Challenges of deploying Wide-Area-Network Distributed Storage System under network and reliability constraints – A case study

Mohd Bazli Ab Karim
Advanced Computing Lab
MIMOS Berhad, Malaysia

email
bazli.abkarim@mimos.my

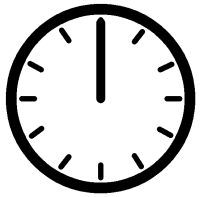
In PRAGMA Student Workshop
PRAGMA26, Tainan, Taiwan
9-11 April 2014

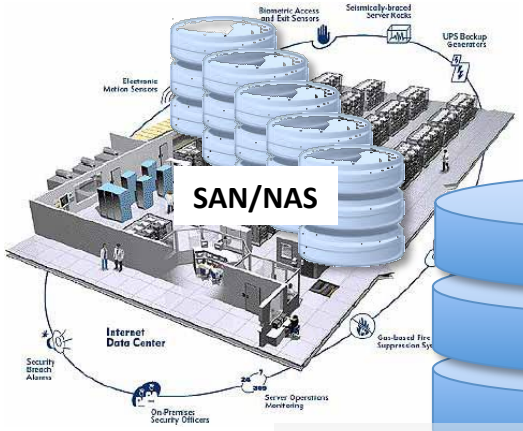
Innovation for Life™

Outline

- Distributed Storage System?
- PRAGMA25
  - DFS over Local Area Network
  - Ceph vs. GlusterFS
- PRAGMA26
  - DFS over Wide Area Network
  - DFS over WAN vs. DFS over LAN

Distributed File System



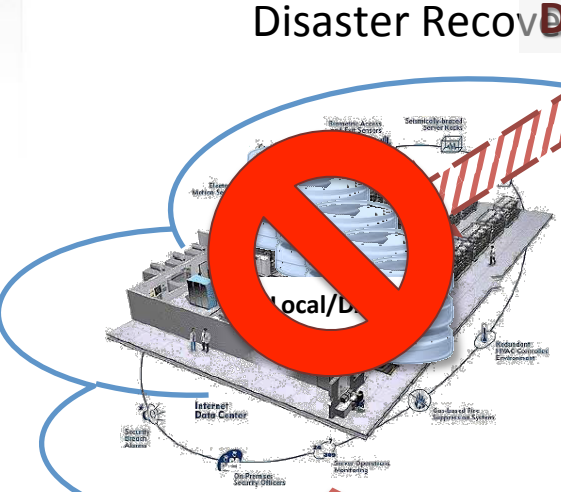


R/W

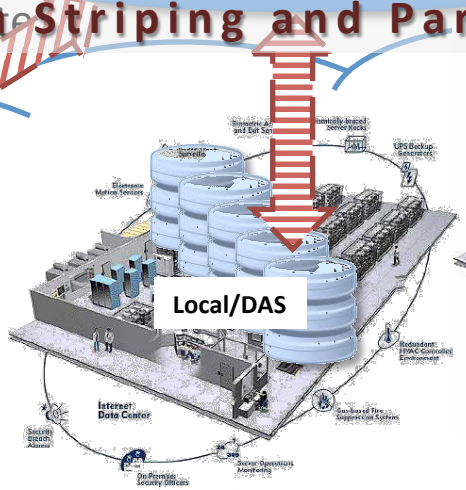
One/Multiple Virtual Volume(s)

Disaster Recovery Site

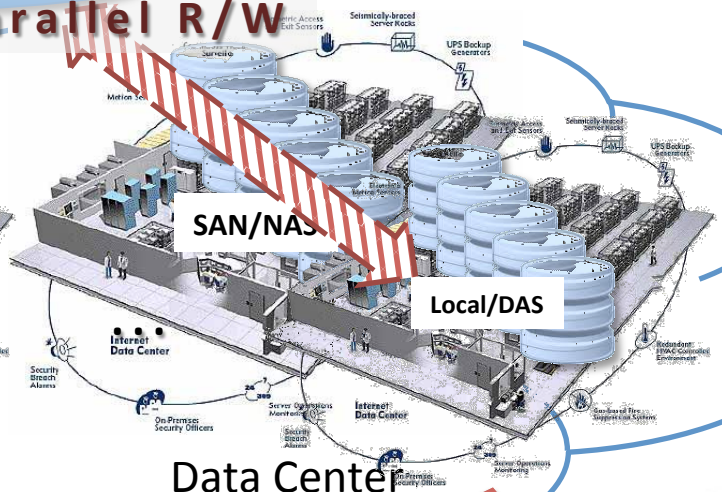
Data Striping and Parallel R/W



Data Center 1



Data Center 2



Data Center

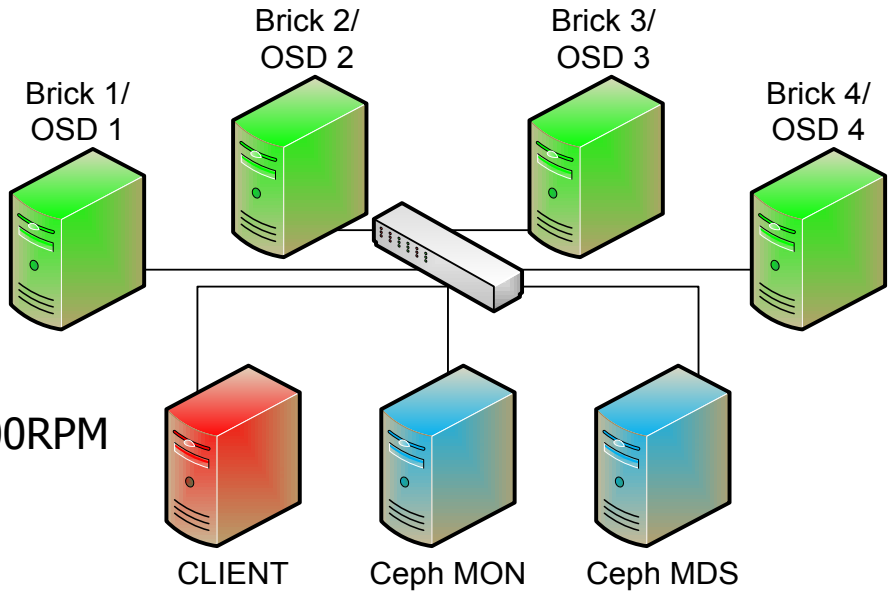
Data Center n

Replications

PRAGMA 25 – DFS on LAN

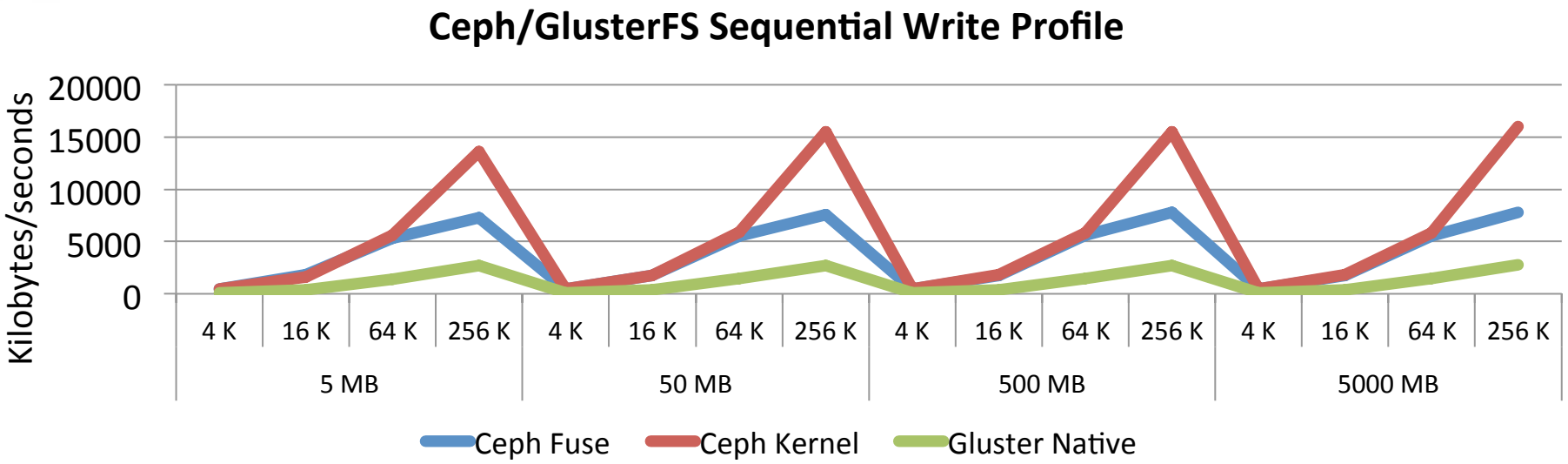
Dell PowerEdge T110 II
Proc: Intel Xeon E3-1220v2 3.10 GHz
Memory: 8 GB
Hard Drives: Seagate Constellation ES 2TB 7200RPM SATA
RAID Controller: LSI Logic SAS2008
Network: 1GbE
Operating System: Ubuntu 12.04
Ceph: 0.61.7 (Cuttlefish)
GlusterFS: 3.4.0

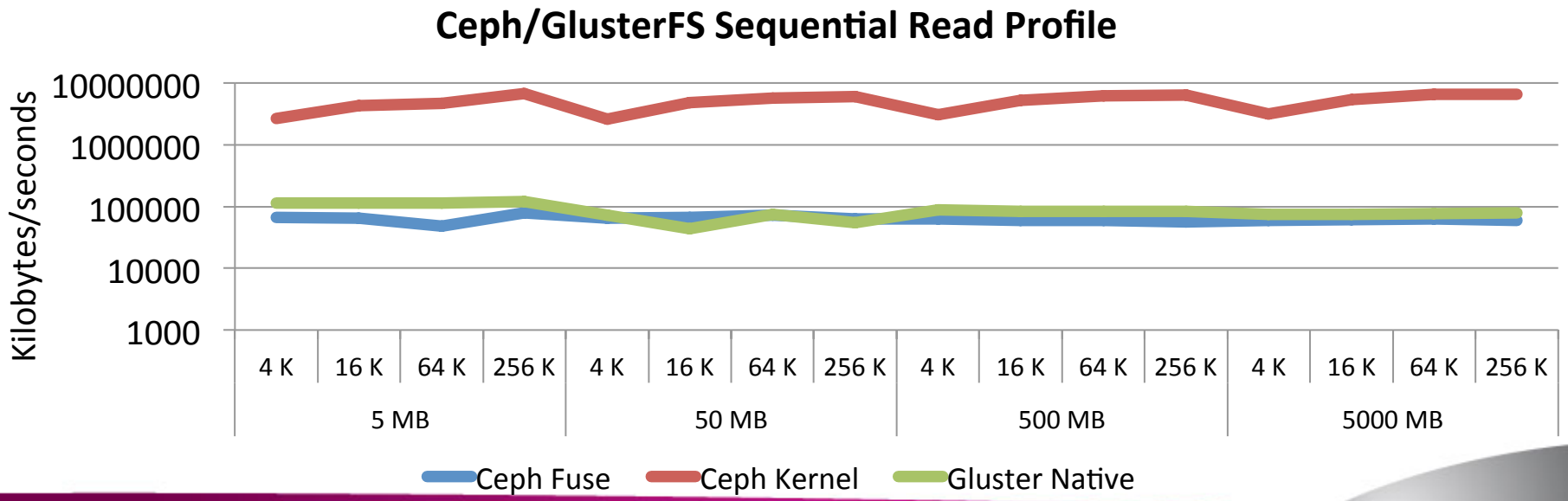
Experiment Hardware Specification



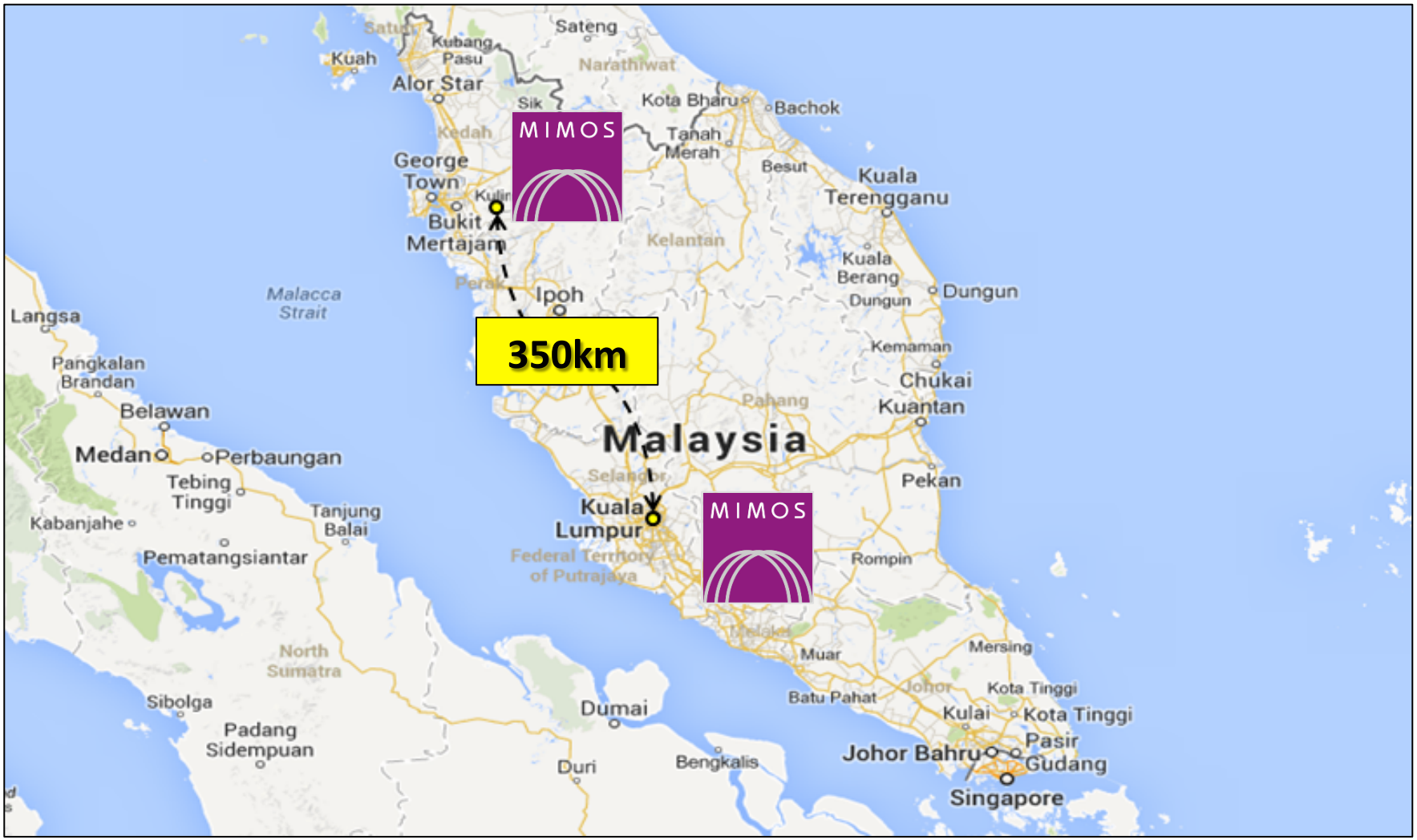
Experiment Network Setup

PRAGMA 25 – Ceph vs GlusterFS





PRAGMA 26 – DFS over WAN

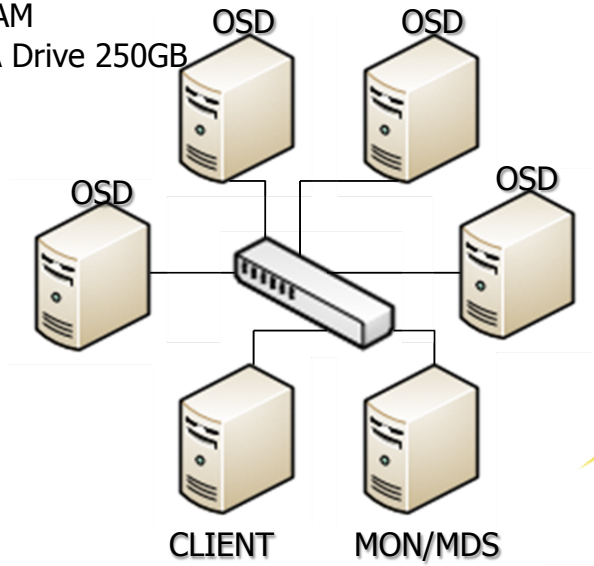


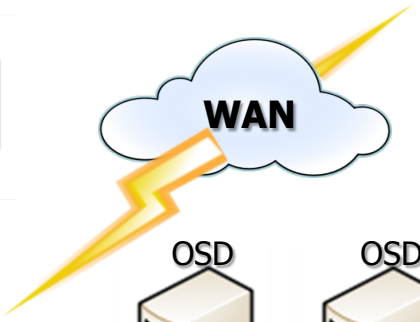
MIMOS Headquarters in Kuala Lumpur and its branch office in Kulim. From Google Map

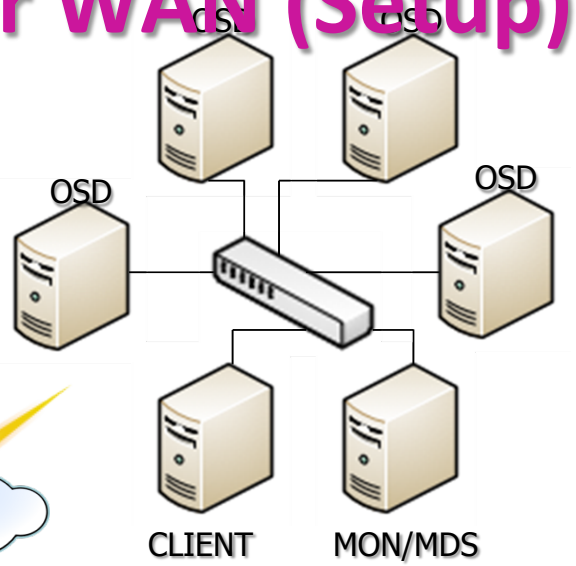
PRAGMA 26 – DFS over WAN (Setup)

Datacenter 1 in HQ office

SGI ALTIX XE 310
2 x Dual Core Intel Xeon CPU X5355 2.66 GHz
16 Gb RAM
3.5 SATA Drive 250GB

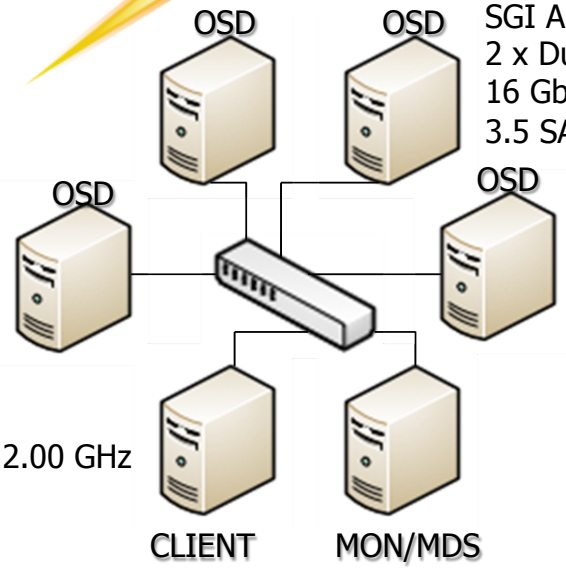






Datacenter 1 in Kulim office

SGI ALTIX XE 310
2 x Dual Core Intel Xeon CPU X5355 2.66 GHz
16 Gb RAM
3.5 SATA Drive 250GB



Datacenter 2 in HQ office

DELL
2 x Dual Core Intel Xeon CPU 5130 2.00 GHz
12 Gb RAM
3.5 SAS Drive 73GB

The storage pool was set with 3 replica counts with minimum number of replica counts required is 2.

PRAGMA 26 - DFS over WAN (Networking)

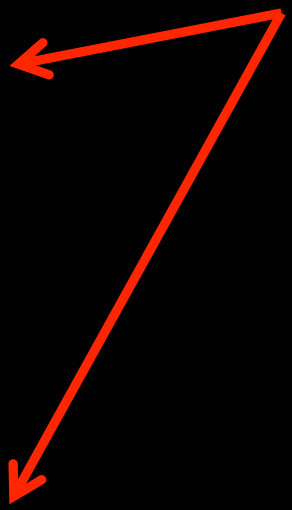
Round-trip time in ms	Bandwidth (Mbps)	2 TCP Iperf	Min	Avg	Max	Mdev
DC1 KL to DC1 Kulim	250	96%	13.149	13.491	16.167	0.684
DC2 KL to DC1 Kulim	250	96%	13.176	14.004	17.665	1.079
DC1 KL to DC2 KL	1000	86%	0.422	0.490	1.203	0.136

CRUSH Map - default

```
root@poc-tpm1-mon1:~/ceph-deploy# ceph osd tree
# id    weight  type name       up/down reweight
-1      2.12    root default
-2      0.23  host poc-tpm1-osd1
0       0.23        osd.0   up      1
-3      0.23  host poc-tpm1-osd2
1       0.23        osd.1   up      1
-4      0.23  host poc-tpm1-osd3
2       0.23        osd.2   up      1
-5      0.23  host poc-tpm1-osd4
3       0.23        osd.3   up      1
-6      0.06999   host poc-tpm2-osd1
4       0.06999       osd.4   up      1
-7      0.06999   host poc-tpm2-osd2
5       0.06999       osd.5   up      1
-8      0.06999   host poc-tpm2-osd3
6       0.06999       osd.6   up      1
-9      0.06999   host poc-tpm2-osd4
7       0.06999       osd.7   up      1
-10     0.23  host poc-khtp-osd1
8       0.23        osd.8   up      1
-11     0.23  host poc-khtp-osd2
9       0.23        osd.9   up      1
-12     0.23  host poc-khtp-osd3
10      0.23        osd.10  up      1
-13     0.23  host poc-khtp-osd4
11      0.23        osd.11  up      1
```

CRUSH Map Rules - default

```
# rules
rule data {
        ruleset 0
        type replicated
        min_size 1
        max_size 10
        step take default
        step chooseleaf firstn 0 type host
        step emit
}
rule metadata {
        ruleset 1
        type replicated
        min_size 1
        max_size 10
        step take default
        step chooseleaf firstn 0 type host
        step emit
}
```



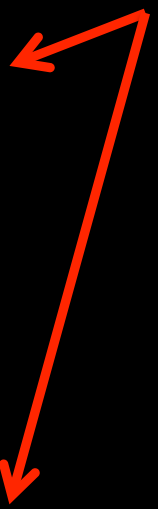
Pick one leaf node of type host

CRUSH Map - New

```
root@poc-tpm1-mon1:~/ceph-deploy# ceph osd tree
# id    weight  type name       up/down reweight
-1      2.12    root default
-23     0.92            datacenter tpm1
-2      0.23                host poc-tpm1-osd1
0       0.23                    osd.0   up      1
-3      0.23                host poc-tpm1-osd2
1       0.23                    osd.1   up      1
-4      0.23                host poc-tpm1-osd3
2       0.23                    osd.2   up      1
-5      0.23                host poc-tpm1-osd4
3       0.23                    osd.3   up      1
-24     0.28            datacenter tpm2
-6      0.06999             host poc-tpm2-osd1
4       0.06999                 osd.4   up      1
-7      0.06999             host poc-tpm2-osd2
5       0.06999                 osd.5   up      1
-8      0.06999             host poc-tpm2-osd3
6       0.06999                 osd.6   up      1
-9      0.06999             host poc-tpm2-osd4
7       0.06999                 osd.7   up      1
-25     0.92            datacenter khtp1
-10     0.23                host poc-khtp-osd1
8       0.23                    osd.8   up      1
-11     0.23                host poc-khtp-osd2
9       0.23                    osd.9   up      1
-12     0.23                host poc-khtp-osd3
10      0.23                    osd.10  up      1
-13     0.23                host poc-khtp-osd4
11      0.23                    osd.11  up      1
```

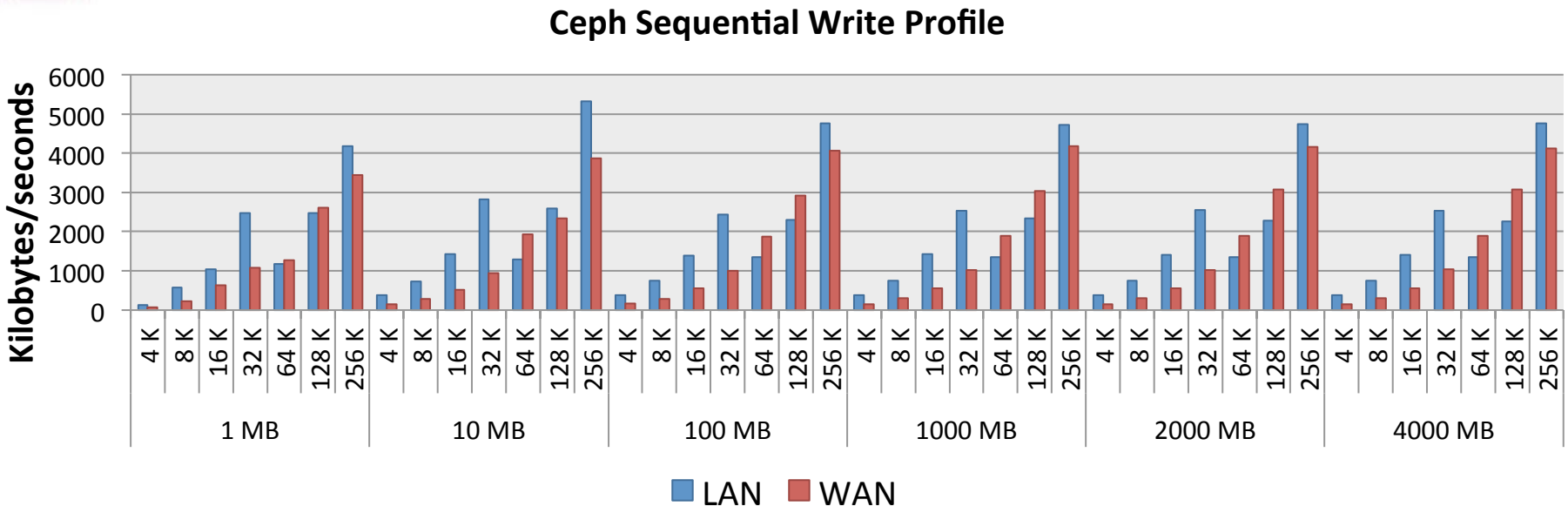
CRUSH Map Rules – New

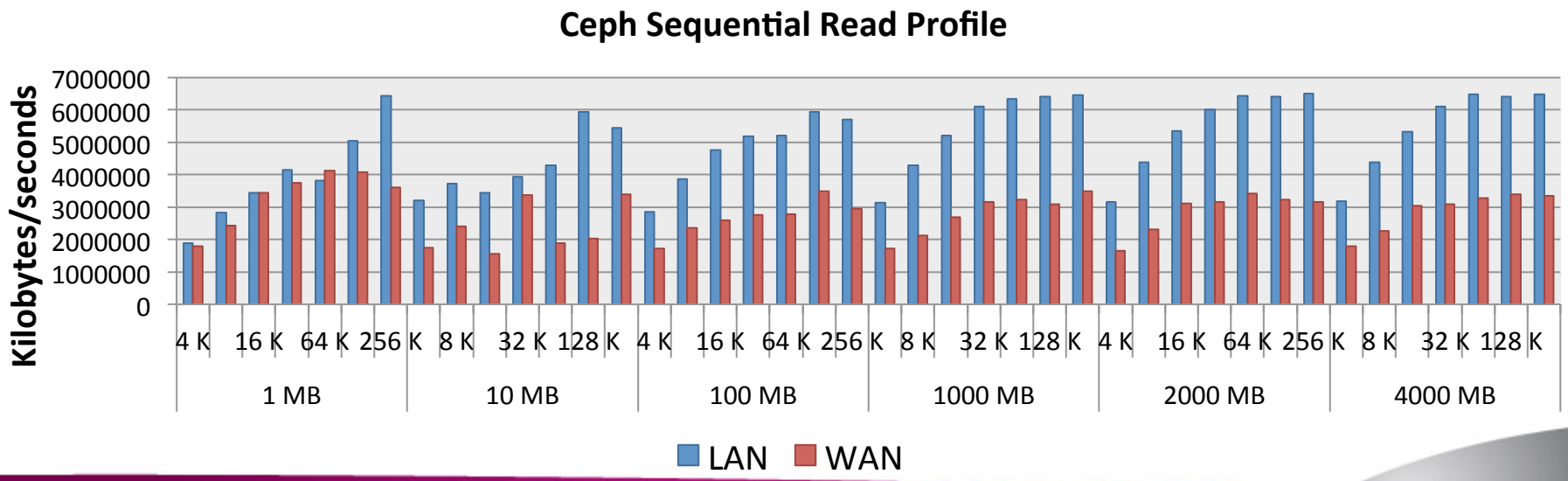
```
# rules
rule data {
        ruleset 0
        type replicated
        min_size 2
        max_size 10
        step take default
        step chooseleaf firstn 0 type datacenter
        step emit
}
rule metadata {
        ruleset 1
        type replicated
        min_size 2
        max_size 10
        step take default
        step chooseleaf firstn 0 type datacenter
        step emit
}
```



Pick one leaf node of type datacenter

DFS on WAN vs. DFS on LAN





Hypothesis:

- Write performance is slower than read performance. This is due to write operation requires a creation of new file and also to store overhead information known as metadata, which typically consists of directory information, and space allocation.
- DFS IO performs better in LAN compared to WAN due to limited capacity of WAN bandwidth and its latency, jitter etc.

Results:

- DFS in LAN provides better overall I/O rates compared to DFS in WAN due to its better network connectivity and bandwidth size.
- DFS in WAN scores better in writing 64K and 128K block sizes compared to DFS in LAN.

Analysis:

- DFS in WAN performances in I/O is still acceptable e.g. smaller files size with 16K, 32K, 64K, 128K block sizes, where DFS in LAN only performs slightly better than in WAN.

Summary

- Distributed file system in wide area network works at acceptable I/O rates and it is ideal for usage of smaller file sizes.

- Investigating distributed file system in wide area network, focusing on features like:
  - support cloud deployment architecture,
  - ability to provide parallel read and write operations on a distributed file system with different geographical locations.

TERIMA KASIH

THANK YOU

www.mimos.my

Innovation for Life™