Ceph Distributed File System: Simulating a Site Failure

Mohd Bazli Ab Karim, Ming-Tat Wong, Jing-Yuan Luke
Advanced Computing Lab
MIMOS Berhad, Malaysia

emails:
{bazli.abkarim, mt.wong, jyluke} @mimos.my

In PRAGMA 26, Tainan, Taiwan
9-11 April 2014

Innovation for Life™

Outline

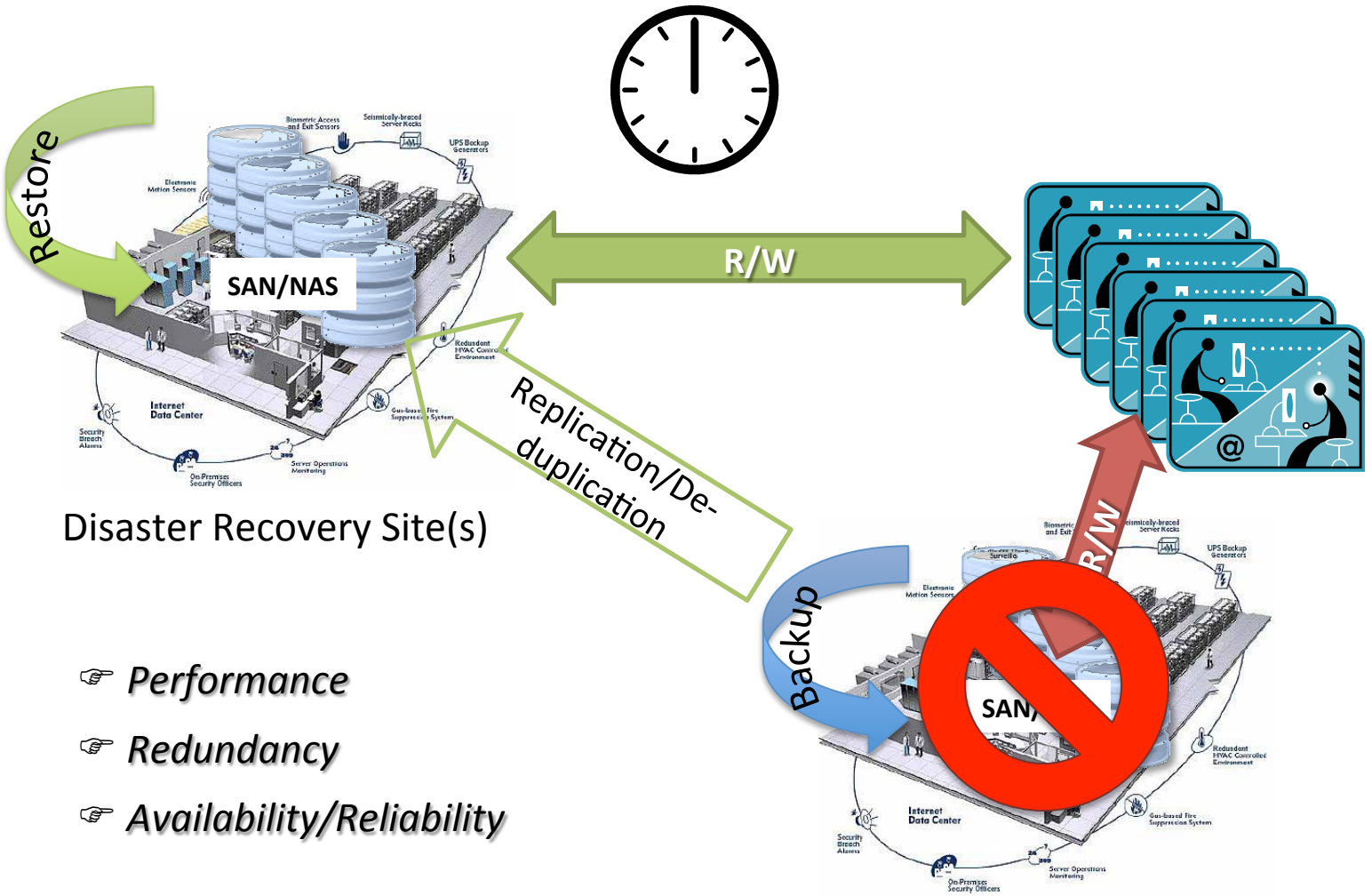
- Motivation
- Problems
- Solution
- Demo
- Moving forward

Motivations

- Explosion of both structured and unstructured data in cloud computing as well as in traditional datacenters presents a challenge for existing storage solution from cost, redundancy, availability, scalability, performance, policy, etc.

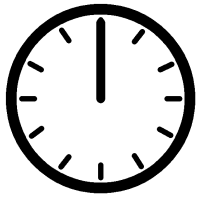
- Our motivation thus focus leveraging on ***commodity*** hardware/storage and networking to create a ***highly available*** storage infrastructure to support future cloud computing deployment in a ***Wide Area Network***, multi-sites/multi-datacenters environment.

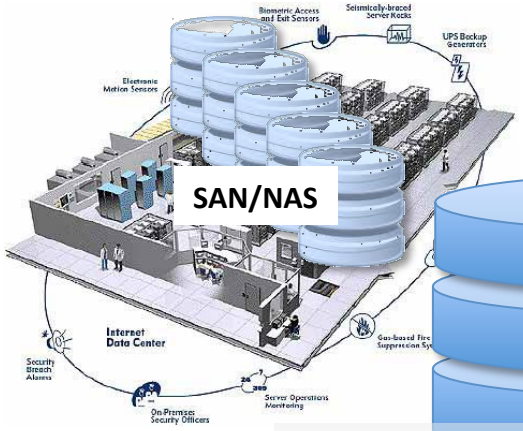
Problems



- *Performance*
- *Redundancy*
- *Availability/Reliability*

Solution



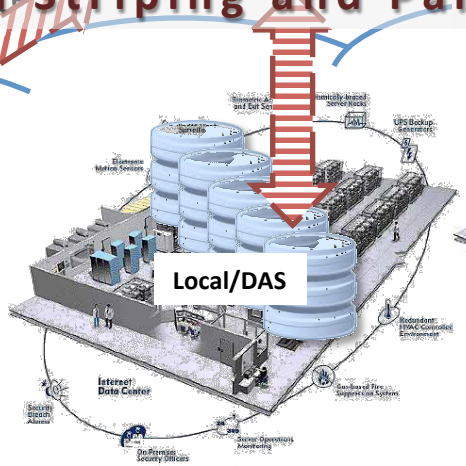


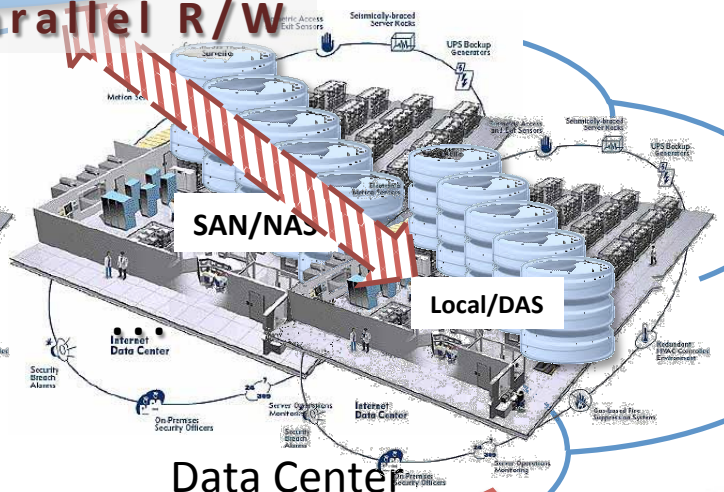
R/W

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

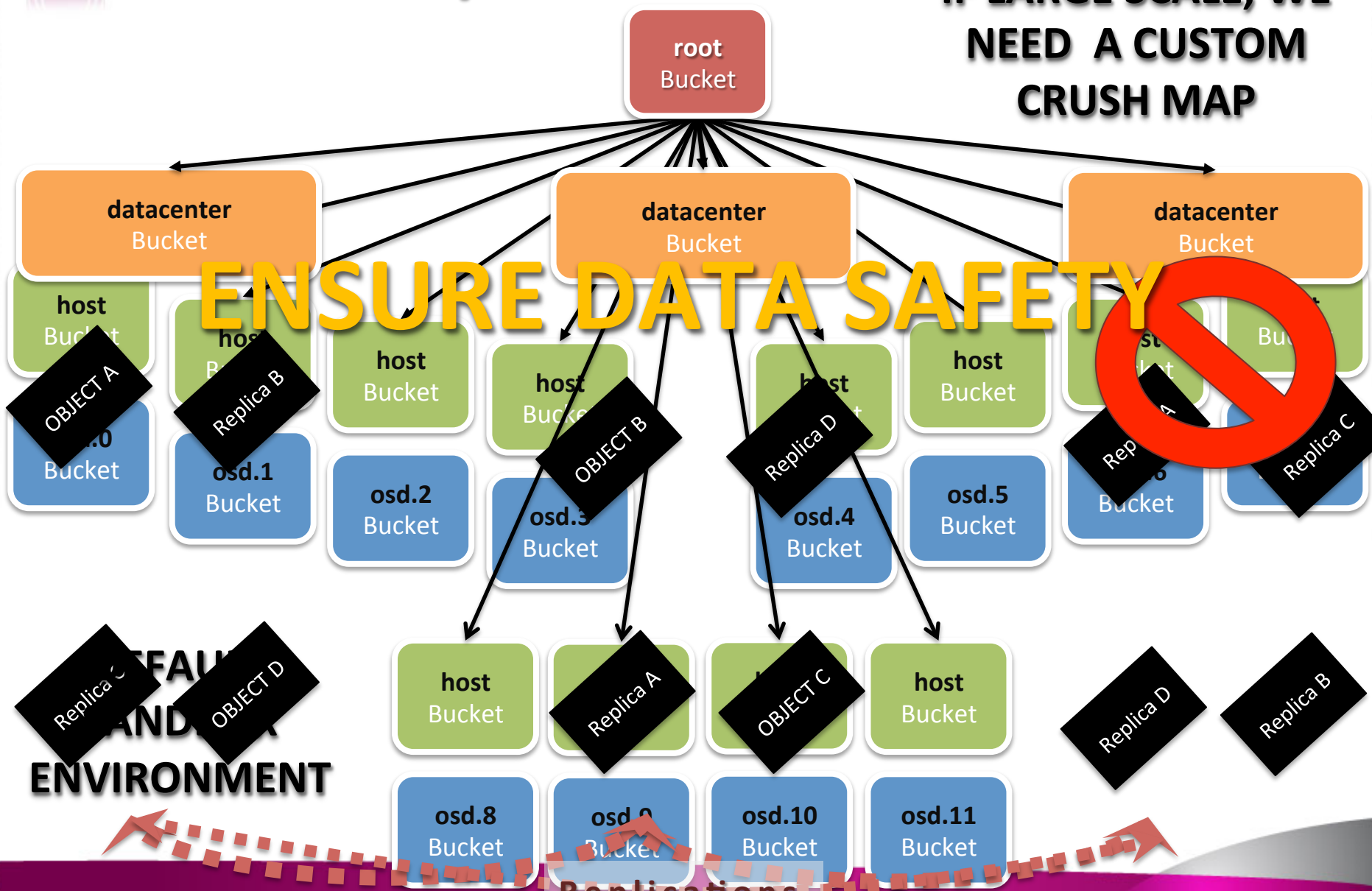
Replications

Challenging the CRUSH algorithm

- **CRUSH** – Controlled, Scalable, Decentralized Placement of Replicated Data
  - It is an algorithm to determine how to store and retrieve data by computing data storage locations.
- Why?
  - To use the algorithm to organize and distribute the data to different datacenters.

CRUSH Map

IF LARGE SCALE, WE NEED A CUSTOM CRUSH MAP

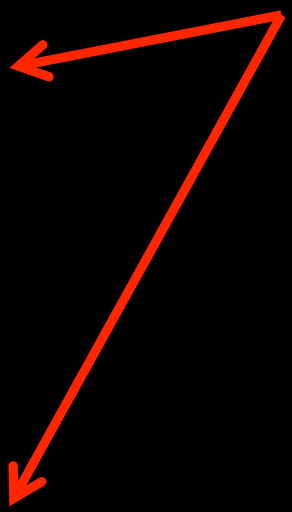


CRUSH Map - default

```
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
}
```

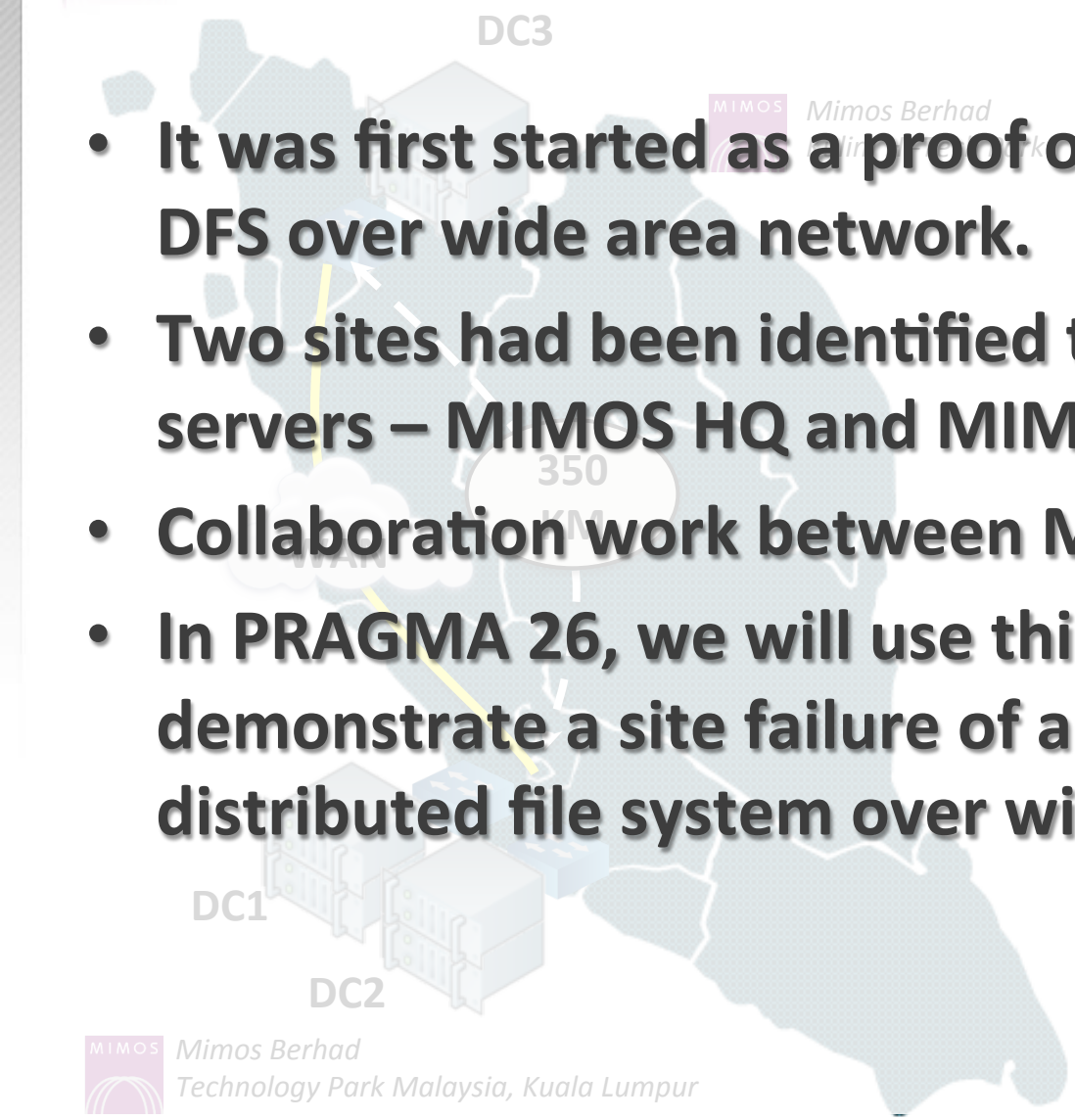
Pick one leaf node of type datacenter

DEMO

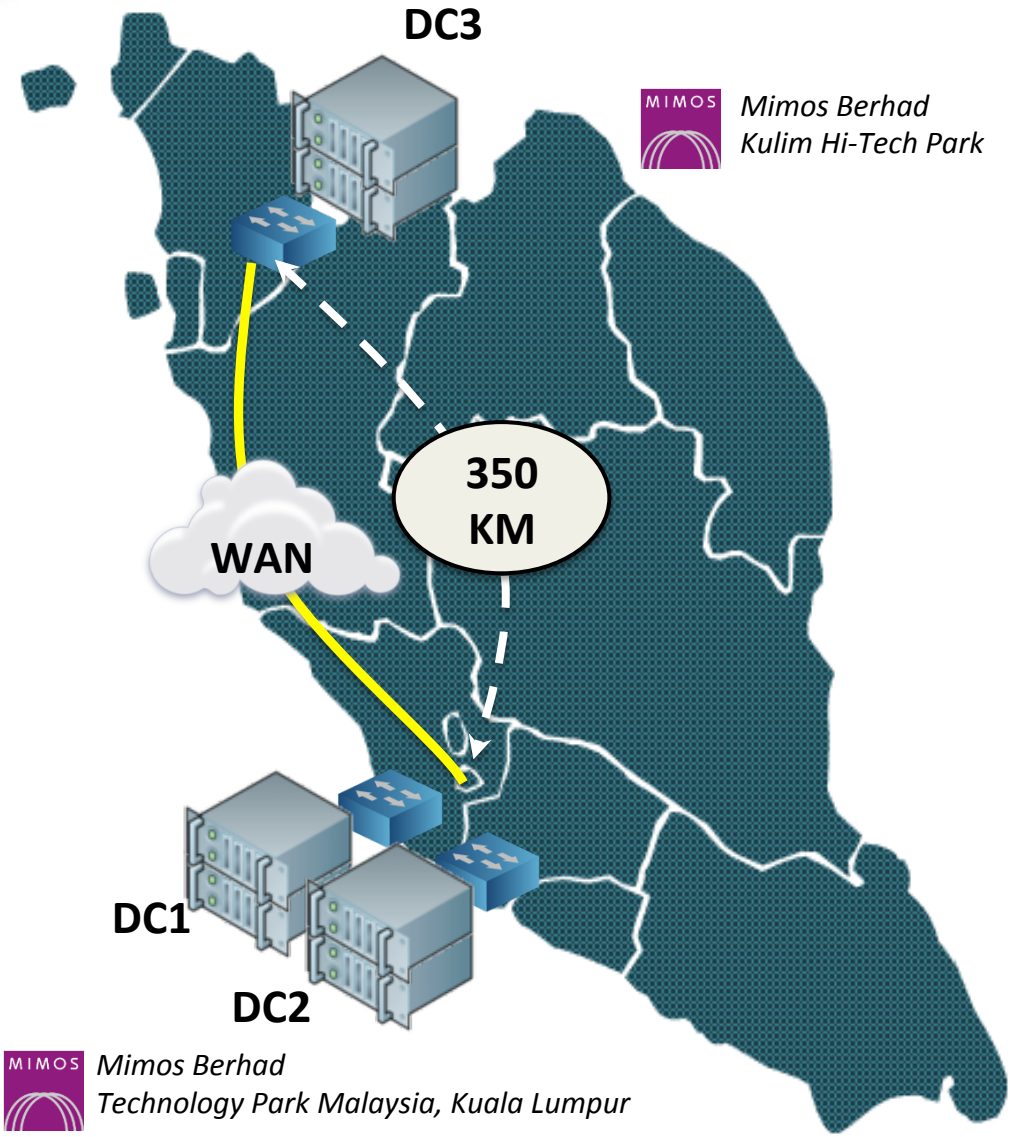
Innovation for Life™

Demo Background

- It was first started as a proof of concept for Ceph as a DFS over wide area network.
- Two sites had been identified to host the storage servers – MIMOS HQ and MIMOS Kulim
- Collaboration work between MIMOS and SGI.
- In PRAGMA 26, we will use this Ceph POC setup to demonstrate a site failure of a geo-replication distributed file system over wide area network.



This Demo…



Demo:

Simulate node/site failure while doing read write ops.

Test Plan:

(a) From DC1, continuously ping servers in Kulim.

(b) Upload 500Mb file to the file system.

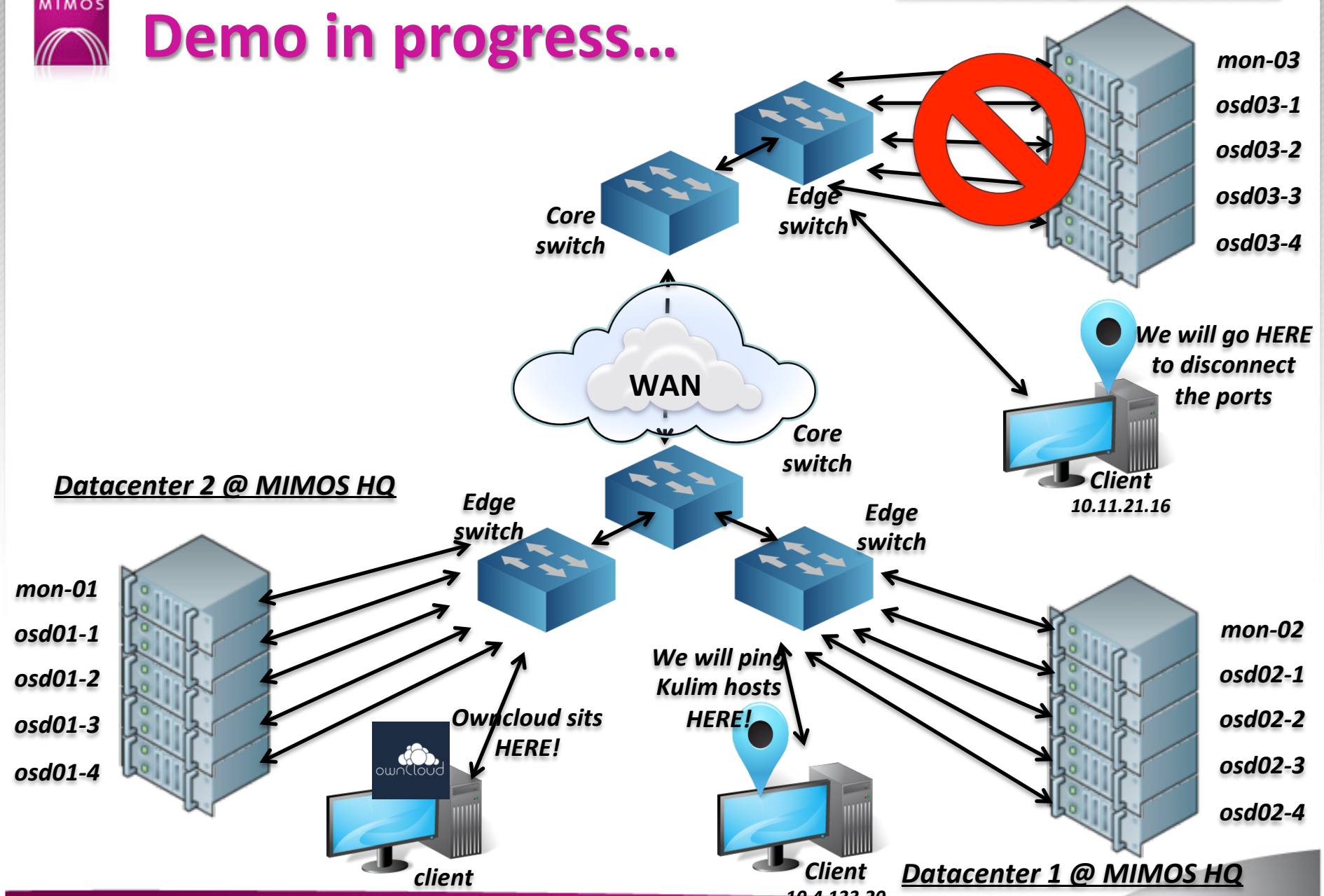
(c) While uploading, take down nodes in Kulim. From (a), check if nodes are down.

(d) Upload completed, download the same file.

(e) While downloading, bring up the nodes in Kulim.

(f) Checksum both files. Both should be same.

Demo in progress…



Moving forward…

- Challenges during POC which running on top of our production network infrastructure.
- Next, can we set up the distributed storage system with virtual machines plus SDN?
  - Simulate DFS performance over WAN in a virtualized environment.
  - Fine-tuning and run experiments: Client's file-layout, TCP parameters for the network, routing, bandwidth size/ throughput, multiple VLANs etc.

TERIMA KASIH

THANK YOU

www.mimos.my

Innovation for Life™