0.00155253  0.00613238
   10      16     26258     26242   10.2495   10.9609  0.00150703  0.00598325
   11      16     28897     28881   10.2547   10.3086   0.0440531  0.00604932
   12      15     31124     31109   10.1254   8.70312  0.00379559  0.00617001
   13      16     33136     33120    9.9507   7.85547   0.0441912  0.00627836
   14      16     35318     35302   9.84869   8.52344  0.00143928   0.0063216
   15      16     36801     36785   9.57828   5.79297  0.00169241  0.00651762
   16      16     39042     39026   9.52671   8.75391  0.00164406  0.00655646
   17      16     41284     41268   9.48143   8.75781  0.00159924  0.00657888
   18      16     42902     42886   9.30578   6.32031  0.00154319  0.00668965
   19      16     45063     45047   9.26023   8.44141  0.00162169  0.00671875
2017-04-27 03:15:09.499472 min lat: 0.0013076 max lat: 0.258616 avg lat: 0.00669559
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
   20      15     47797     47782   9.33135   10.6836  0.00200788  0.00669559
   21      16     50397     50381   9.37038   10.1523  0.00162998  0.00666421
   22      16     52722     52706   9.35723   9.08203  0.00152924  0.00666706
   23      16     55030     55014   9.34233   9.01562   0.0020421  0.00668791
   24      15     56807     56792   9.24243   6.94531  0.00156314  0.00676021
   25      16     59560     59544   9.30269     10.75  0.00142126  0.00671586
   26      16     62145     62129   9.33323   10.0977  0.00211848  0.00669446
```

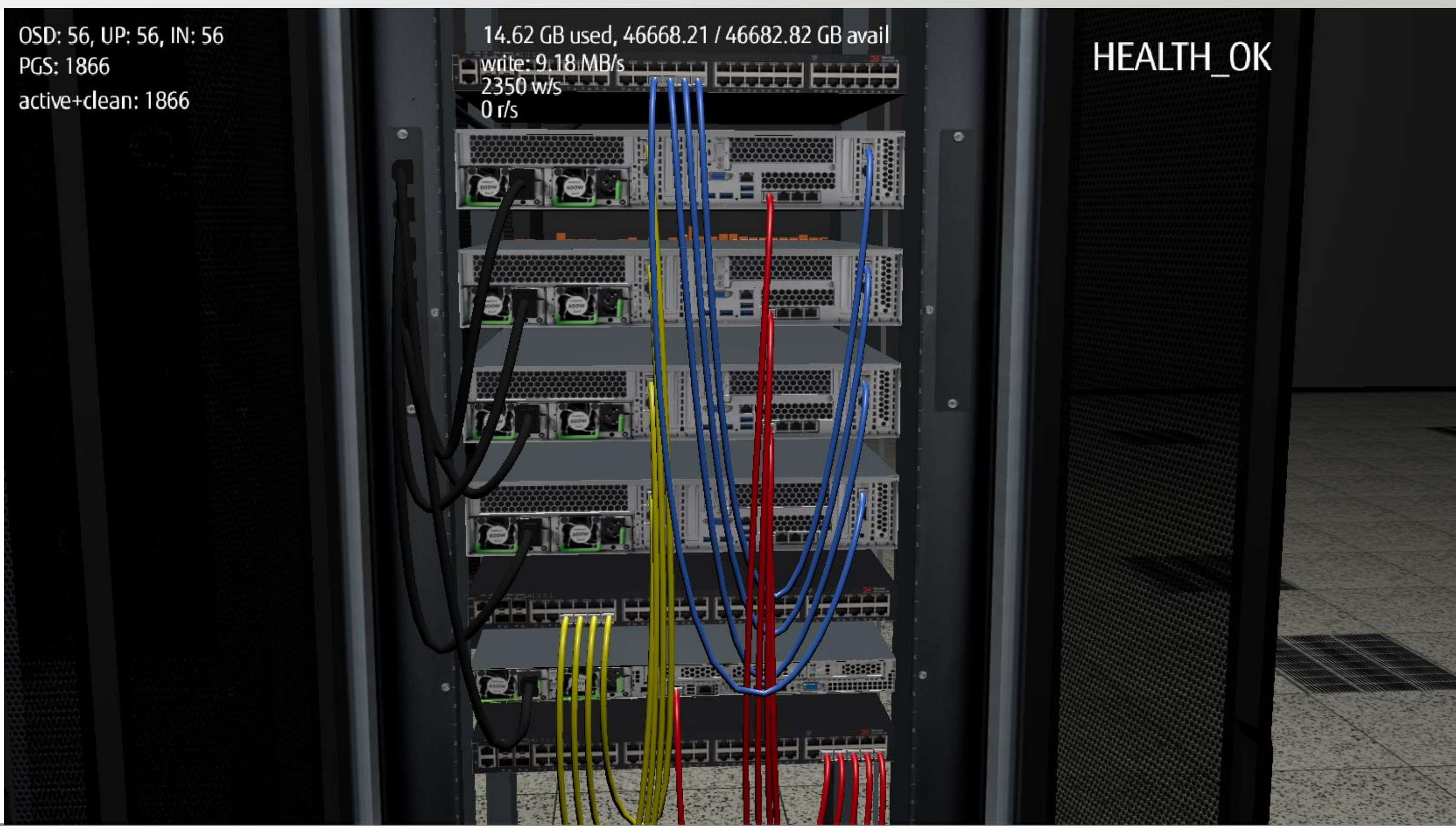
Let’s look on the back side

Network cabling:

Blue – public

Yellow – cluster

Red – admin



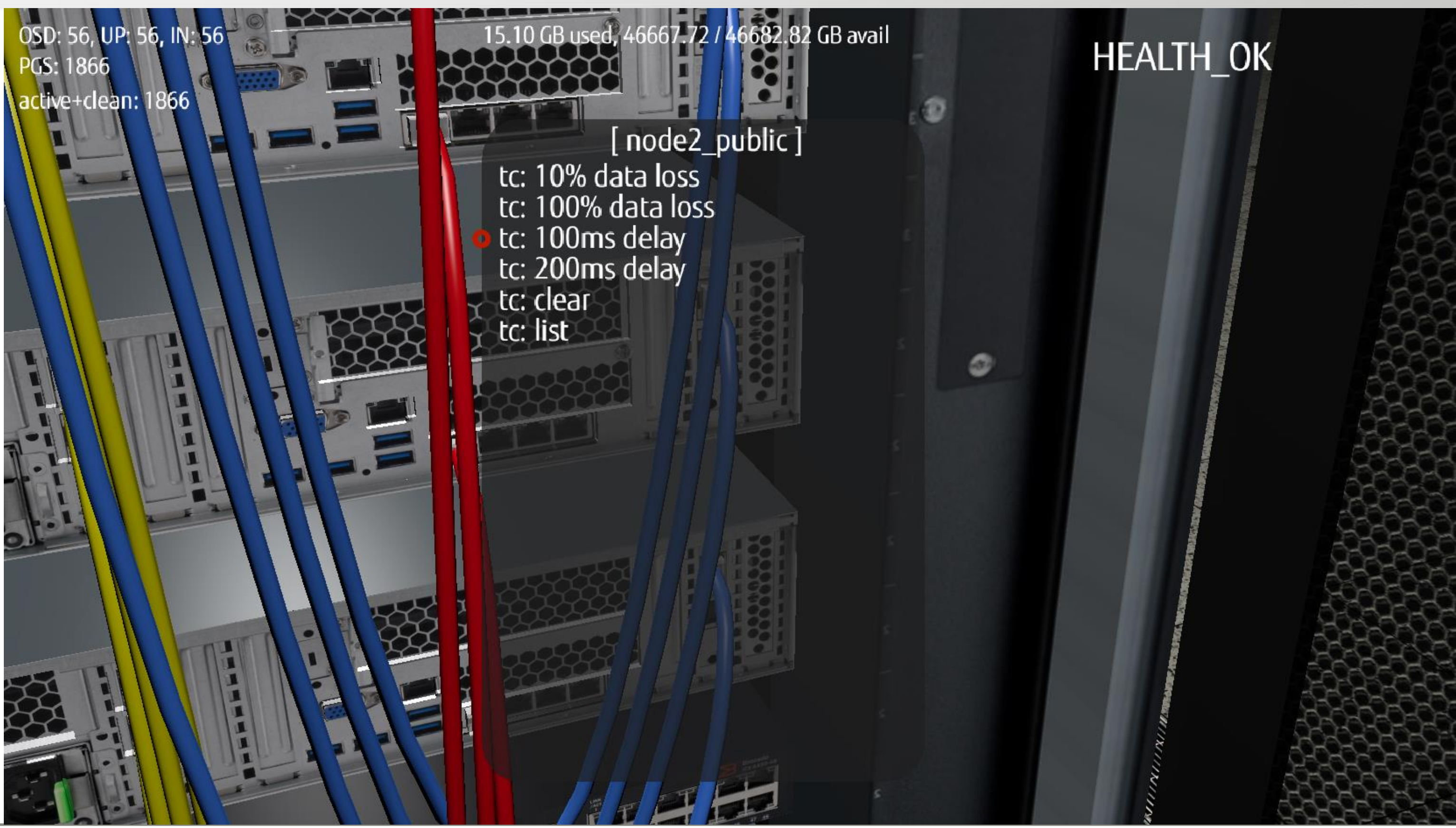
```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
14.62 GB used, 46668.21 / 46682.82 GB avail
write: 9.18 MB/s
2350 w/s
0 r/s
```

```
HEALTH_OK
```

First fireball – public network delay

Add 100ms delay for public interface of node2.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

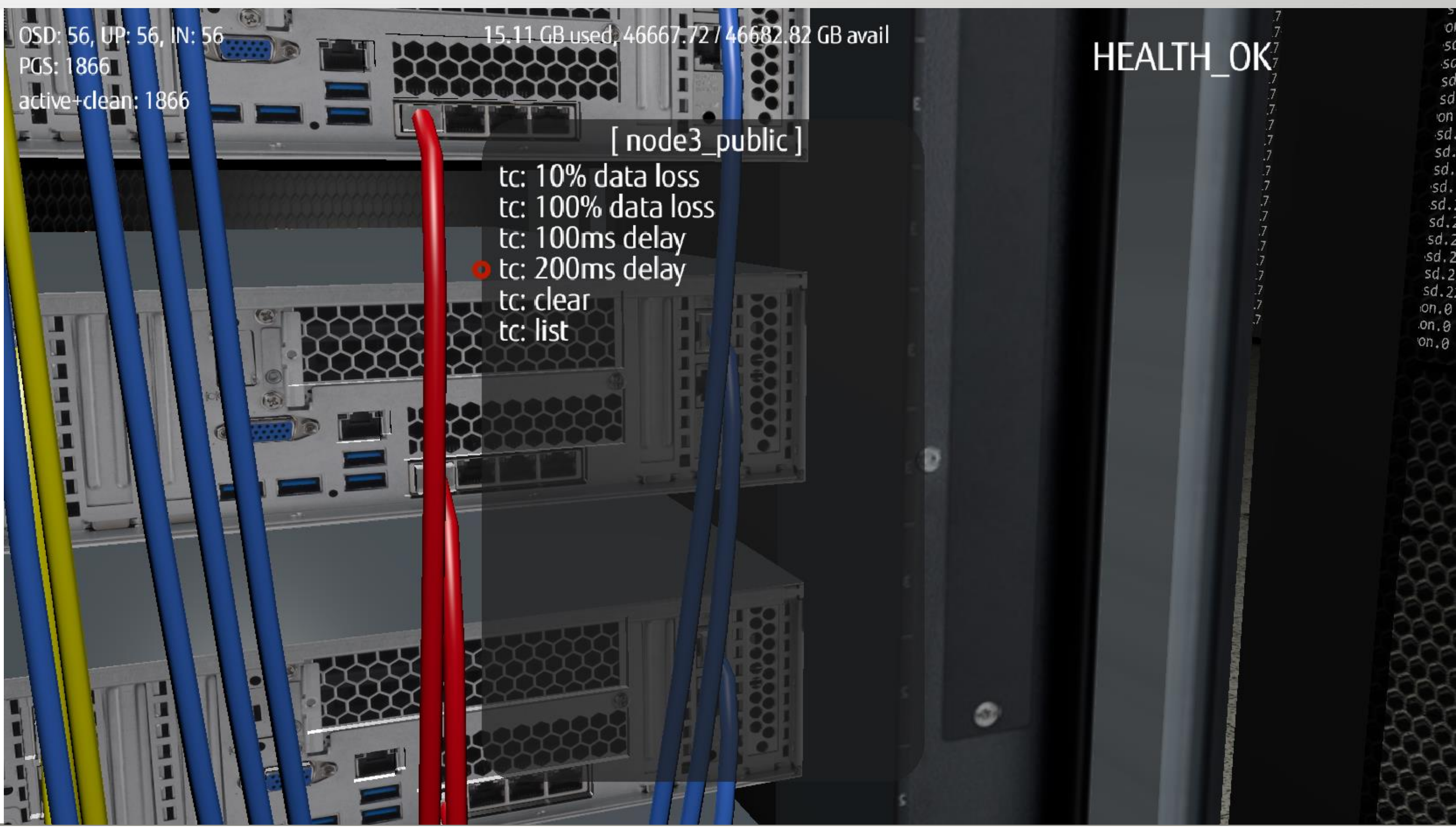
15.10 GB used, 46667.72 / 46682.82 GB avail

HEALTH_OK

[node2_public]
tc: 10% data loss
tc: 100% data loss
○ tc: 100ms delay
tc: 200ms delay
tc: clear
tc: list

First fireball – public network delay

Add another delay, this time 200ms for public interface of node3.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

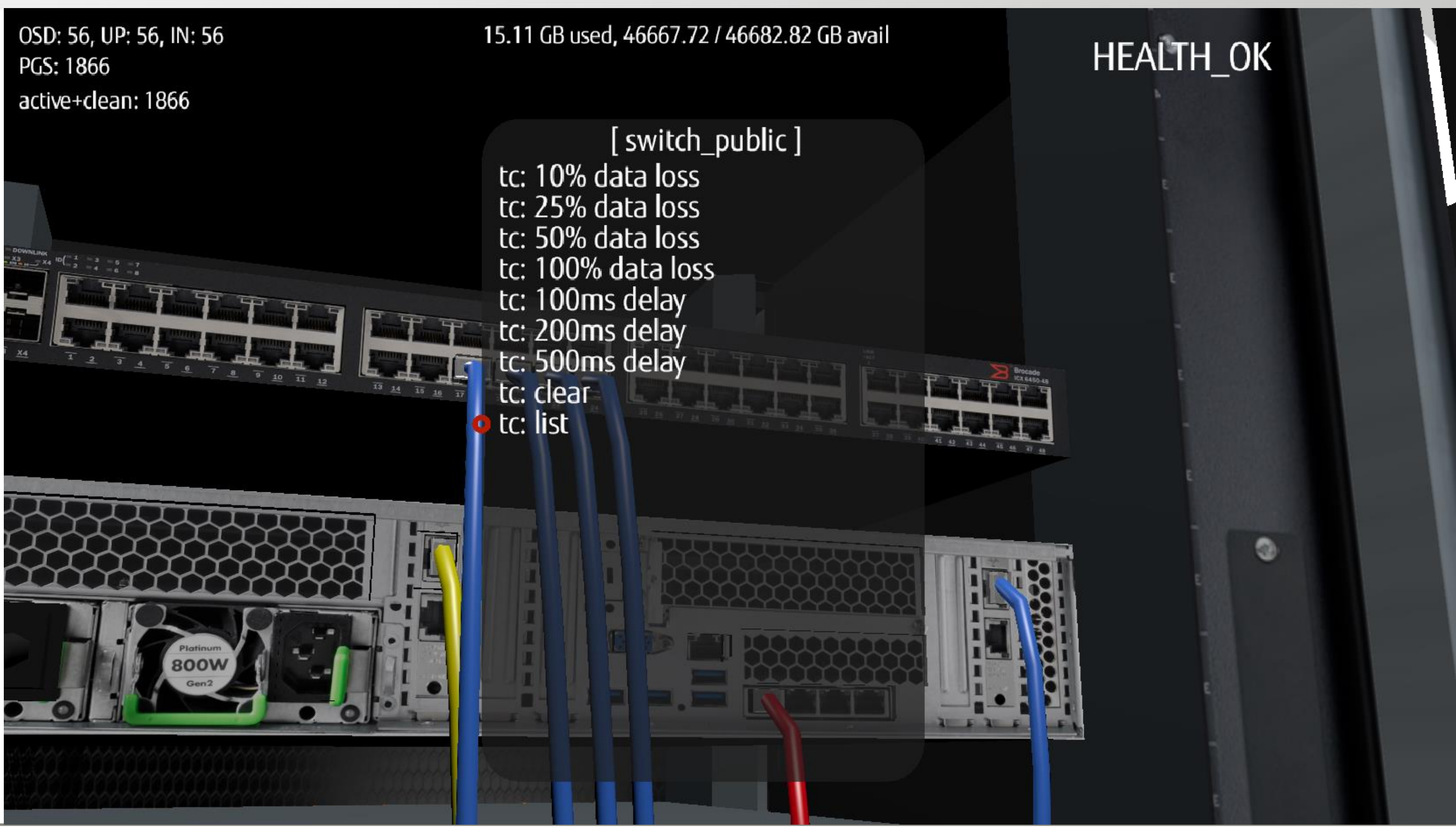
15.11 GB used, 46667.72 / 46682.82 GB avail

HEALTH_OK

[node3_public]
- tc: 10% data loss
- tc: 100% data loss
- tc: 100ms delay
- ● tc: 200ms delay
- tc: clear
- tc: list

First fireball – public network delay

Have a global look on what we’ve set on the public network switch.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

15.11 GB used, 46667.72 / 46682.82 GB avail

HEALTH_OK

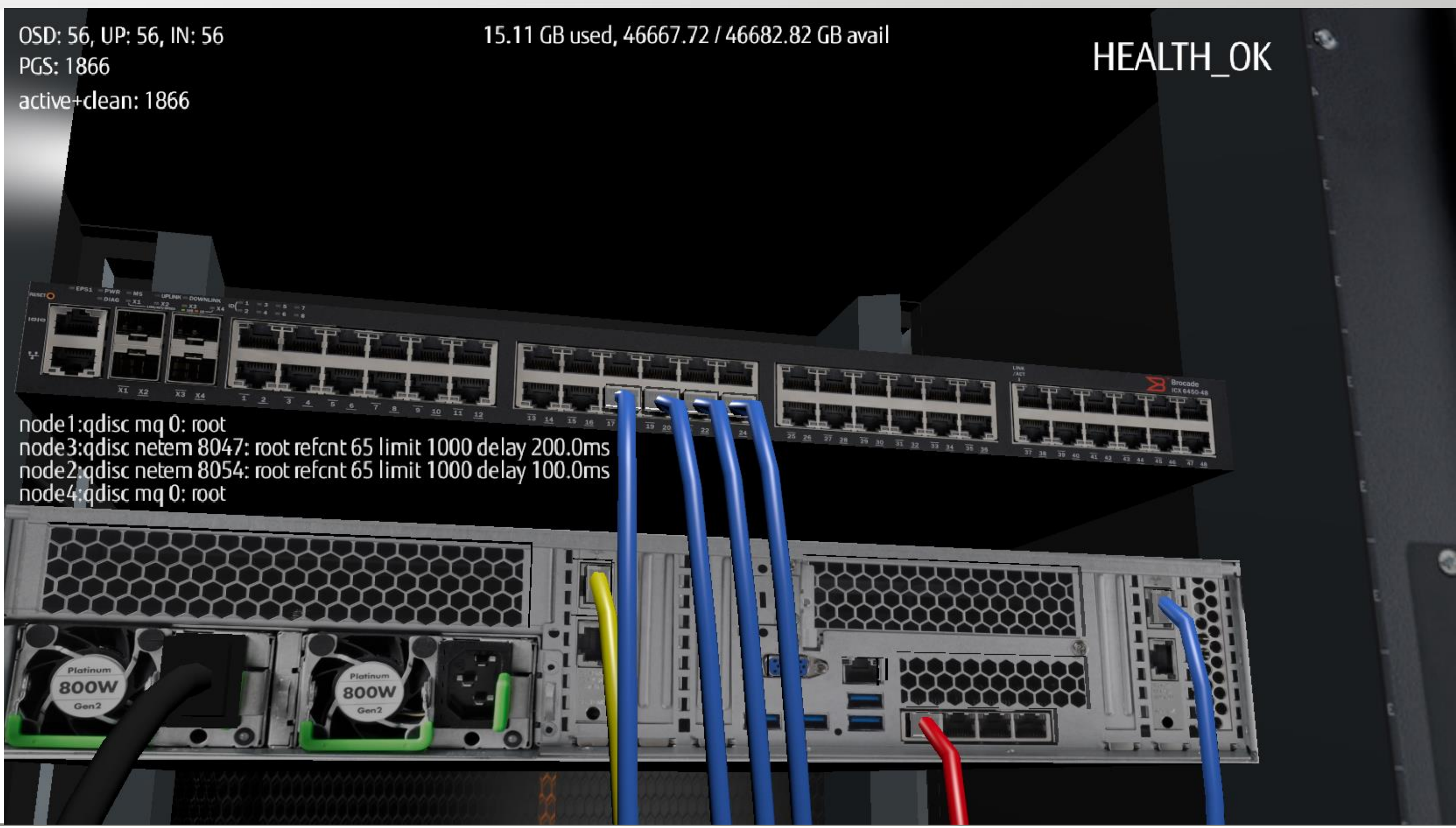
[switch_public]
tc: 10% data loss
tc: 25% data loss
tc: 50% data loss
tc: 100% data loss
tc: 100ms delay
tc: 200ms delay
tc: 500ms delay
tc: clear
tc: list

First fireball – public network delay

Public network delays:

node3: 200ms

node2: 100ms

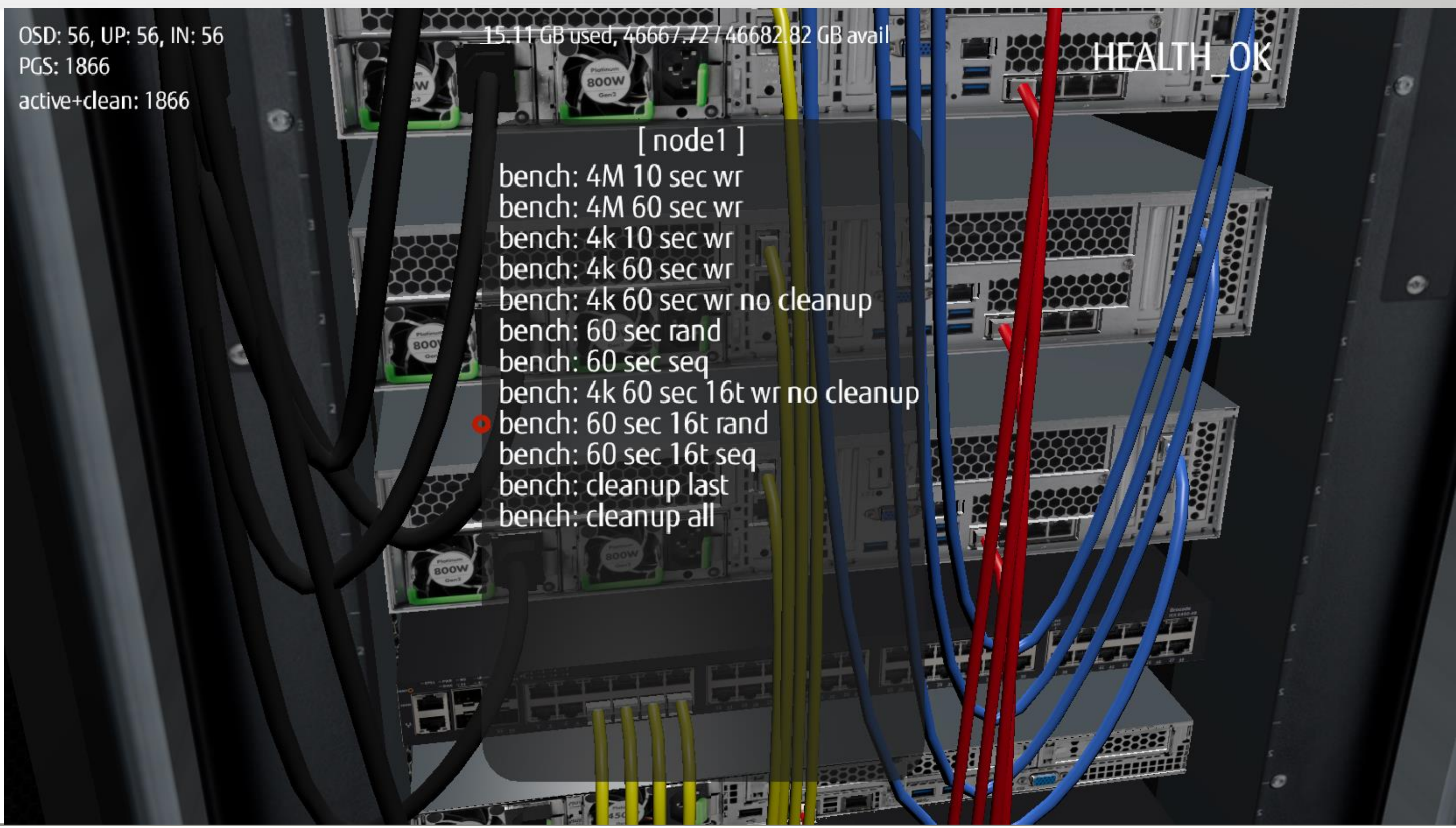


```
OSD: 56, UP: 56, IN: 56          15.11 GB used, 46667.72 / 46682.82 GB avail          HEALTH_OK
PGS: 1866
active+clean: 1866
```

```
node1:qdisc mq 0: root
node3:qdisc netem 8047: root refcnt 65 limit 1000 delay 200.0ms
node2:qdisc netem 8054: root refcnt 65 limit 1000 delay 100.0ms
node4:qdisc mq 0: root
```

First fireball – public network delay

Since we slighty broke the public network, let’s run a read test.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

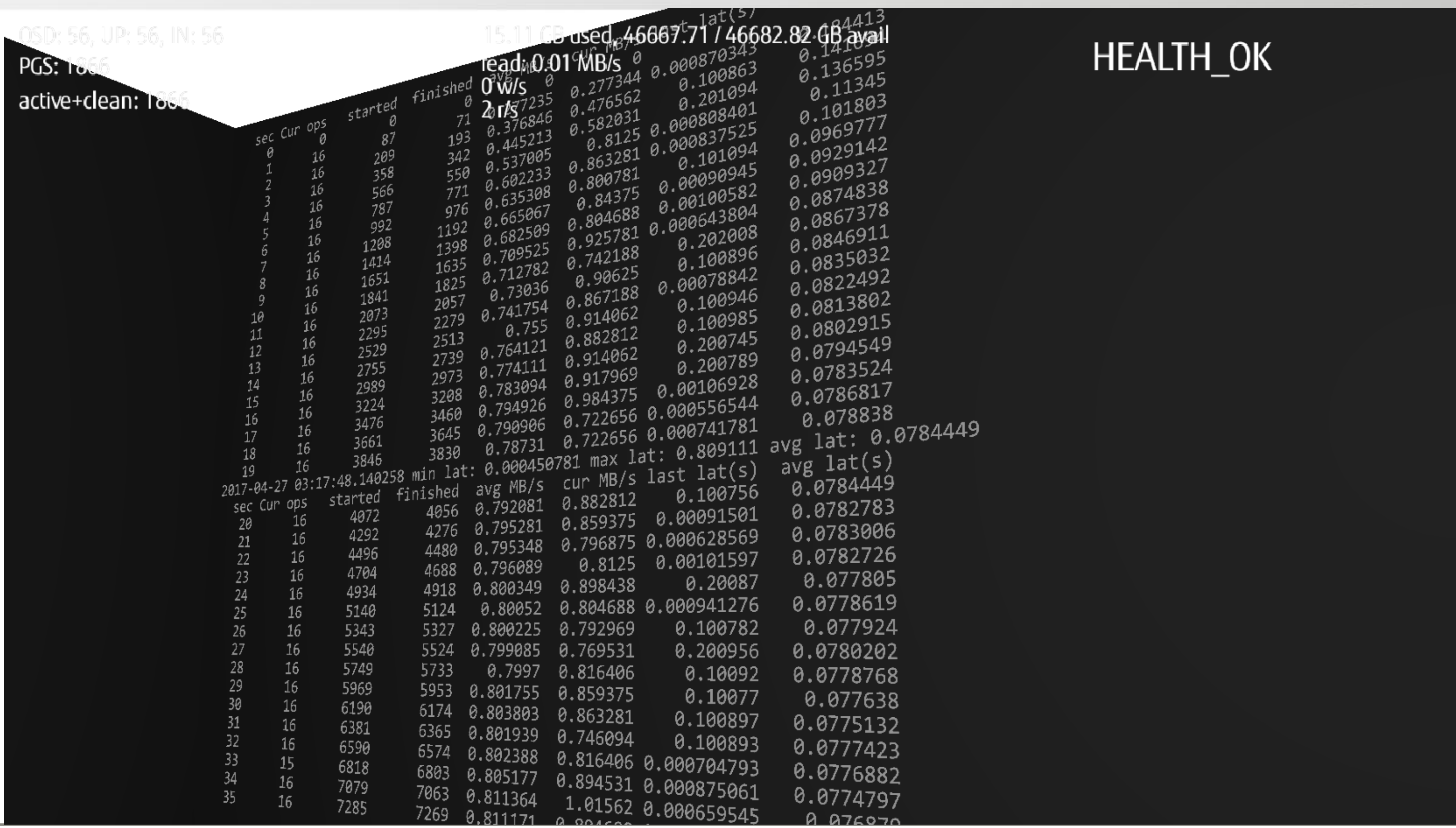
15.11 GB used, 46667.72 / 46682.82 GB avail

HEALTH_OK

First fireball – public network delay

Look closely on the latency, which sometimes is very low, but sometimes reaches above 100ms and 200ms.

These are the values we have set as delays of node2 and node3.

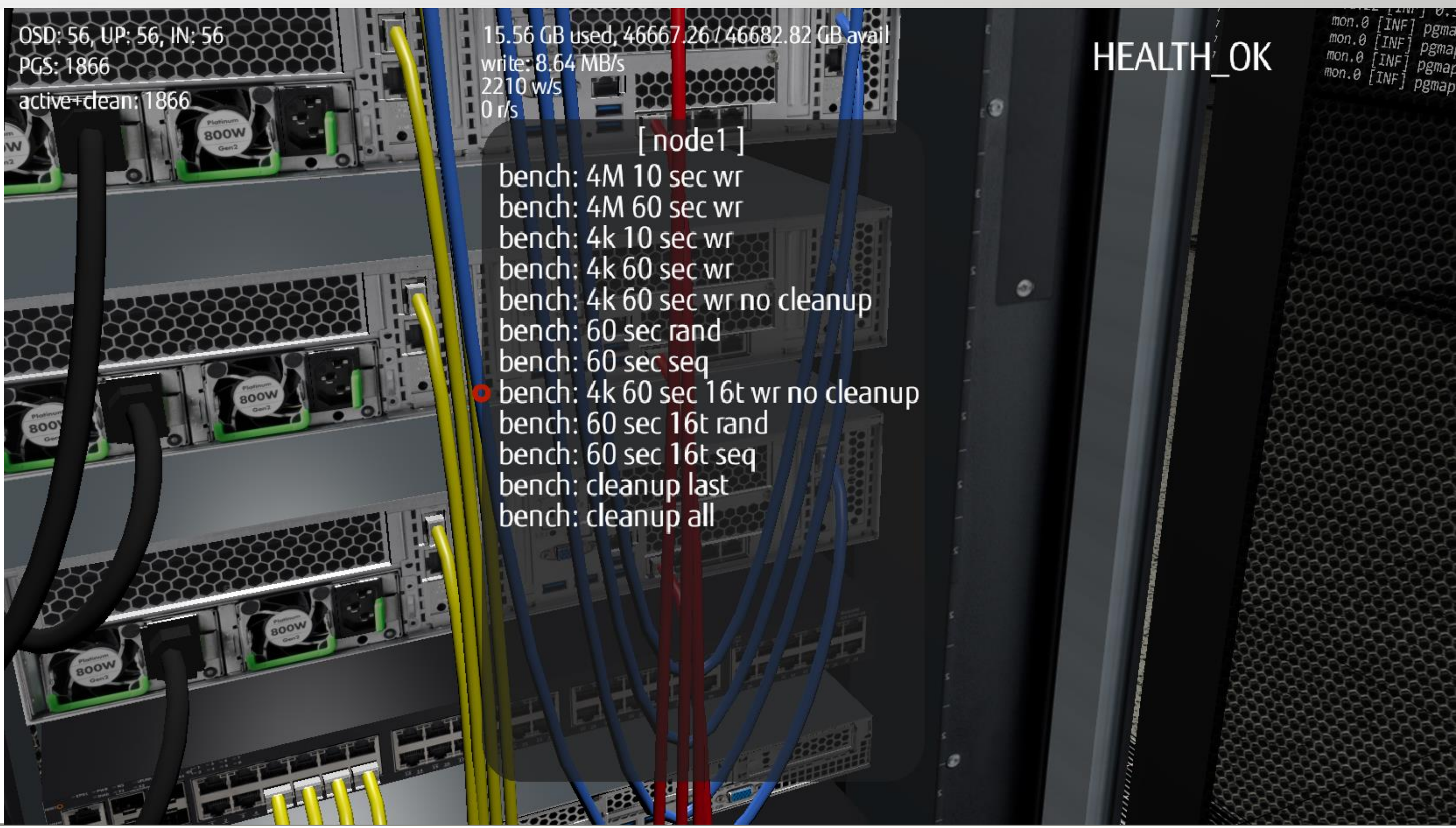


```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

15.11 GB used, 46667.71 / 46682.82 GB avail
read: 0.01 MB/s
0 w/s
2 r/s

HEALTH_OK
```

```
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
    0       0         0         0         0         0           -           0
    1      16        87        71  0.277235  0.277344 0.000870343    0.184413
    2      16       209       193  0.376846  0.476562    0.100863     0.14109
    3      16       358       342  0.445213  0.582031    0.201094    0.136595
    4      16       566       550  0.537005    0.8125 0.000808401     0.11345
    5      16       787       771  0.602233  0.863281 0.000837525    0.101803
    6      16       992       976  0.635308  0.800781    0.101094   0.0969777
    7      16      1208      1192  0.665067   0.84375  0.00090945   0.0929142
    8      16      1414      1398  0.682509  0.804688  0.00100582   0.0909327
    9      16      1651      1635  0.709525  0.925781 0.000643804   0.0874838
   10      16      1841      1825  0.712782  0.742188    0.202008   0.0867378
   11      16      2073      2057   0.73036   0.90625    0.100896   0.0846911
   12      16      2295      2279  0.741754  0.867188  0.00078842   0.0835032
   13      16      2529      2513     0.755  0.914062    0.100946   0.0822492
   14      16      2755      2739  0.764121  0.882812    0.100985   0.0813802
   15      16      2989      2973  0.774111  0.914062    0.200745   0.0802915
   16      16      3224      3208  0.783094  0.917969    0.200789   0.0794549
   17      16      3476      3460  0.794926  0.984375  0.00106928   0.0783524
   18      16      3661      3645  0.790906  0.722656 0.000556544   0.0786817
   19      16      3846      3830   0.78731  0.722656 0.000741781    0.078838
2017-04-27 03:17:48.140258 min lat: 0.000450781 max lat: 0.809111 avg lat: 0.0784449
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
   20      16      4072      4056  0.792081  0.882812    0.100756   0.0784449
   21      16      4292      4276  0.795281  0.859375  0.00091501   0.0782783
   22      16      4496      4480  0.795348  0.796875 0.000628569   0.0783006
   23      16      4704      4688  0.796089    0.8125  0.00101597   0.0782726
   24      16      4934      4918  0.800349  0.898438     0.20087    0.077805
   25      16      5140      5124   0.80052  0.804688 0.000941276   0.0778619
   26      16      5343      5327  0.800225  0.792969    0.100782    0.077924
   27      16      5540      5524  0.799085  0.769531    0.200956   0.0780202
   28      16      5749      5733    0.7997  0.816406     0.10092   0.0778768
   29      16      5969      5953  0.801755  0.859375     0.10077    0.077638
   30      16      6190      6174  0.803803  0.863281    0.100897   0.0775132
   31      16      6381      6365  0.801939  0.746094    0.100893   0.0777423
   32      16      6590      6574  0.802388  0.816406 0.000704793   0.0776882
   33      15      6818      6803  0.805177  0.894531 0.000875061   0.0774797
   34      16      7079      7063  0.811364   1.01562 0.000659545    0.076879
   35      16      7285      7269  0.811171
```

Second fireball – kill one of cluster network interfaces

Start rados bench first to see what is going to happen after we kill one network card.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

15.56 GB used, 46667.26 / 46682.82 GB avail
write: 8.64 MB/s
2210 w/s
0 r/s

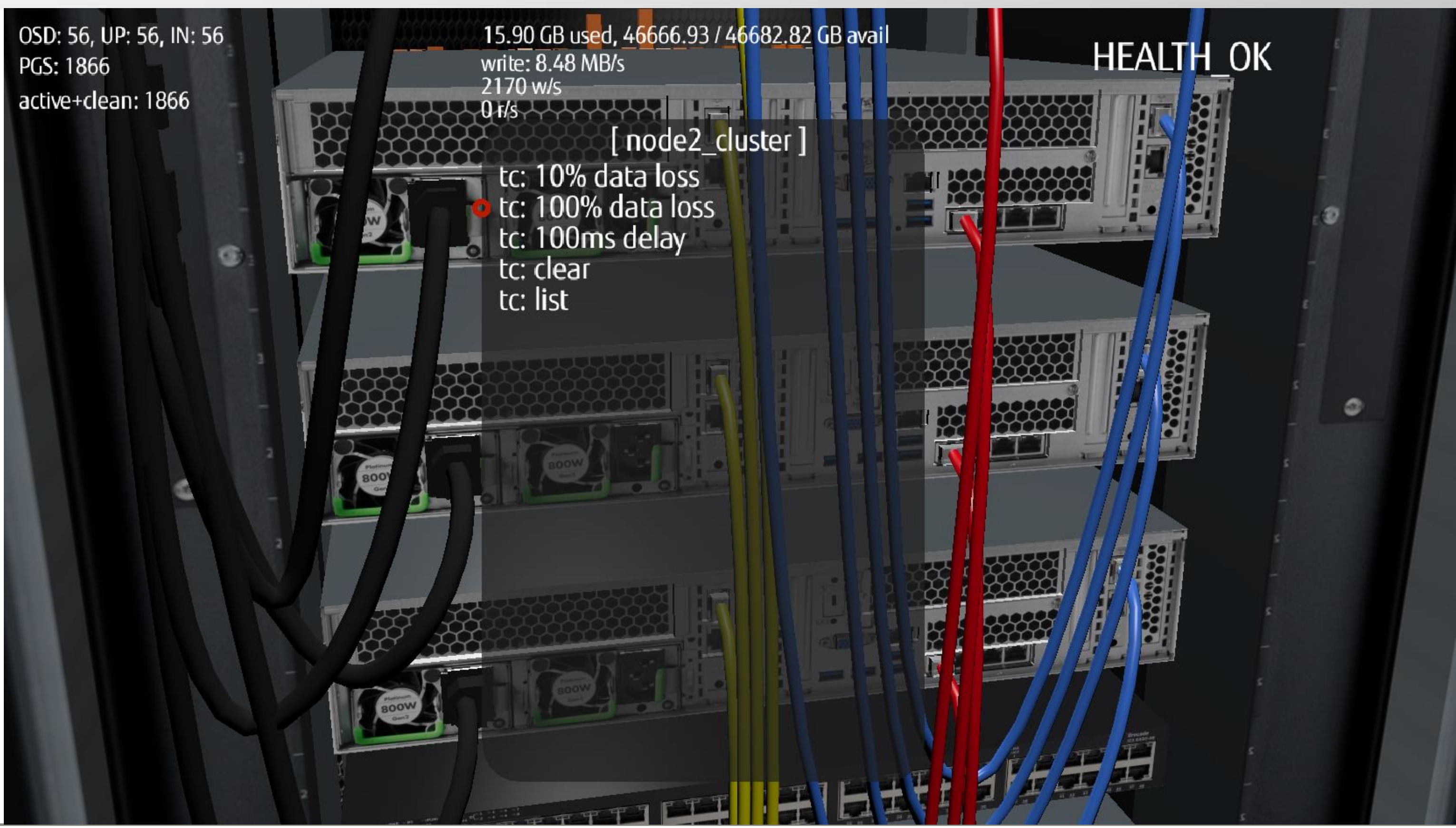
HEALTH_OK

```
mon.0 [INF] pgma
mon.0 [INF] pgmap
mon.0 [INF] pgmap
mon.0 [INF] pgmap
```

[node1]
bench: 4M 10 sec wr
bench: 4M 60 sec wr
bench: 4k 10 sec wr
bench: 4k 60 sec wr
bench: 4k 60 sec wr no cleanup
bench: 60 sec rand
bench: 60 sec seq
bench: 4k 60 sec 16t wr no cleanup
bench: 60 sec 16t rand
bench: 60 sec 16t seq
bench: cleanup last
bench: cleanup all

Second fireball – kill one of cluster network interfaces

Adding 100% data loss to a network interface could simulate e.g. NIC overheat.



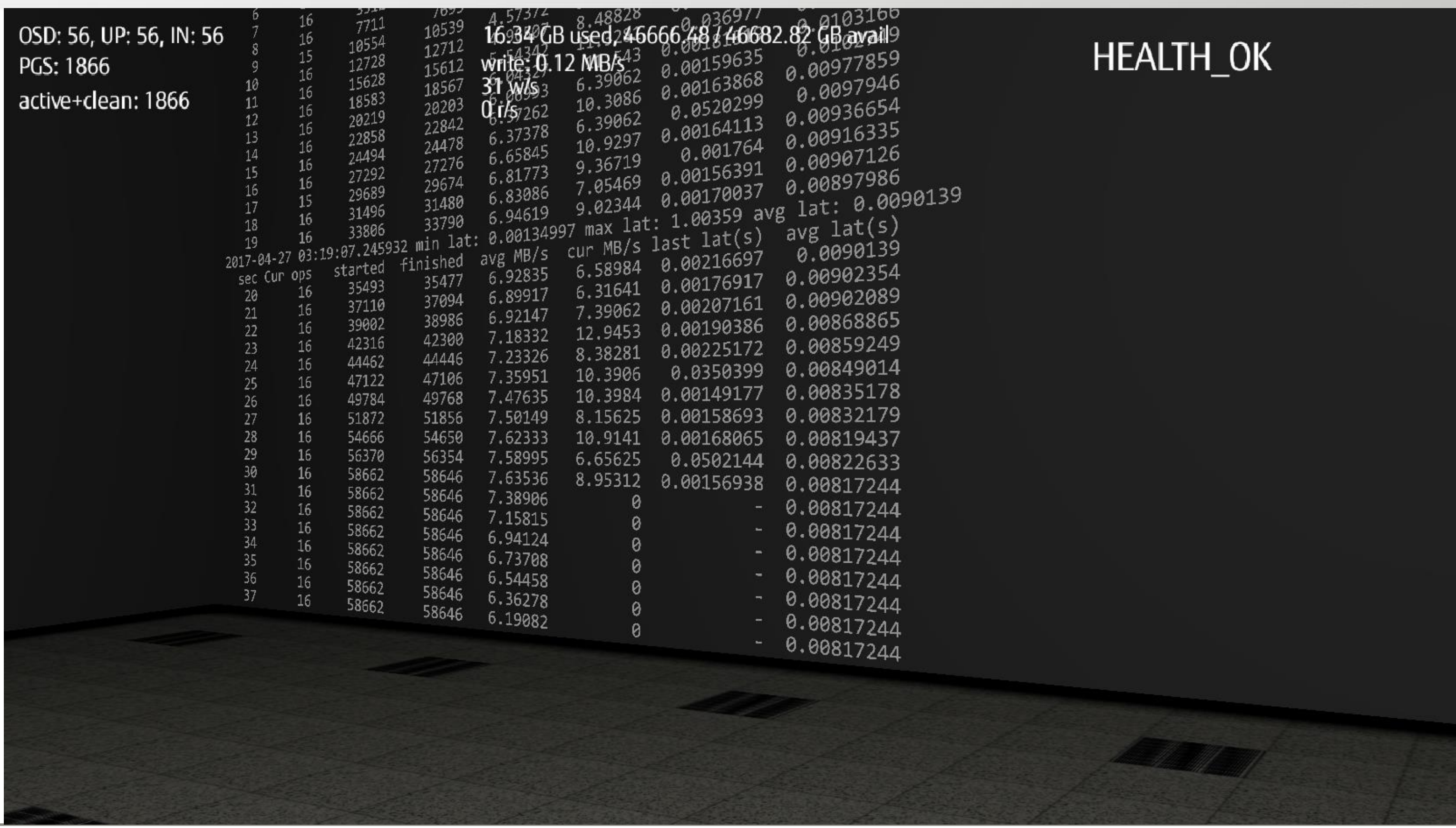
```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
15.90 GB used, 46666.93 / 46682.82 GB avail
write: 8.48 MB/s
2170 w/s
0 r/s
```

HEALTH_OK

Second fireball – kill one of cluster network interfaces

Writes are now blocked, because there is no communication via cluster network interface of node2.

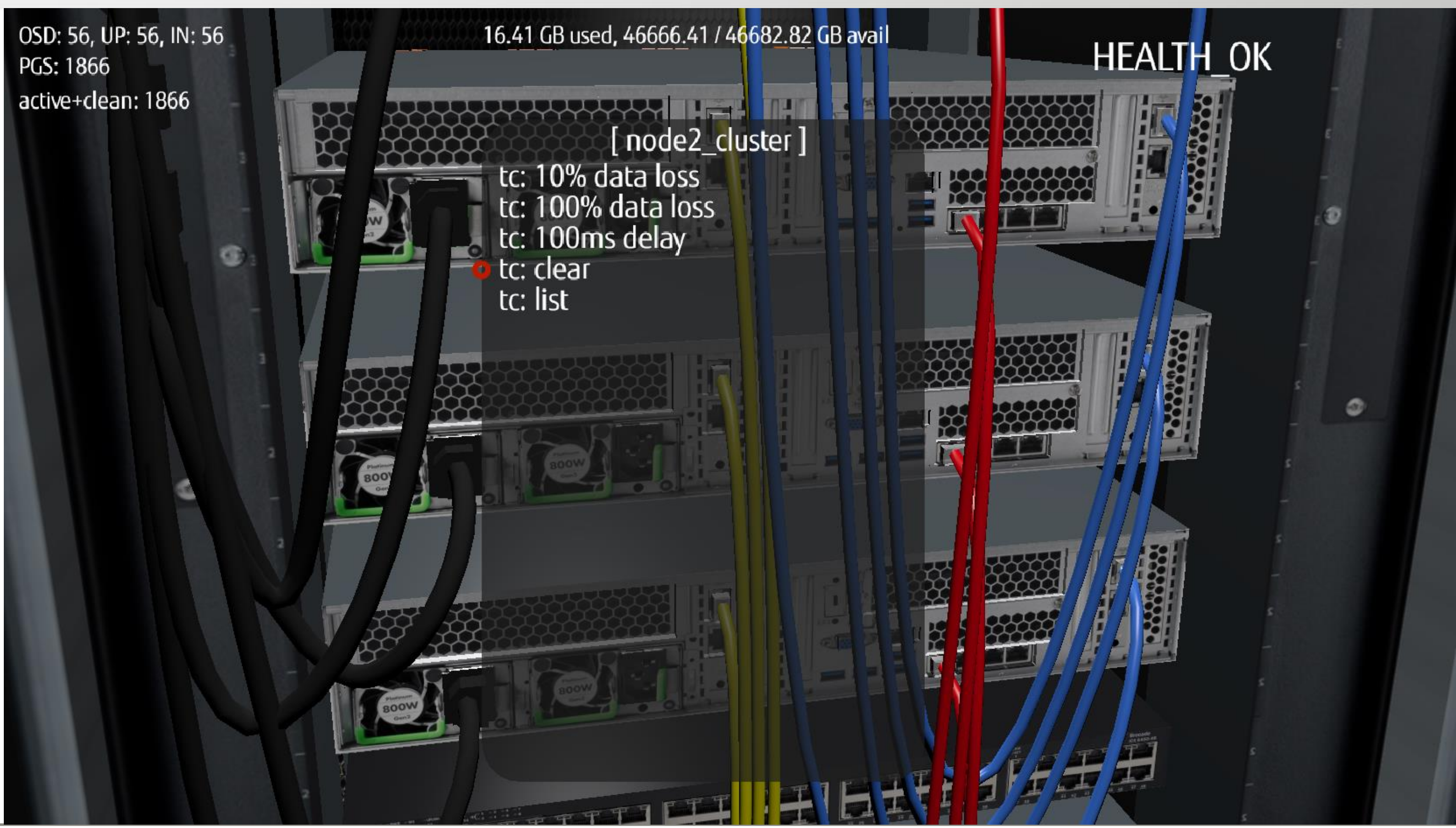


```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

16.34 GB used, 46666.48 / 46682.82 GB avail
write: 0.12 MB/s
31 w/s
0 r/s

HEALTH_OK
```

```
    6      16     [illegible]     [illegible]
    7      16     [illegible]     7695     [illegible]     [illegible]
    8      16      7711    10539   4.57372   [illegible]     [illegible]
    9      15     10554    12712   [illegible]   8.48828     [illegible]     [illegible]
   10      16     12728    15612   5.54342   [illegible]   0.036977     [illegible]
   11      16     15628    18567   6.04327   [illegible]   [illegible]   0.0103166
   12      16     18583    20203   [illegible]   6.39062   0.00159635   [illegible]
   13      16     20219    22842   [illegible]   10.3086   0.00163868   0.00977859
   14      16     22858    24478   6.37378   6.39062   0.0520299   0.0097946
   15      16     24494    27276   6.65845   10.9297   0.00164113   0.00936654
   16      16     27292    29674   6.81773   9.36719   0.001764   0.00916335
   17      15     29689    31480   6.83086   7.05469   0.00156391   0.00907126
   18      16     31496    33790   6.94619   9.02344   0.00170037   0.00897986
   19      16     33806
2017-04-27 03:19:07.245932 min lat: 0.00134997 max lat: 1.00359 avg lat: 0.0090139
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
   20      16     35493     35477   6.92835   6.58984  0.00216697   0.0090139
   21      16     37110     37094   6.89917   6.31641  0.00176917  0.00902354
   22      16     39002     38986   6.92147   7.39062  0.00207161  0.00902089
   23      16     42316     42300   7.18332   12.9453  0.00190386  0.00868865
   24      16     44462     44446   7.23326   8.38281  0.00225172  0.00859249
   25      16     47122     47106   7.35951   10.3906   0.0350399  0.00849014
   26      16     49784     49768   7.47635   10.3984  0.00149177  0.00835178
   27      16     51872     51856   7.50149   8.15625  0.00158693  0.00832179
   28      16     54666     54650   7.62333   10.9141  0.00168065  0.00819437
   29      16     56370     56354   7.58995   6.65625   0.0502144  0.00822633
   30      16     58662     58646   7.63536   8.95312  0.00156938  0.00817244
   31      16     58662     58646   7.38906         0           -  0.00817244
   32      16     58662     58646   7.15815         0           -  0.00817244
   33      16     58662     58646   6.94124         0           -  0.00817244
   34      16     58662     58646   6.73708         0           -  0.00817244
   35      16     58662     58646   6.54458         0           -  0.00817244
   36      16     58662     58646   6.36278         0           -  0.00817244
   37      16     58662     58646   6.19082         0           -  0.00817244
```

Second fireball – kill one of cluster network interfaces

Ok, it is enough, clear 100% packet drop on this interface and let's check if writes to the cluster will start working again.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

16.41 GB used, 46666.41 / 46682.82 GB avail

HEALTH_OK

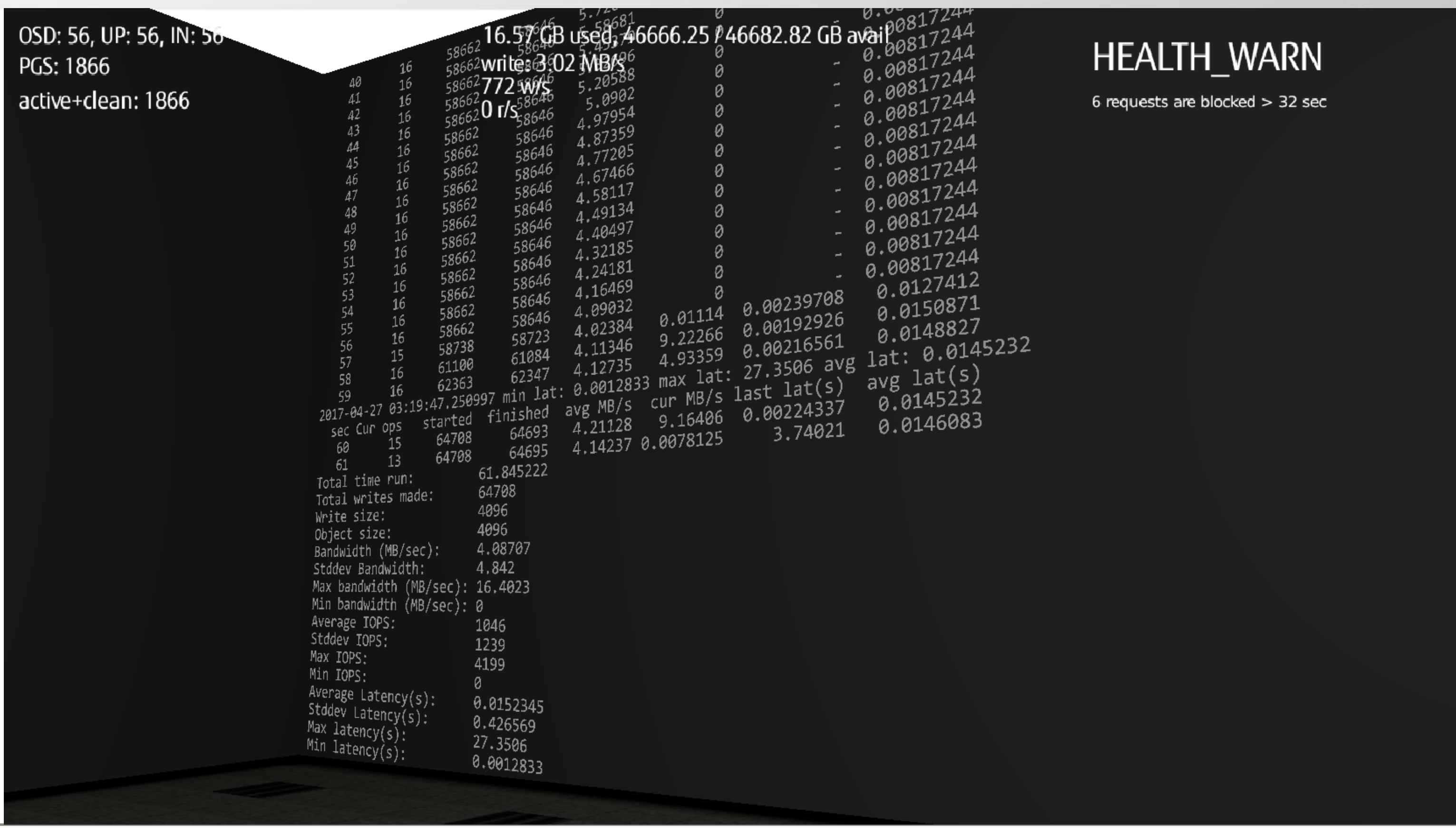
[node2_cluster]
tc: 10% data loss
tc: 100% data loss
tc: 100ms delay
● tc: clear
tc: list

Second fireball – kill one of cluster network interfaces

At the end of rados bench log, you can see that writes were unblocked.

Cluster went into the HEALTH_WARN state, and reported slow requests.

Probably some threads were constantly waiting for the cluster network response to finish write operation.

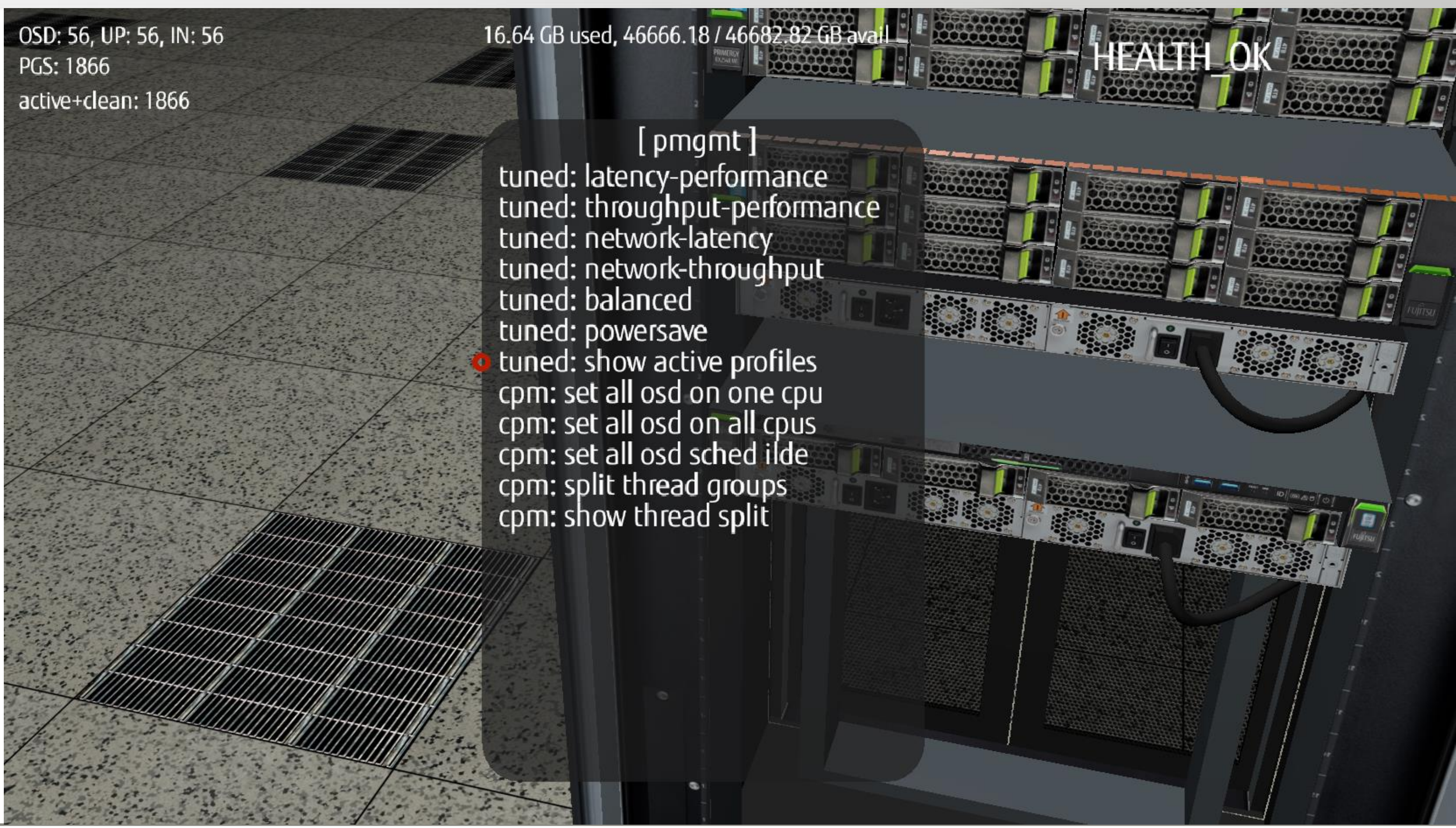


```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

16.57 GB used, 46666.25 / 46682.82 GB avail
write: 3.02 MB/s
772 w/s
0 r/s

HEALTH_WARN
6 requests are blocked > 32 sec
```

```
   40      16     58662     58646      5.72           0           -  0.00817244
   41      16     58662     58646   5.58681           0           -  0.00817244
   42      16     58662     58646                     0           -  0.00817244
   43      16     58662     58646                     0           -  0.00817244
   44      16     58662     58646   5.20588           0           -  0.00817244
   45      16     58662     58646    5.0902           0           -  0.00817244
   46      16     58662     58646   4.97954           0           -  0.00817244
   47      16     58662     58646   4.87359           0           -  0.00817244
   48      16     58662     58646   4.77205           0           -  0.00817244
   49      16     58662     58646   4.67466           0           -  0.00817244
   50      16     58662     58646   4.58117           0           -  0.00817244
   51      16     58662     58646   4.49134           0           -  0.00817244
   52      16     58662     58646   4.40497           0           -  0.00817244
   53      16     58662     58646   4.32185           0           -  0.00817244
   54      16     58662     58646   4.24181           0           -  0.00817244
   55      16     58662     58646   4.16469           0           -  0.00817244
   56      16     58662     58646   4.09032           0           -  0.00817244
   57      15     58738     58723   4.02384     0.01114  0.00239708   0.0127412
   58      16     61100     61084   4.11346     9.22266  0.00192926   0.0150871
   59      16     62363     62347   4.12735     4.93359  0.00216561   0.0148827
2017-04-27 03:19:47.250997 min lat: 0.0012833 max lat: 27.3506 avg lat: 0.0145232
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
   60      15     64708     64693   4.21128     9.16406  0.00224337   0.0145232
   61      13     64708     64695   4.14237   0.0078125     3.74021   0.0146083
Total time run:         61.845222
Total writes made:      64708
Write size:             4096
Object size:            4096
Bandwidth (MB/sec):     4.08707
Stddev Bandwidth:       4.842
Max bandwidth (MB/sec): 16.4023
Min bandwidth (MB/sec): 0
Average IOPS:           1046
Stddev IOPS:            1239
Max IOPS:               4199
Min IOPS:               0
Average Latency(s):     0.0152345
Stddev Latency(s):      0.426569
Max latency(s):         27.3506
Min latency(s):         0.0012833
```

Third fireball – CPU time

Have a look on tuned profiles.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

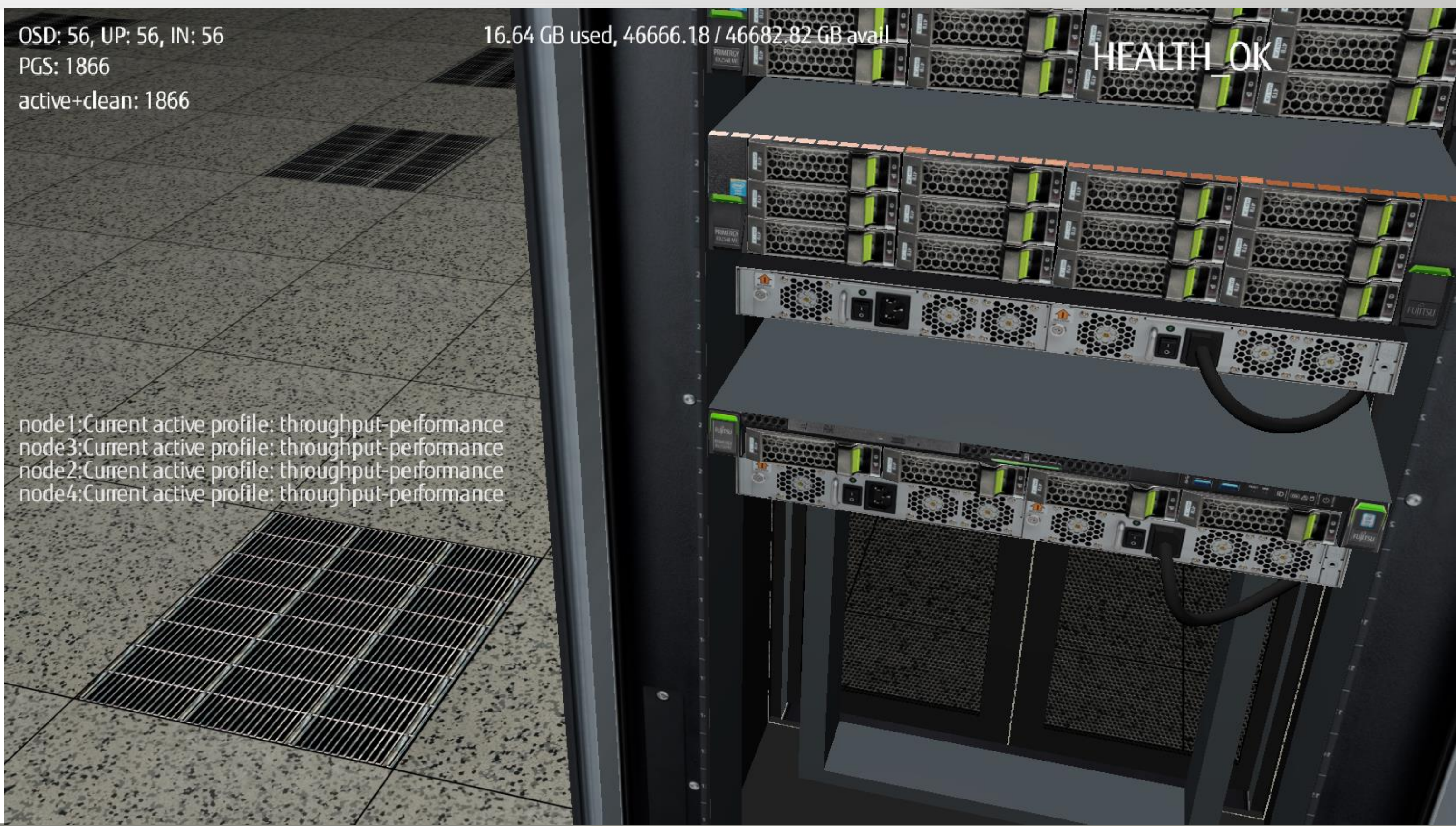
16.64 GB used, 46666.18 / 46682.82 GB avail

HEALTH_OK

```
[ pmgmt ]
tuned: latency-performance
tuned: throughput-performance
tuned: network-latency
tuned: network-throughput
tuned: balanced
tuned: powersave
tuned: show active profiles
cpm: set all osd on one cpu
cpm: set all osd on all cpus
cpm: set all osd sched ilde
cpm: split thread groups
cpm: show thread split
```

Third fireball – CPU time

The settings are telling us that the cluster is profiled to achieve max performance.



```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

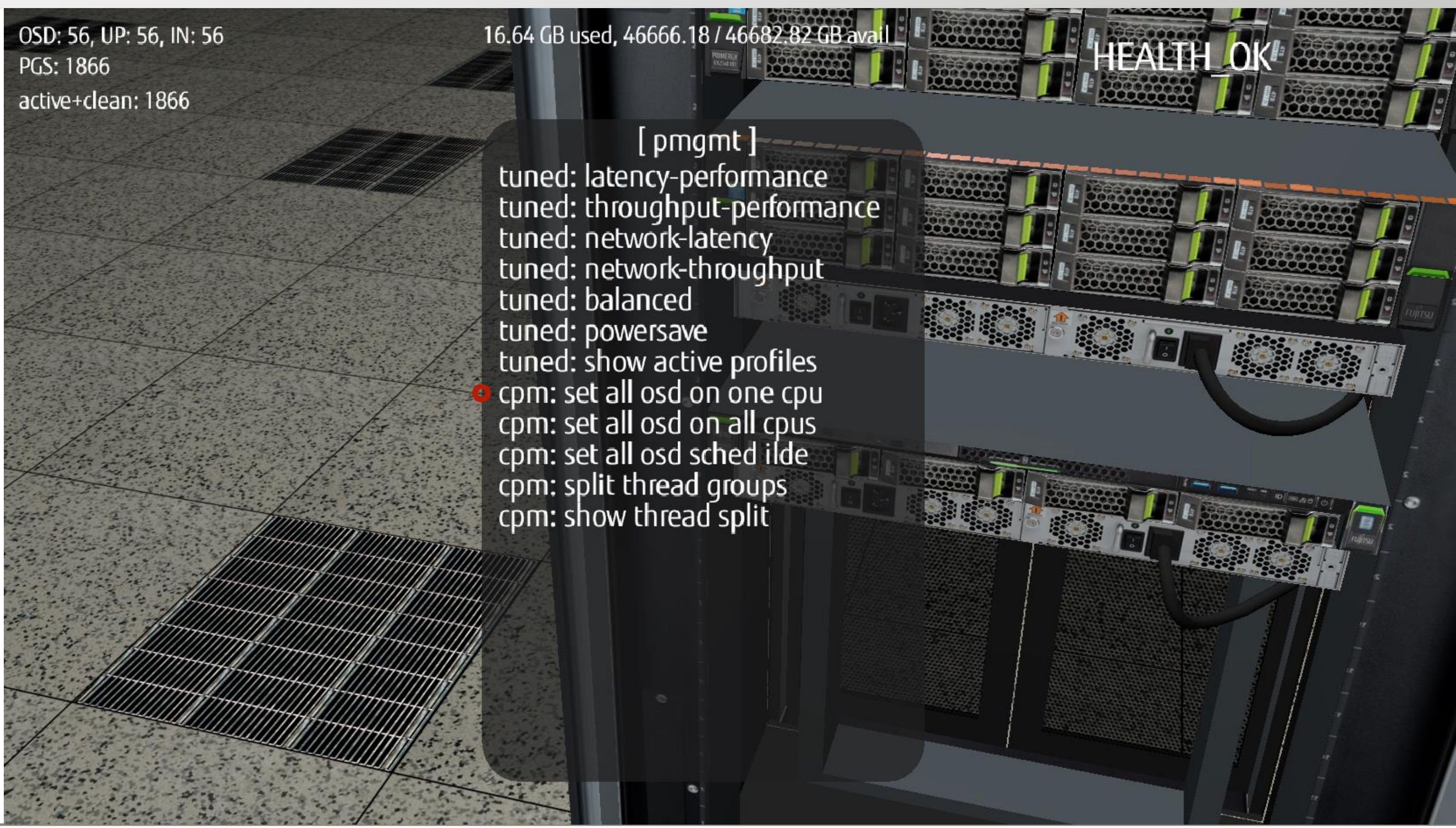
```
16.64 GB used, 46666.18 / 46682.82 GB avail
```

```
HEALTH_OK
```

```
node1:Current active profile: throughput-performance
node3:Current active profile: throughput-performance
node2:Current active profile: throughput-performance
node4:Current active profile: throughput-performance
```

Third fireball – CPU time

Using CPM (our newly developed tool), pin all ceph-osd processes to first logical CPU core on every node in the cluster.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

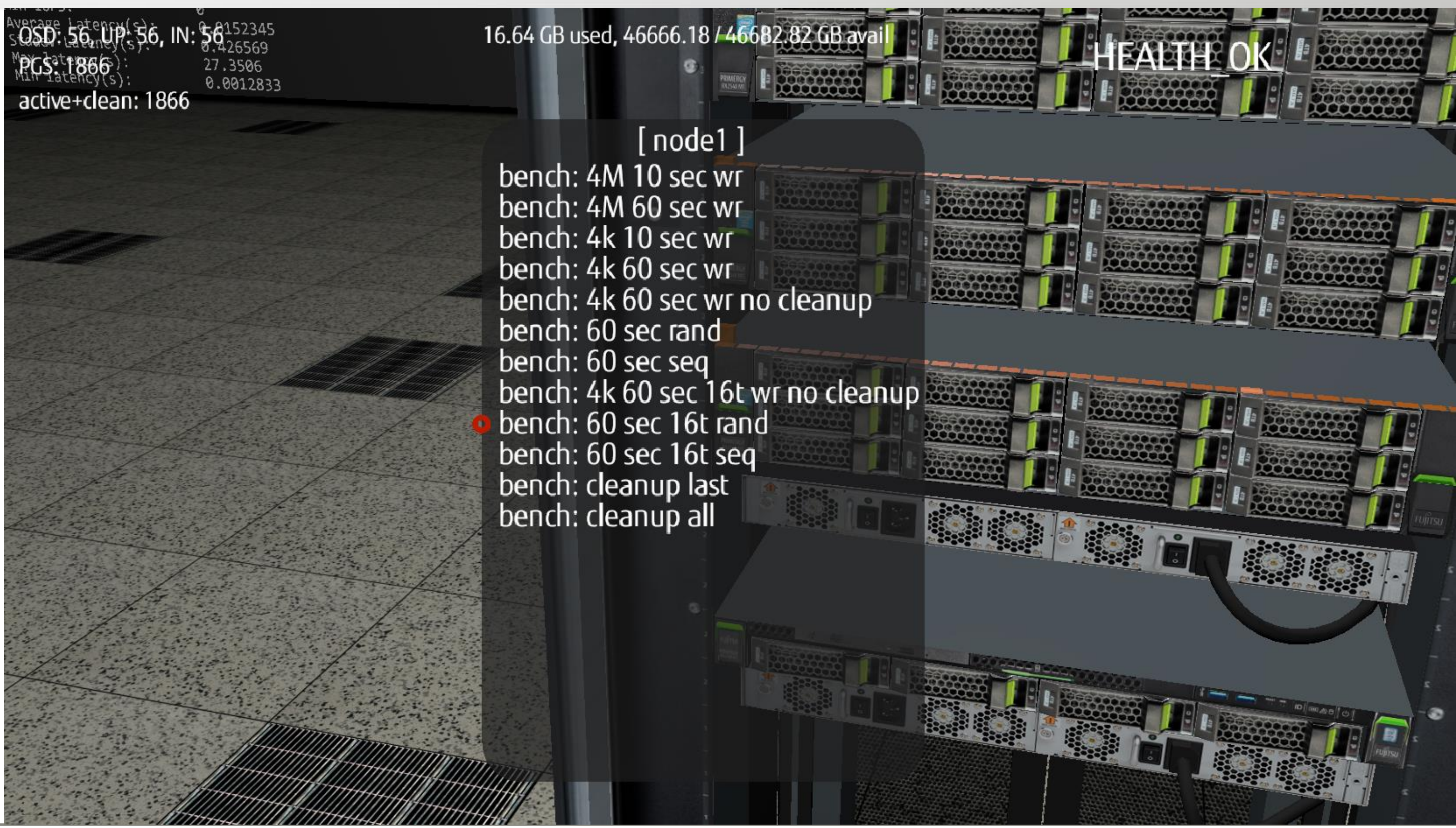
16.64 GB used, 46666.18 / 46682.82 GB avail

HEALTH_OK

[pmgmt]
tuned: latency-performance
tuned: throughput-performance
tuned: network-latency
tuned: network-throughput
tuned: balanced
tuned: powersave
tuned: show active profiles
cpm: set all osd on one cpu
cpm: set all osd on all cpus
cpm: set all osd sched ilde
cpm: split thread groups
cpm: show thread split

Third fireball – CPU time

Start rados bench test with16 threads to see cpu usage.



```
Min IOPS:                 0
Average Latency(s):       0.0152345
Stddev Latency(s):        0.426569
Max latency(s):           27.3506
Min latency(s):           0.0012833
```

OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

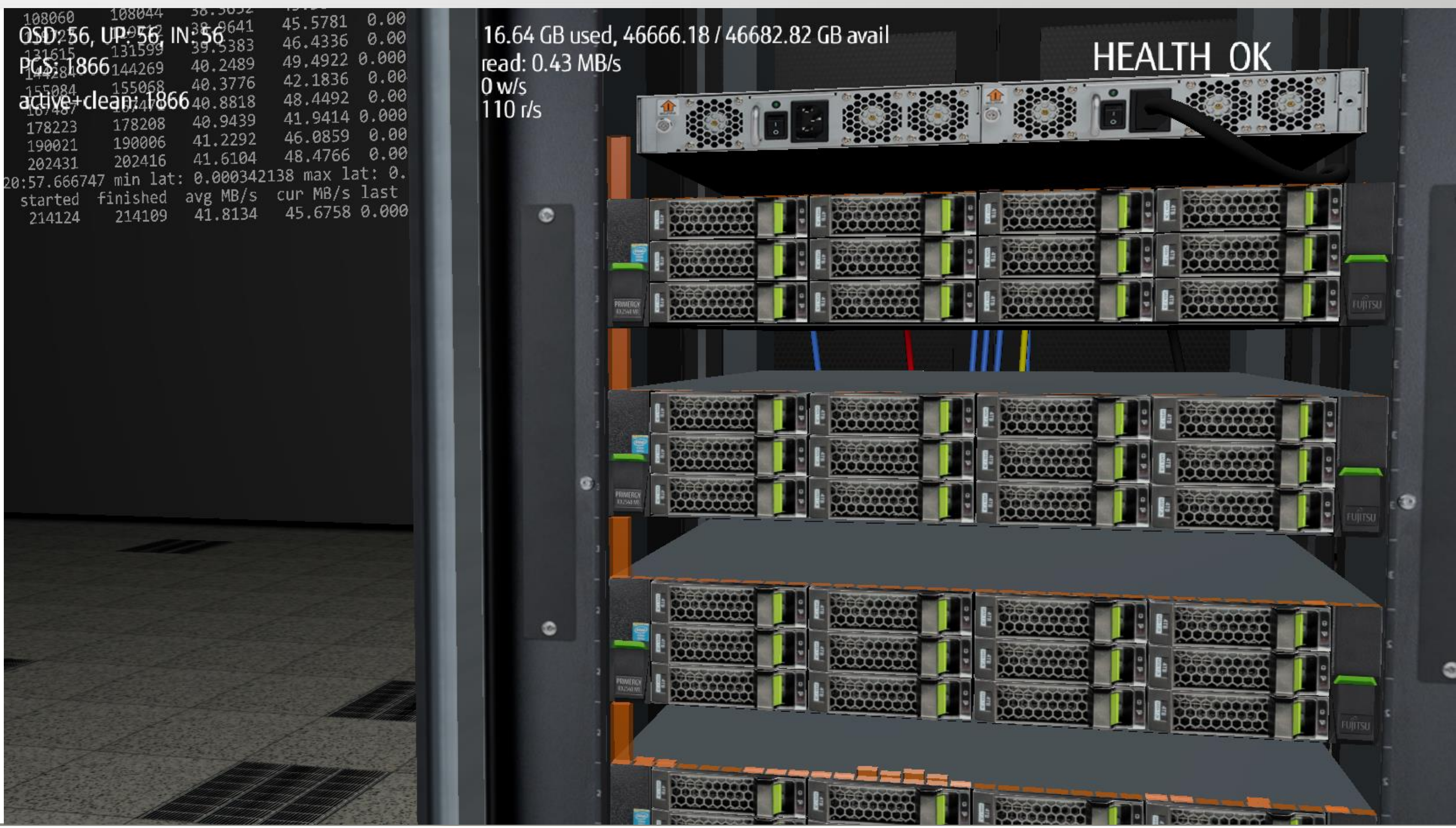
16.64 GB used, 46666.18 / 46682.82 GB avail

HEALTH_OK

[node1]
bench: 4M 10 sec wr
bench: 4M 60 sec wr
bench: 4k 10 sec wr
bench: 4k 60 sec wr
bench: 4k 60 sec wr no cleanup
bench: 60 sec rand
bench: 60 sec seq
bench: 4k 60 sec 16t wr no cleanup
● bench: 60 sec 16t rand
bench: 60 sec 16t seq
bench: cleanup last
bench: cleanup all

Third fireball – CPU time

OSDs are allowed only to use first logical CPU core, bandwith dropped twice.



```
 108060     108044    38.5652    
 [illegible] [illegible] 38.9641  45.5781  0.00
 131615     131599    39.5383    46.4336  0.00
 [illegible] 144269   40.2489    49.4922 0.000
 155084     155068    40.3776    42.1836  0.00
 [illegible] [illegible] 40.8818  48.4492  0.00
 178223     178208    40.9439    41.9414 0.000
 190021     190006    41.2292    46.0859  0.00
 202431     202416    41.6104    48.4766  0.00
20:57.666747 min lat: 0.000342138 max lat: 0.
 started  finished  avg MB/s  cur MB/s last
 214124     214109    41.8134    45.6758 0.000
```

OSD: 56, UP: 56, IN: 56

PGS: 1866

active+clean: 1866

16.64 GB used, 46666.18 / 46682.82 GB avail

read: 0.43 MB/s

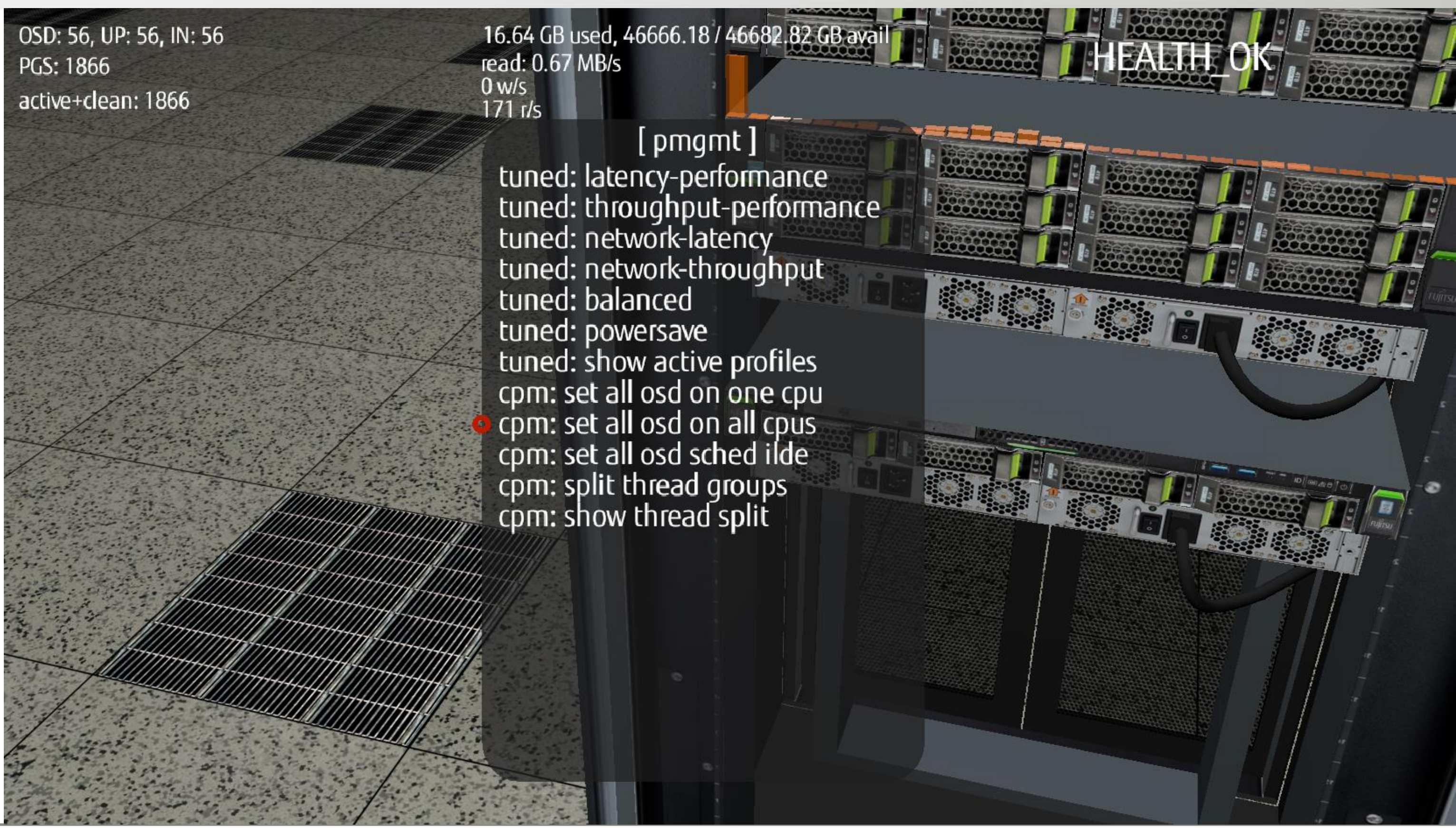
0 w/s

110 r/s

HEALTH_OK

Third fireball – CPU time

Allow each OSD to use every logical core.



```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
16.64 GB used, 46666.18 / 46682.82 GB avail
read: 0.67 MB/s
0 w/s
171 r/s
```

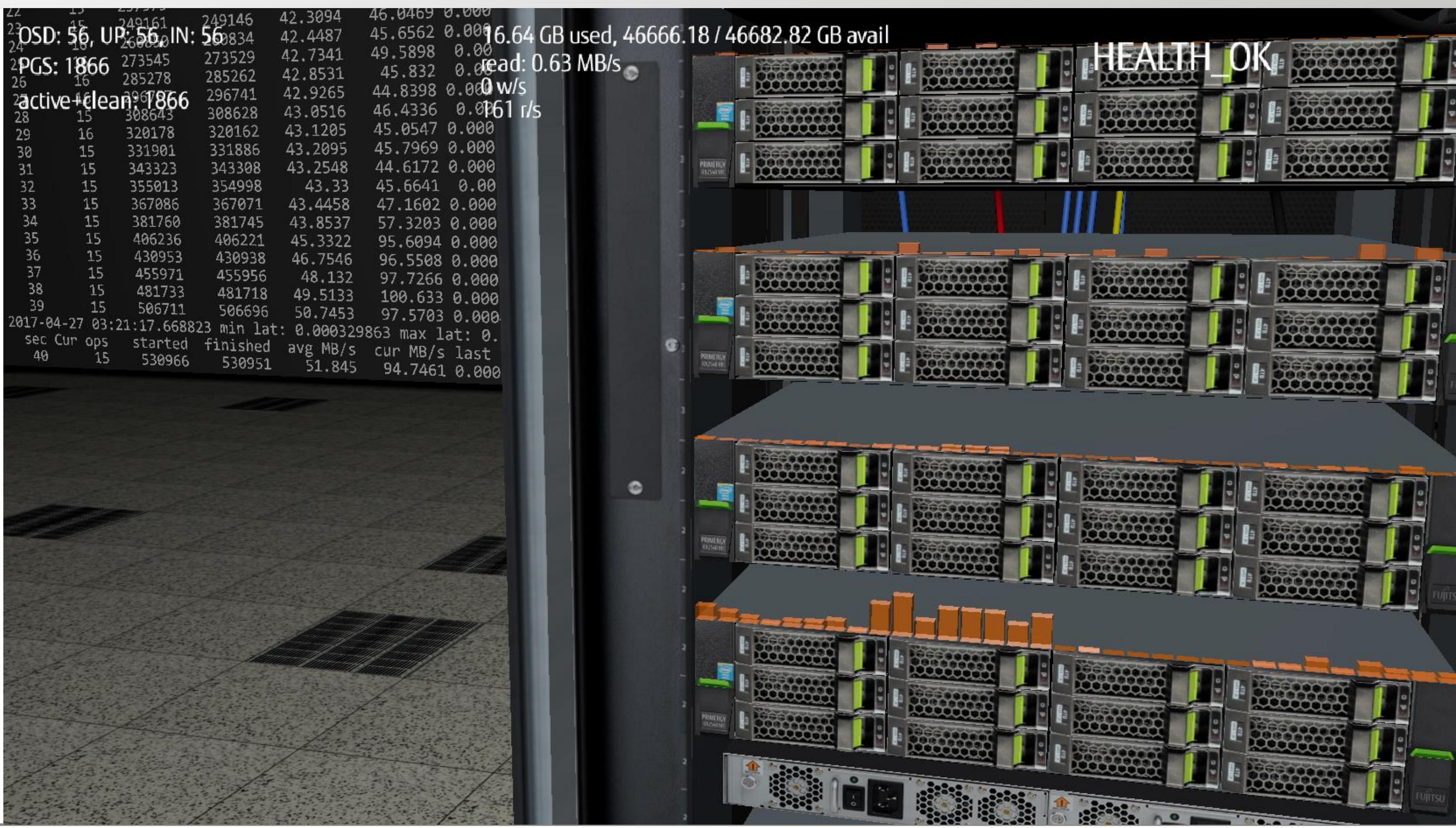
```
HEALTH_OK
```

[pmgmt]
tuned: latency-performance
tuned: throughput-performance
tuned: network-latency
tuned: network-throughput
tuned: balanced
tuned: powersave
tuned: show active profiles
cpm: set all osd on one cpu
cpm: set all osd on all cpus
cpm: set all osd sched ilde
cpm: split thread groups
cpm: show thread split

Third fireball – CPU time

Check rados bench wall – bandwith jumped back to normal level.

OSDs are now using every core (0-31).



OSD: 56, UP: 56, IN: 56

PGS: 1866

active+clean: 1866

16.64 GB used, 46666.18 / 46682.82 GB avail

read: 0.63 MB/s

0 w/s

161 r/s

HEALTH_OK

```
 22      15
 23      15    249161    249146   42.3094   46.0469 0.000
 24      16    260850    260834   42.4487   45.6562 0.000
 25      16    273545    273529   42.7341   49.5898  0.00
 26      16    285278    285262   42.8531    45.832  0.00
 27      16    296757    296741   42.9265   44.8398 0.000
 28      15    308643    308628   43.0516   46.4336 0.0
 29      16    320178    320162   43.1205   45.0547 0.000
 30      15    331901    331886   43.2095   45.7969 0.000
 31      15    343323    343308   43.2548   44.6172 0.000
 32      15    355013    354998     43.33   45.6641  0.00
 33      15    367086    367071   43.4458   47.1602 0.000
 34      15    381760    381745   43.8537   57.3203 0.000
 35      15    406236    406221   45.3322   95.6094 0.000
 36      15    430953    430938   46.7546   96.5508 0.000
 37      15    455971    455956    48.132   97.7266 0.000
 38      15    481733    481718   49.5133   100.633 0.000
 39      15    506711    506696   50.7453   97.5703 0.000
2017-04-27 03:21:17.668823 min lat: 0.000329863 max lat: 0.
  sec Cur ops   started  finished  avg MB/s  cur MB/s last
   40      15    530966    530951    51.845   94.7461 0.000
```

Third fireball – CPU time

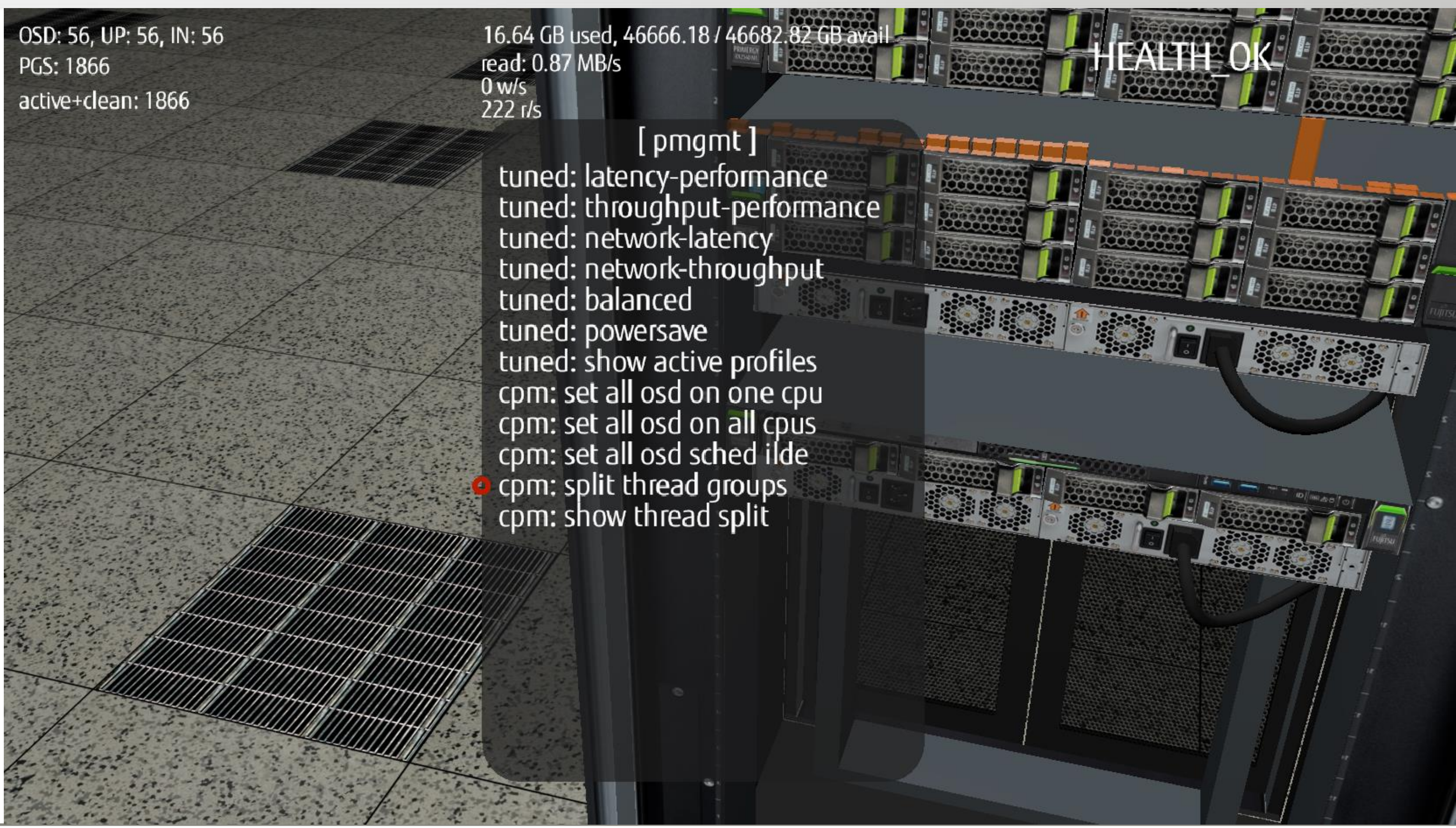
OSD has about 28 uniqe thread names.

Let’s pin each OSD thread name to each of logical cpu cores (0-27).

Thanks to this we could easily check which thread groups need more CPU time than the others.

Unique thread names for OSD:

admin_socket, ceph-osd, filestore_sync, fn_anonymous, fn_appl_fstore, fn_jrn_objstore, fn_odsk_fstore, journal_write, journal_wrt_fin, log, ms_accepter, ms_dispatch, ms_local, ms_pipe_read, ms_pipe_write, ms_reaper, osd_srv_agent, osd_srv_heartbt, safe_timer, service, sginal_handler, tp_fstore_op, tp_osd, tp_osd_cmd, tp_osd_disk, tp_osd_recov, tp_osd_tp, wb_throttle



```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
16.64 GB used, 46666.18 / 46682.82 GB avail
read: 0.87 MB/s
0 w/s
222 r/s
```

HEALTH_OK

[pmgmt]
tuned: latency-performance
tuned: throughput-performance
tuned: network-latency
tuned: network-throughput
tuned: balanced
tuned: powersave
tuned: show active profiles
cpm: set all osd on one cpu
cpm: set all osd on all cpus
cpm: set all osd sched ilde
● cpm: split thread groups
cpm: show thread split

Third fireball – CPU time

The ones with 100% usage are:
- ms_accepter
- tp_osd_tp

OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

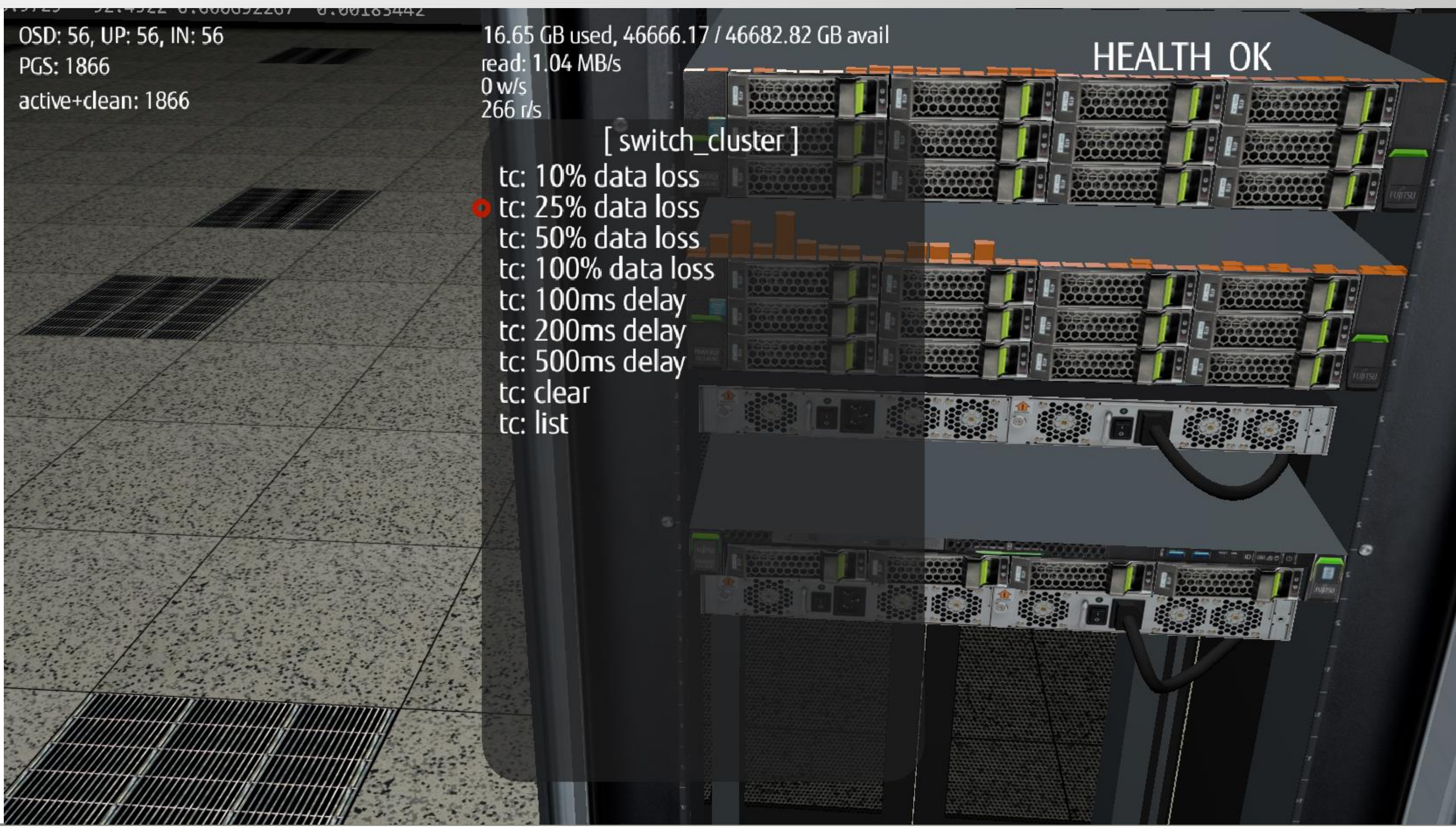
16.64 GB used, 46666.18 / 46682.82 GB avail
read: 0.28 MB/s
0 w/s
72 r/s

HEALTH_OK

```
  210643    210628    54.8407
                      54.7778     53.8398 0.000
  239328    239313    54.9795     58.2109  0.00
            253587    55.0224     55.7578 0.000
  267977    267962    55.0816     56.1523 0.000
                      0.000334096 max lat: 0.
 started  finished   avg MB/s   cur MB/s last
  283061    283046    55.2732     58.9219  0.00
  298735    298719     55.556     61.2227 0.000
  312575    312560    55.4881     54.0664 0.000
  327806    327790    55.6619     59.4922 0.000
  343000    342985    55.8155     59.3555 0.000
  358578    358563    56.0167     60.8516 0.000
```

Multiple fireballs at once

Add 25% data loss to every NIC connected to cluster switch.



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

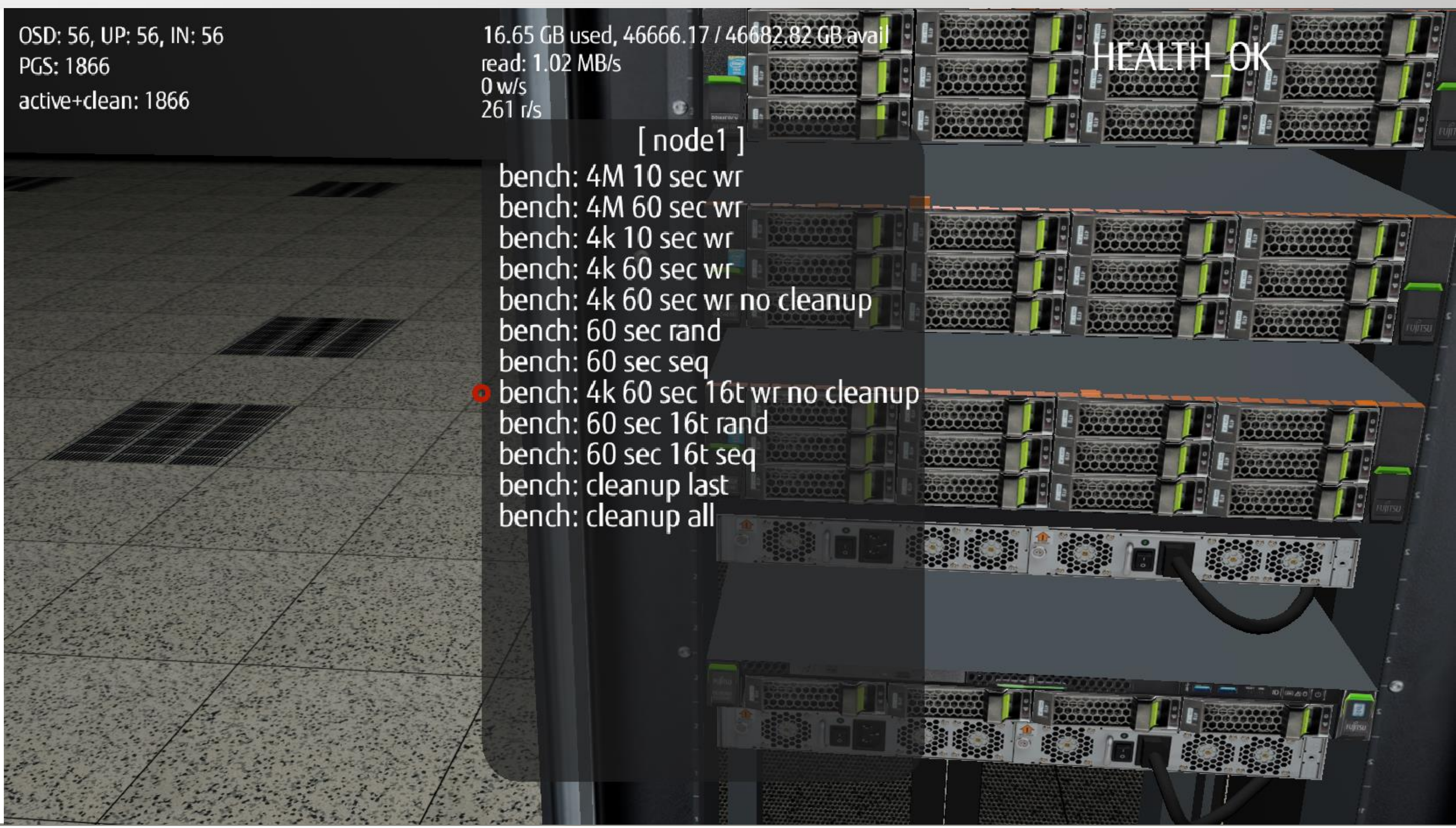
16.65 GB used, 46666.17 / 46682.82 GB avail
read: 1.04 MB/s
0 w/s
266 r/s

HEALTH_OK

[switch_cluster]
tc: 10% data loss
tc: 25% data loss
tc: 50% data loss
tc: 100% data loss
tc: 100ms delay
tc: 200ms delay
tc: 500ms delay
tc: clear
tc: list

Multiple fireballs at once

Start rados bench write to see cluster reaction.



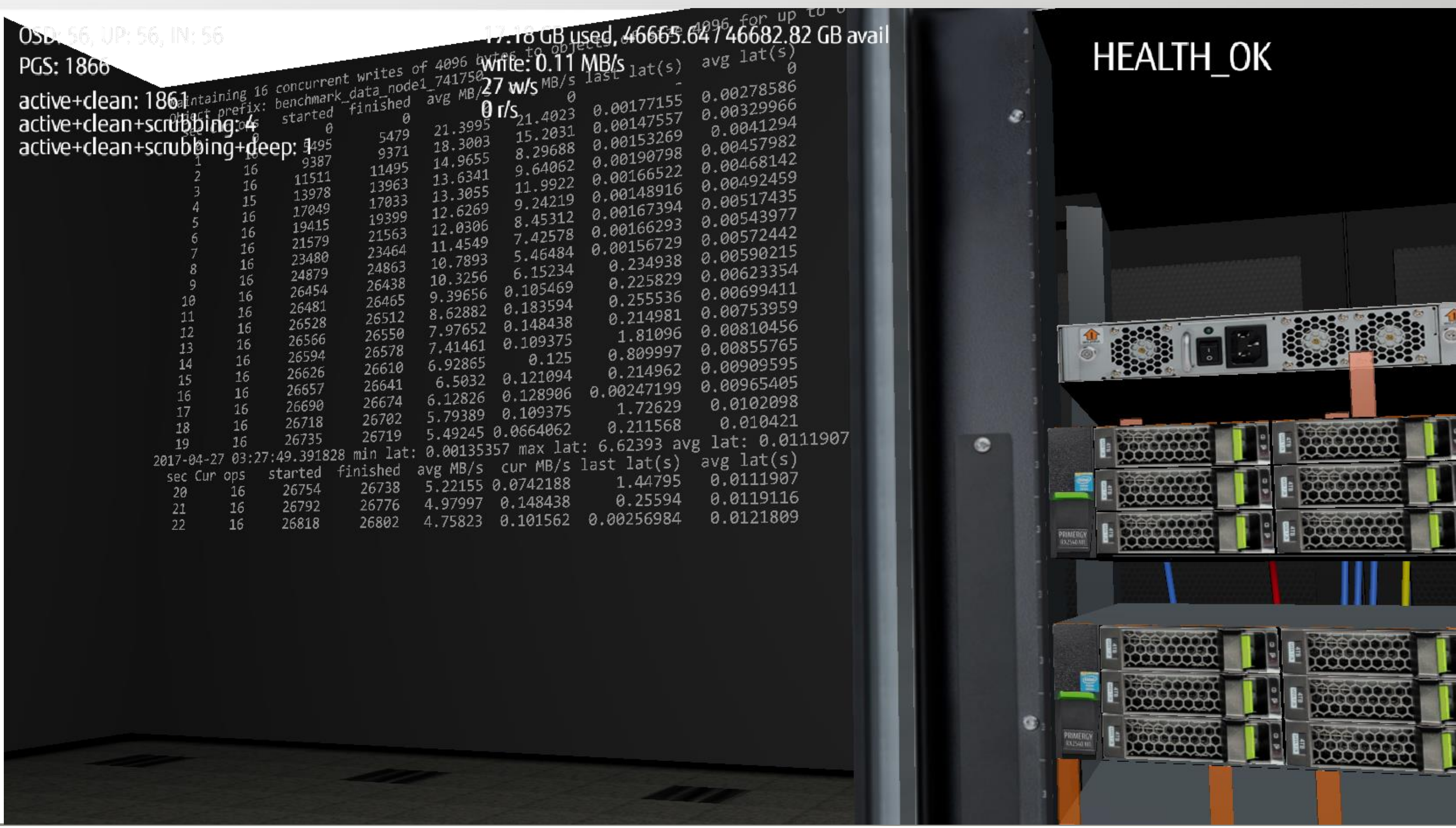
```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866
```

```
16.65 GB used, 46666.17 / 46682.82 GB avail
read: 1.02 MB/s
0 w/s
261 r/s
```

HEALTH_OK

Multiple fireballs at once

As we suspected, writes are starting to be unstable.



```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1861
active+clean+scrubbing: 4
active+clean+scrubbing+deep: 1
```

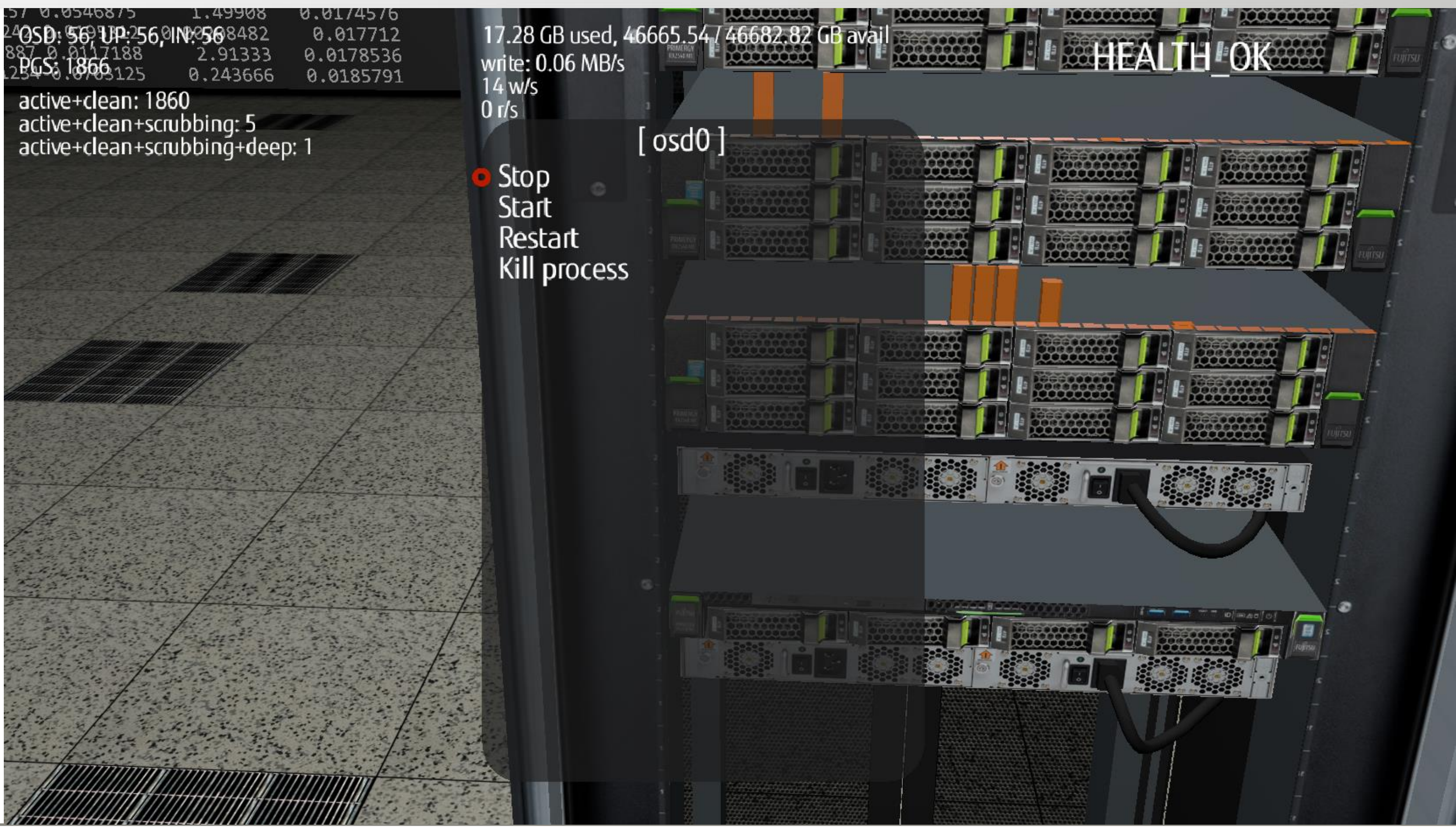
```
17.18 GB used, 46665.64 / 46682.82 GB avail
write: 0.11 MB/s
27 w/s
0 r/s
```

```
HEALTH_OK
```

```
Maintaining 16 concurrent writes of 4096 bytes to objects of size 4096 for up to 6
Object prefix: benchmark_data_node1_741750
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
    0       0         0         0         0         0           -           0
    0      16      5495      5479   21.3995   21.4023  0.00177155  0.00278586
    1      16      9387      9371   18.3003   15.2031  0.00147557  0.00329966
    2      16     11511     11495   14.9655   8.29688  0.00153269   0.0041294
    3      16     13978     13963   13.6341   9.64062  0.00190798  0.00457982
    4      15     17049     17033   13.3055   11.9922  0.00166522  0.00468142
    5      16     19415     19399   12.6269   9.24219  0.00148916  0.00492459
    6      16     21579     21563   12.0306   8.45312  0.00167394  0.00517435
    7      16     23480     23464   11.4549   7.42578  0.00166293  0.00543977
    8      16     24879     24863   10.7893   5.46484  0.00156729  0.00572442
    9      16     26454     26438   10.3256   6.15234    0.234938  0.00590215
   10      16     26481     26465   9.39656  0.105469    0.225829  0.00623354
   11      16     26528     26512   8.62882  0.183594    0.255536  0.00699411
   12      16     26566     26550   7.97652  0.148438    0.214981  0.00753959
   13      16     26594     26578   7.41461  0.109375     1.81096  0.00810456
   14      16     26626     26610   6.92865     0.125    0.809997  0.00855765
   15      16     26657     26641    6.5032  0.121094    0.214962  0.00909595
   16      16     26690     26674   6.12826  0.128906  0.00247199  0.00965405
   17      16     26718     26702   5.79389  0.109375     1.72629   0.0102098
   18      16     26735     26719   5.49245 0.0664062    0.211568    0.010421
   19      16     26735     26719   5.49245 0.0664062    0.211568    0.010421
2017-04-27 03:27:49.391828 min lat: 0.00135357 max lat: 6.62393 avg lat: 0.0111907
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
   20      16     26754     26738   5.22155 0.0742188     1.44795   0.0111907
   21      16     26792     26776   4.97997  0.148438     0.25594   0.0119116
   22      16     26818     26802   4.75823  0.101562  0.00256984   0.0121809
```

Multiple fireballs at once

Stop OSD.0 gracefully, this should create small rebalance.



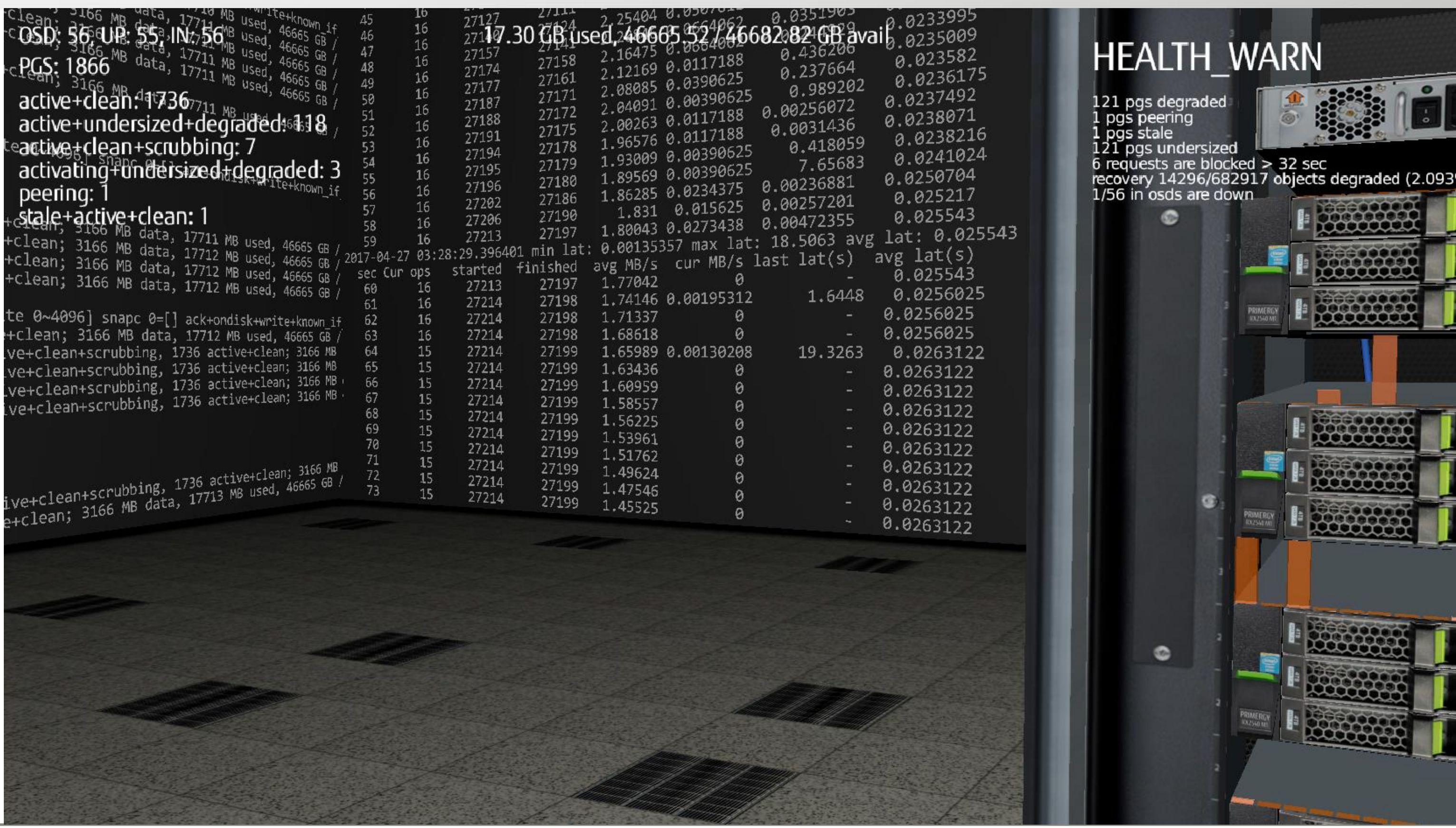
```
OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1860
active+clean+scrubbing: 5
active+clean+scrubbing+deep: 1
```

```
17.28 GB used, 46665.54 / 46682.82 GB avail
write: 0.06 MB/s
14 w/s
0 r/s
```

HEALTH_OK

Multiple fireballs at once

Writes were blocked again.



```
OSD: 56, UP: 55, IN: 56
PGS: 1866
active+clean: 1736
active+undersized+degraded: 118
active+clean+scrubbing: 7
activating+undersized+degraded: 3
peering: 1
stale+active+clean: 1
```

```
17.30 GB used, 46665.52 / 46682.82 GB avail
```

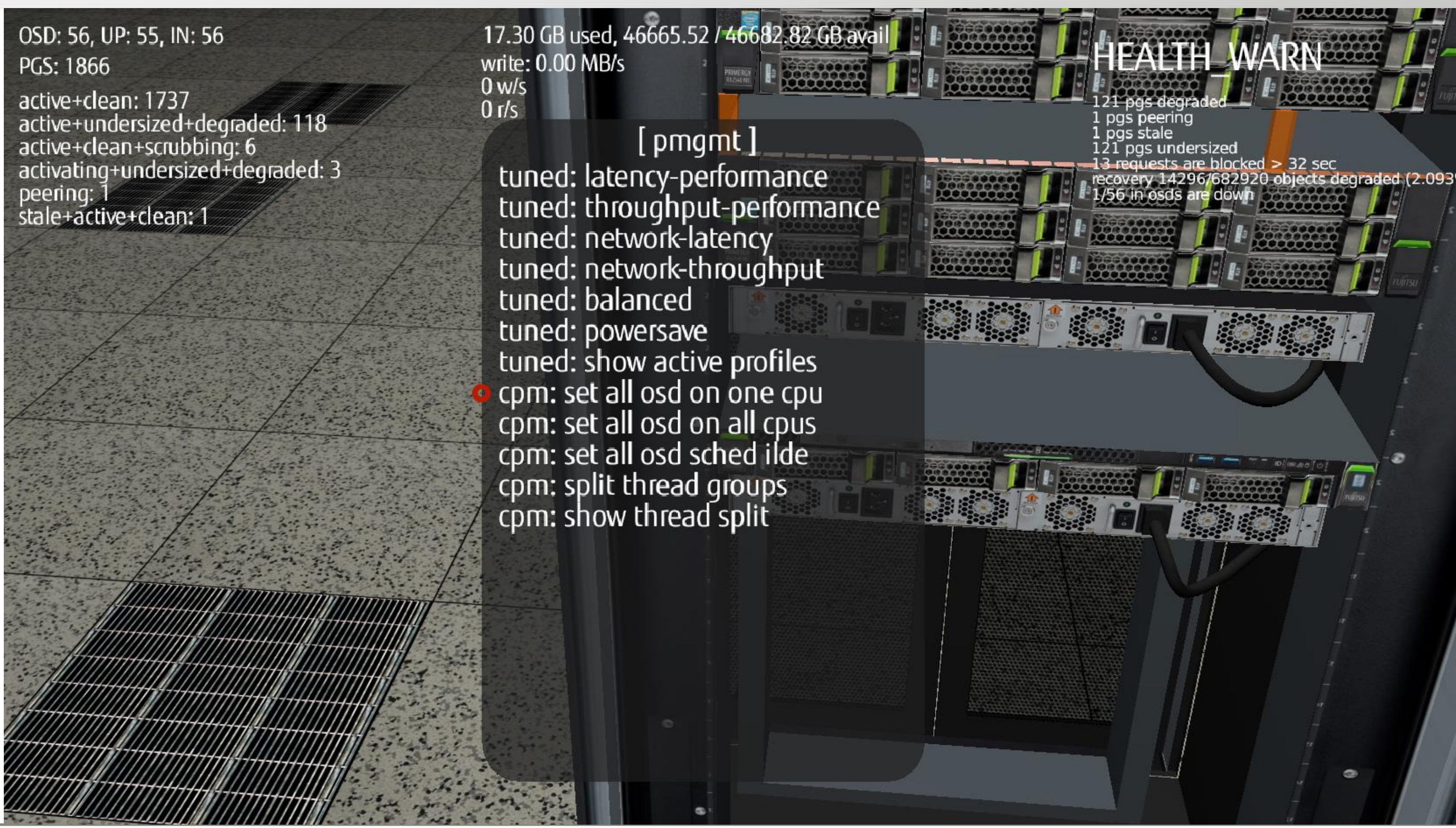
```
+clean; 3166 MB data, 17711 MB used, 46665 GB /
+clean; 3166 MB data, 17712 MB used, 46665 GB /
+clean; 3166 MB data, 17712 MB used, 46665 GB /
+clean; 3166 MB data, 17712 MB used, 46665 GB /
te 0~4096] snapc 0=[] ack+ondisk+write+known_if
+clean; 3166 MB data, 17712 MB used, 46665 GB /
ve+clean+scrubbing, 1736 active+clean; 3166 MB
ve+clean+scrubbing, 1736 active+clean; 3166 MB
ve+clean+scrubbing, 1736 active+clean; 3166 MB
ive+clean+scrubbing, 1736 active+clean; 3166 MB
ive+clean+scrubbing, 1736 active+clean; 3166 MB
e+clean; 3166 MB data, 17713 MB used, 46665 GB /
```

```
 45     16   [illegible]
 46     16   27127    27111   2.25404  [illegible]  0.0351905  [illegible]
 47     16   [illegible] 27124 [illegible] 0.0664062 [illegible] 0.0233995
 48     16   27157    27141   [illegible] 0.0664062 0.436206  0.0235009
 49     16   27174    27158   2.16475  0.0117188   0.237664   0.023582
 50     16   27177    27161   2.12169  0.0390625   0.989202   0.0236175
 51     16   27187    27171   2.08085  0.00390625  0.00256072 0.0237492
 52     16   27188    27172   2.04091  0.0117188   0.0031436  0.0238071
 53     16   27191    27175   2.00263  0.0117188   0.418059   0.0238216
 54     16   27194    27178   1.96576  0.00390625  7.65683    0.0241024
 55     16   27195    27179   1.93009  0.00390625  0.00236881 0.0250704
 56     16   27196    27180   1.89569  0.0234375   0.00257201 0.025217
 57     16   27202    27186   1.86285  0.015625    0.00472355 0.025543
 58     16   27206    27190   1.831    0.0273438
 59     16   27213    27197   1.80043
2017-04-27 03:28:29.396401 min lat: 0.00135357 max lat: 18.5063 avg lat: 0.025543
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
                                                             -    0.025543
   60      16     27213     27197   1.77042         0           -    0.025543
   61      16     27214     27198   1.74146 0.00195312     1.6448   0.0256025
   62      16     27214     27198   1.71337         0           -   0.0256025
   63      16     27214     27198   1.68618         0           -   0.0256025
   64      15     27214     27199   1.65989 0.00130208    19.3263   0.0263122
   65      15     27214     27199   1.63436         0           -   0.0263122
   66      15     27214     27199   1.60959         0           -   0.0263122
   67      15     27214     27199   1.58557         0           -   0.0263122
   68      15     27214     27199   1.56225         0           -   0.0263122
   69      15     27214     27199   1.53961         0           -   0.0263122
   70      15     27214     27199   1.51762         0           -   0.0263122
   71      15     27214     27199   1.49624         0           -   0.0263122
   72      15     27214     27199   1.47546         0           -   0.0263122
   73      15     27214     27199   1.45525         0           -   0.0263122
```

```
HEALTH_WARN
121 pgs degraded
1 pgs peering
1 pgs stale
121 pgs undersized
6 requests are blocked > 32 sec
recovery 14296/682917 objects degraded (2.093
1/56 in osds are down
```

Multiple fireballs at once

Allow OSDs only to use single logical CPU core.



```
OSD: 56, UP: 55, IN: 56
PGS: 1866
active+clean: 1737
active+undersized+degraded: 118
active+clean+scrubbing: 6
activating+undersized+degraded: 3
peering: 1
stale+active+clean: 1
```

```
17.30 GB used, 46665.52 / 46682.82 GB avail
write: 0.00 MB/s
0 w/s
0 r/s
```

```
HEALTH_WARN
121 pgs degraded
1 pgs peering
1 pgs stale
121 pgs undersized
13 requests are blocked > 32 sec
recovery 14296/682920 objects degraded (2.093
1/56 in osds are down
```

[pmgmt]
tuned: latency-performance
tuned: throughput-performance
tuned: network-latency
tuned: network-throughput
tuned: balanced
tuned: powersave
tuned: show active profiles
cpm: set all osd on one cpu
cpm: set all osd on all cpus
cpm: set all osd sched ilde
cpm: split thread groups
cpm: show thread split

Multiple fireballs at once

Writes are still blocked, and since OSDs are working only on one cpu, recovery process now runs slower.

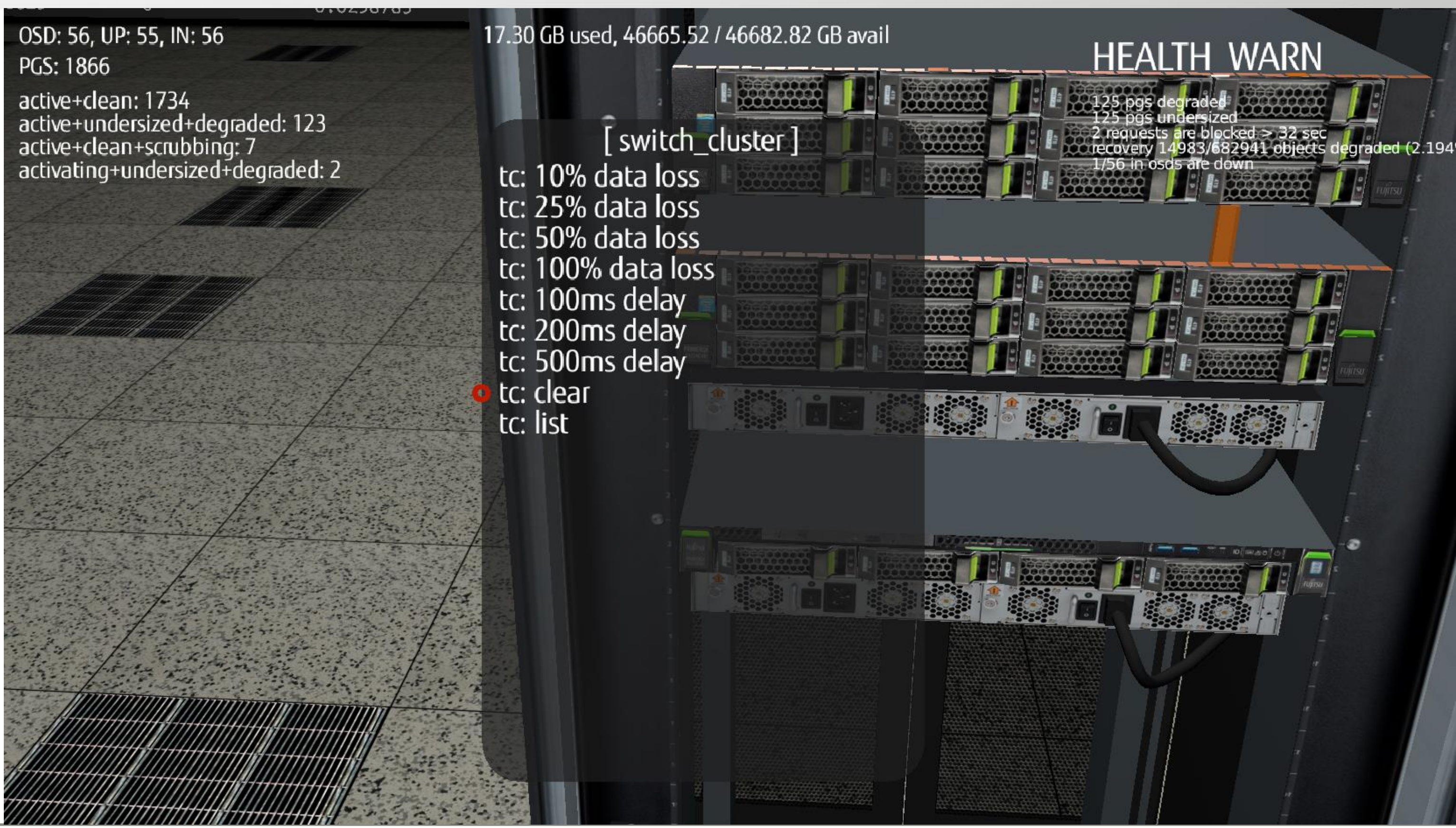
```
OSD: 56, UP: 55, IN: 56
PGS: 1866
active+clean: 1735
active+undersized+degraded: 123
active+clean+scrubbing: 6
activating+undersized+degraded: 2
```

```
17.30 GB used, 46665.52 / 46682.82 GB avail
```

```
HEALTH_WARN
125 pgs degraded
125 pgs undersized
11 requests are blocked > 32 sec
recovery 14983/682941 objects degraded (2.194
1/56 in osds are down
```

Multiple fireballs at once

Let’s check if cluster will recover after cluster network switch will be healed.



OSD: 56, UP: 55, IN: 56
PGS: 1866

active+clean: 1734
active+undersized+degraded: 123
active+clean+scrubbing: 7
activating+undersized+degraded: 2

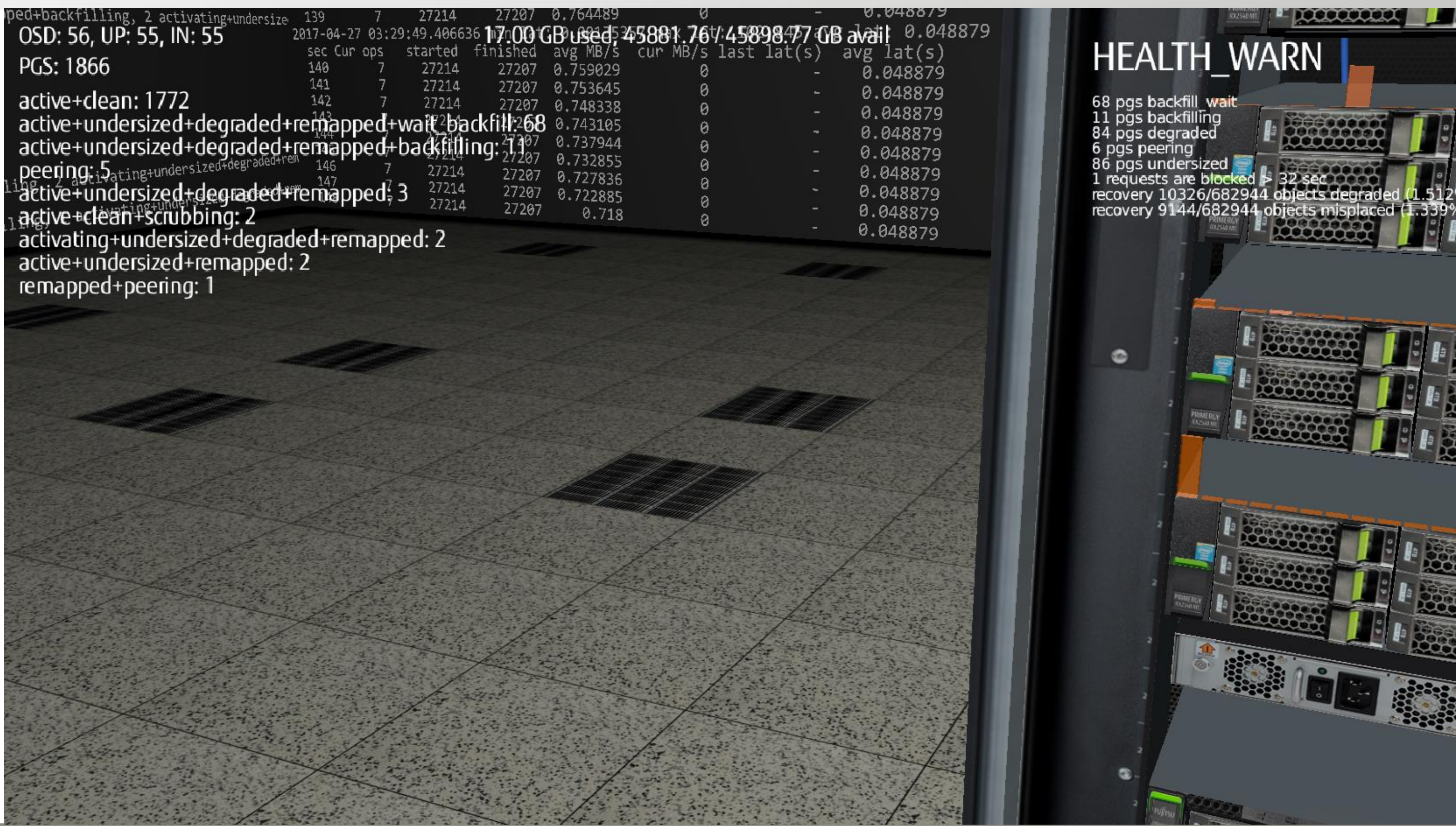
17.30 GB used, 46665.52 / 46682.82 GB avail

HEALTH WARN

125 pgs degraded
125 pgs undersized
2 requests are blocked > 32 sec
recovery 14983/682941 objects degraded (2.194
1/56 in osds are down

Multiple fireballs at once

Writes are still blocked, but there is some bigger movement on cluster.



```
OSD: 56, UP: 55, IN: 55
PGS: 1866

active+clean: 1772
active+undersized+degraded+remapped+wait_backfill: 68
active+undersized+degraded+remapped+backfilling: 11
peering: 5
active+undersized+degraded+remapped: 3
active+clean+scrubbing: 2
activating+undersized+degraded+remapped: 2
active+undersized+remapped: 2
remapped+peering: 1
```

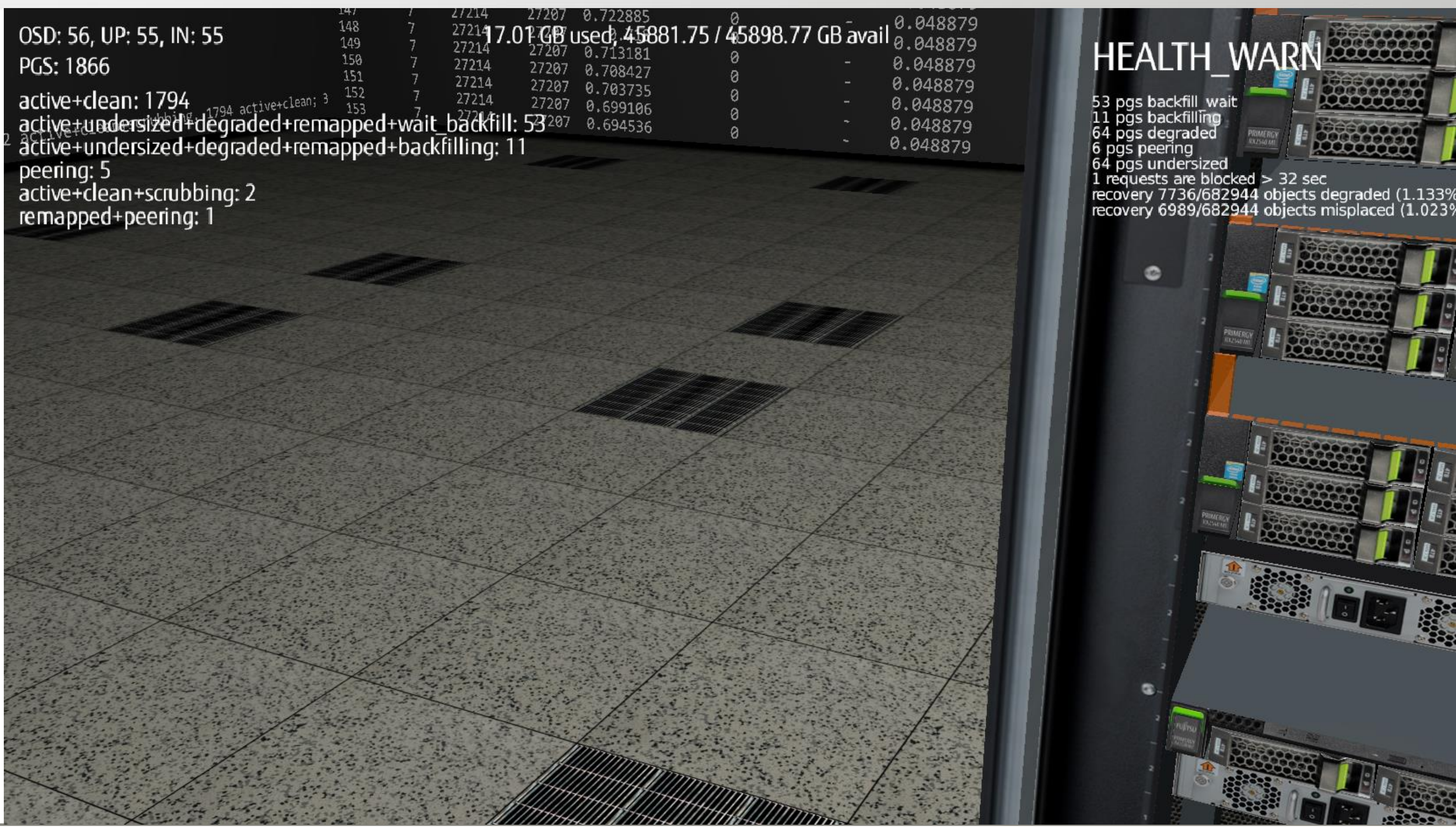
```
  139       7     27214     27207  0.764489         0          -    0.048879
2017-04-27 03:29:49.406636
  sec Cur ops   started  finished  avg MB/s  cur MB/s last lat(s)  avg lat(s)
  140       7     27214     27207  0.759029         0          -    0.048879
  141       7     27214     27207  0.753645         0          -    0.048879
  142       7     27214     27207  0.748338         0          -    0.048879
  143       7     27214     27207  0.743105         0          -    0.048879
  144       7     27214     27207  0.737944         0          -    0.048879
  145       7     27214     27207  0.732855         0          -    0.048879
  146       7     27214     27207  0.727836         0          -    0.048879
  147       7     27214     27207  0.722885         0          -    0.048879
  148       7     27214     27207     0.718         0          -    0.048879
```

```
17.00 GB used, 45881.76 / 45898.77 GB avail
```

```
HEALTH_WARN

68 pgs backfill_wait
11 pgs backfilling
84 pgs degraded
6 pgs peering
86 pgs undersized
1 requests are blocked > 32 sec
recovery 10326/682944 objects degraded (1.512
recovery 9144/682944 objects misplaced (1.339%
```

Multiple fireballs at once

Cluster starts healing.



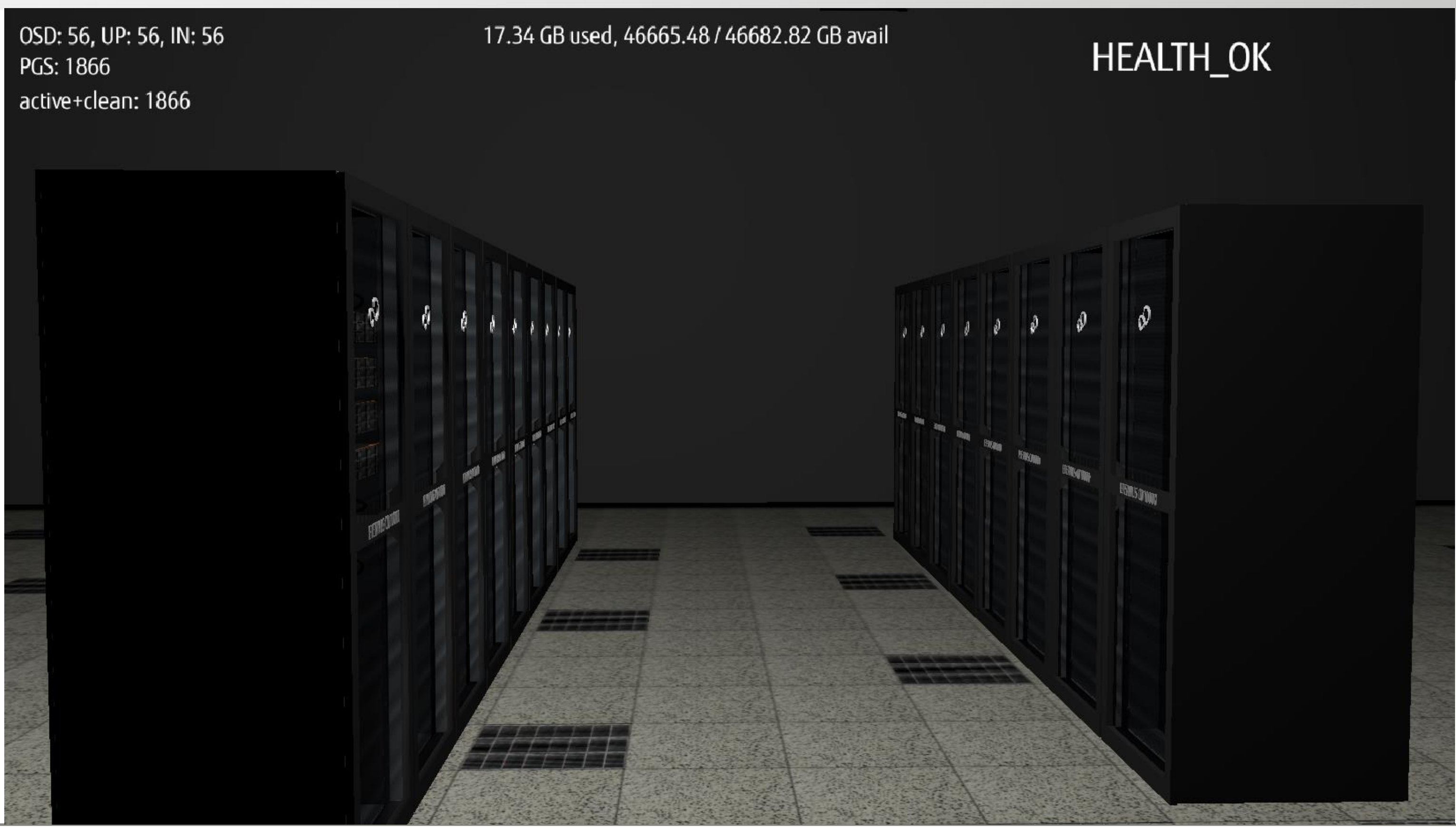
```
OSD: 56, UP: 55, IN: 55
PGS: 1866
active+clean: 1794
active+undersized+degraded+remapped+wait_backfill: 53
active+undersized+degraded+remapped+backfilling: 11
peering: 5
active+clean+scrubbing: 2
remapped+peering: 1
```

```
17.01 GB used, 45881.75 / 45898.77 GB avail
```

```
HEALTH_WARN
53 pgs backfill_wait
11 pgs backfilling
64 pgs degraded
6 pgs peering
64 pgs undersized
1 requests are blocked > 32 sec
recovery 7736/682944 objects degraded (1.133%
recovery 6989/682944 objects misplaced (1.023%
```

Multiple fireballs at once

This time Ceph won!



OSD: 56, UP: 56, IN: 56
PGS: 1866
active+clean: 1866

17.34 GB used, 46665.48 / 46682.82 GB avail

HEALTH_OK

Thank you for your attention ☺

Igor.Podoski@ts.fujitsu.com

Zofia.Domaradzka@ts.fujitsu.com

FUJITSU

shaping tomorrow with you