HITACHI
Inspire the Next

IC2E 2018

Feasibility Study of Location-Conscious Multi-Site Erasure-Coded Ceph Storage for Disaster Recovery

Keitaro Uehara* Hitachi Ltd.

Yih-Farn Robin Chen AT&T Labs-Research
Matti Hiltunen
Kaustubh Joshi
Richard Schlichting

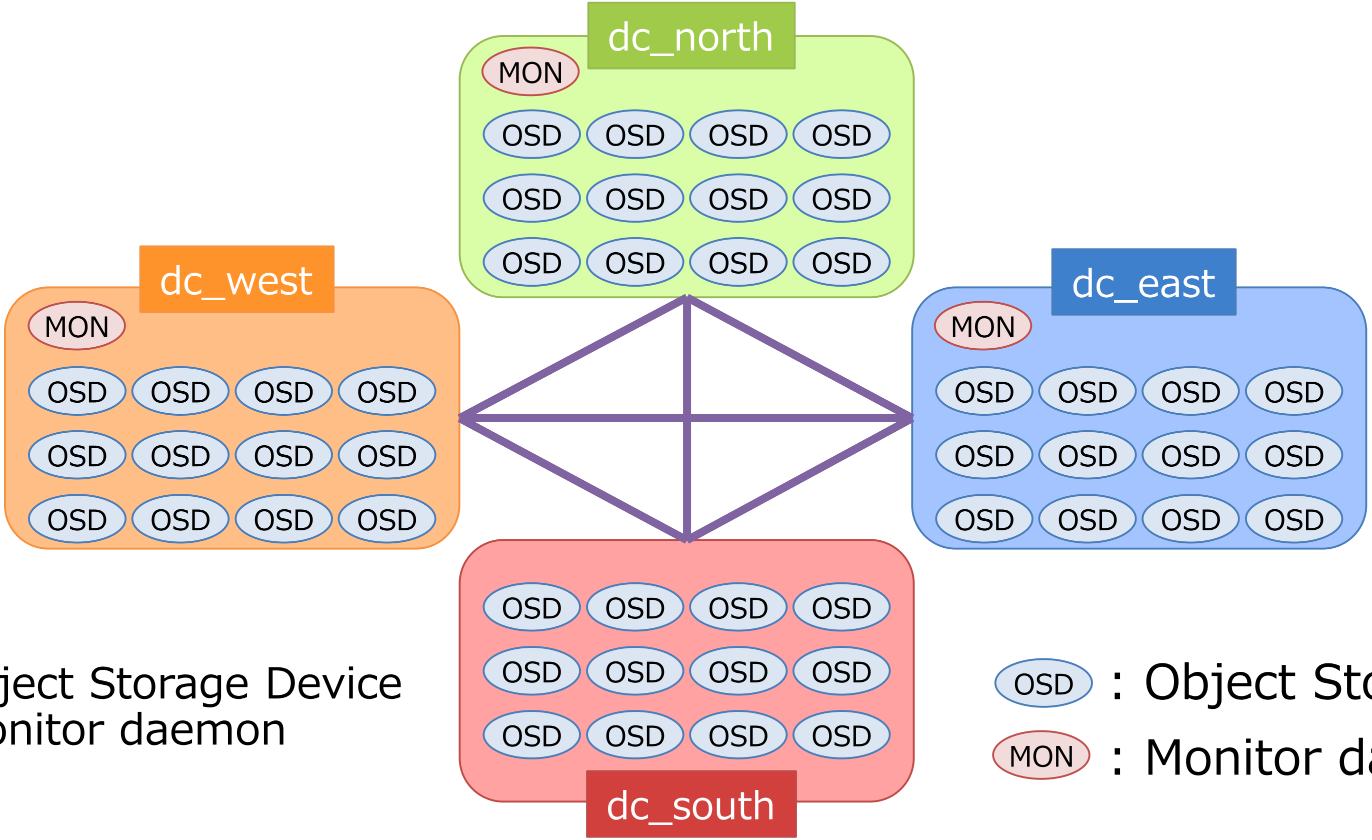
HITACHI
Inspire the Next

Background

Introduction

- Software-Defined Storage (SDS) is emerging.
  - Ceph is one of the most popular SDS open source project.
- To achieve high availability for disaster recovery, erasure code is a key technology.
- But performance drawback occurs in using erasure codes.
- We have studied the feasibility of Ceph's flexible mechanism to implement storage system with both high availability and performance improvement.

Assumption: 48 nodes in 4 data centers



OSD: Object Storage Device
MON: Monitor daemon

Assumptions and availabilities

HITACHI Inspire the Next

Assumptions for failure probabilities

Factor	Parameter
Node failure rate	Once per 4.3 months
Datacenter power outage	Once per year
Average disk life time	Three years
MTTR for node failure and DC power outage	One day
Target availability	99.999% (Five nines)

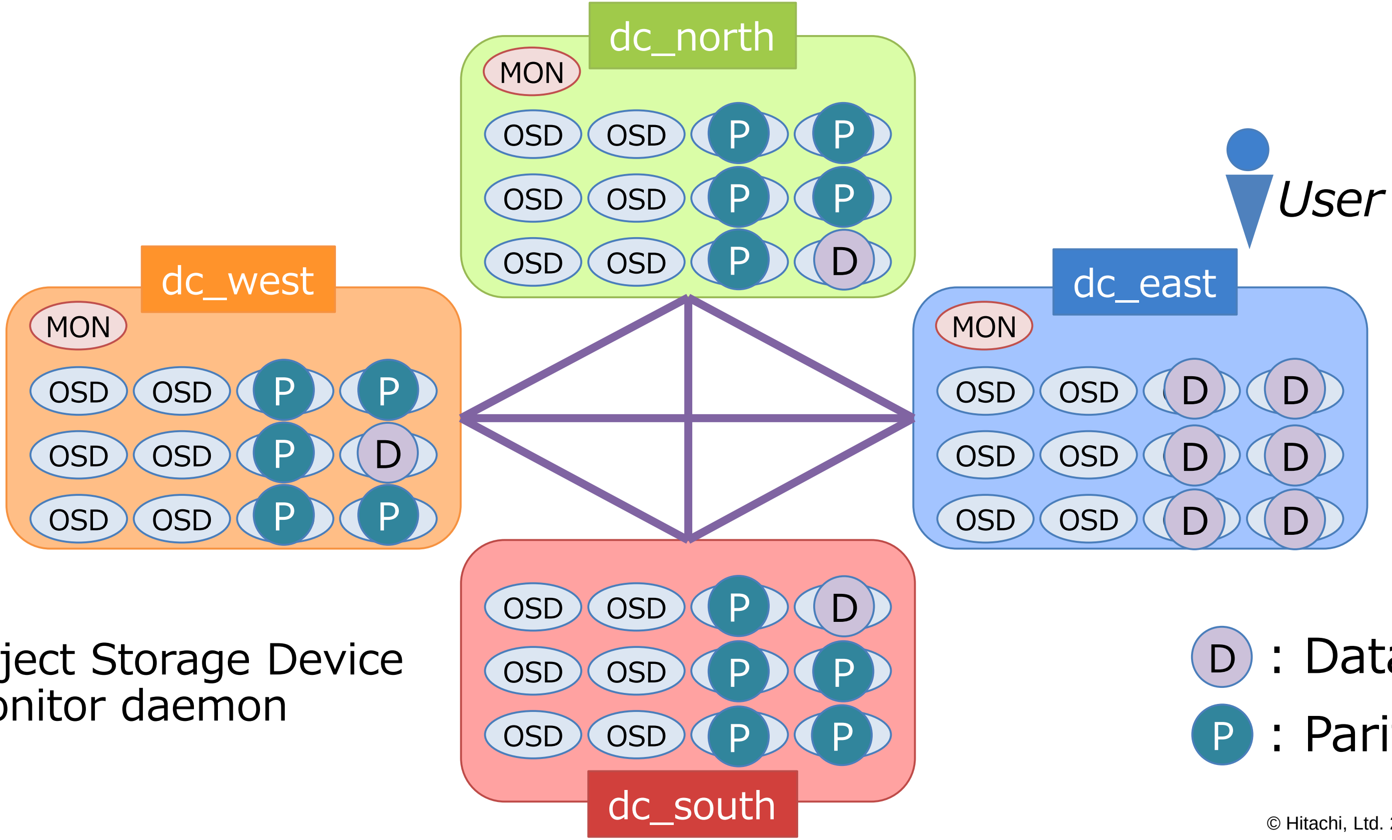
Availability comparison between x3 replication and **9+15 erasure code**

Failure cause	x3 replication	9+15 erasure code
simultaneous nodes failure	99.774% (3nodes failure in 3DC)	100% (16nodes failure)
1DC + nodes failure	99.978% (1DC + 2nodes failure)	100% (1DC + 7nodes failure)
2DC + nodes failure	99.999% (2DC + 1node failure)	99.999% (2DC + 2nodes failure)

Issue: Longer read latency in symmetric distribution

HITACHI
Inspire the Next

9+15 erasure code in symmetric distribution (6 chunks each) on 4 data centers



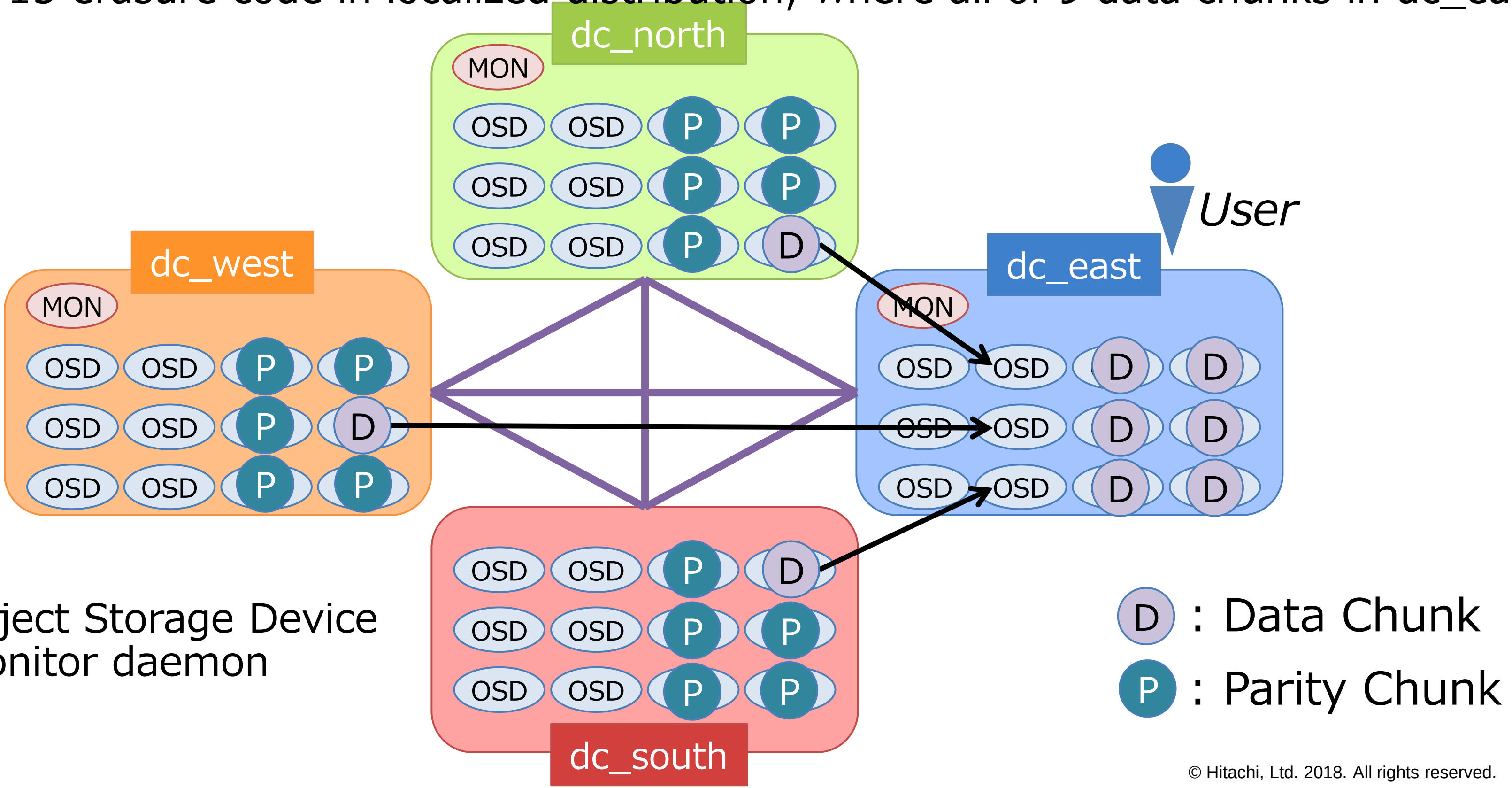
OSD: Object Storage Device
MON: Monitor daemon

D : Data Chunk
P : Parity Chunk

Solution: Asymmetric/localized distribution

HITACHI
Inspire the Next

9+15 erasure code in symmetric distribution (6 chunks each) on 4 data centers
-> 9+15 erasure code in localized distribution, where all of 9 data chunks in dc_east.

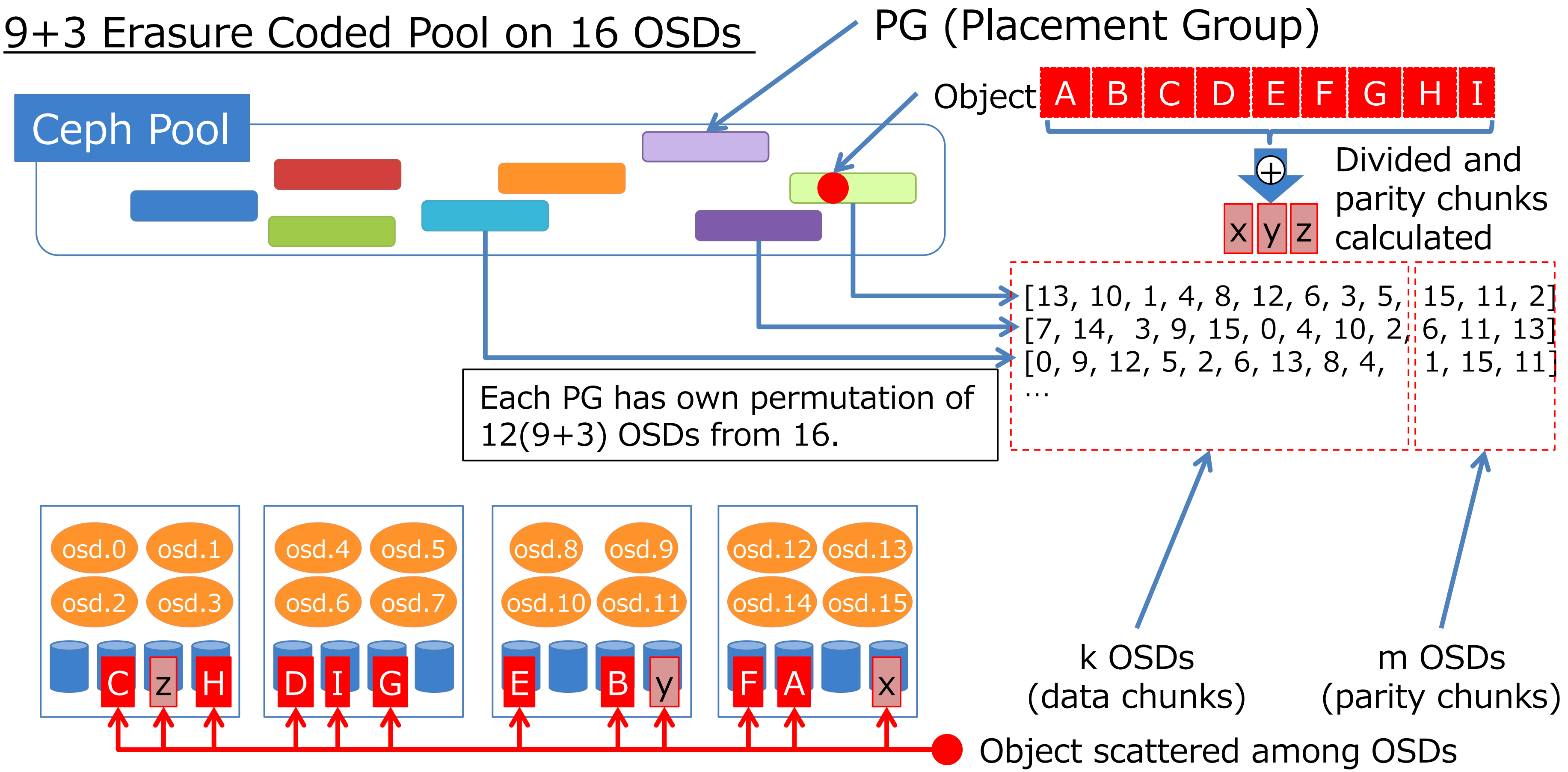


OSD: Object Storage Device
MON: Monitor daemon

Implementation with Ceph CRUSH map

Erasure Coded Pool on Ceph

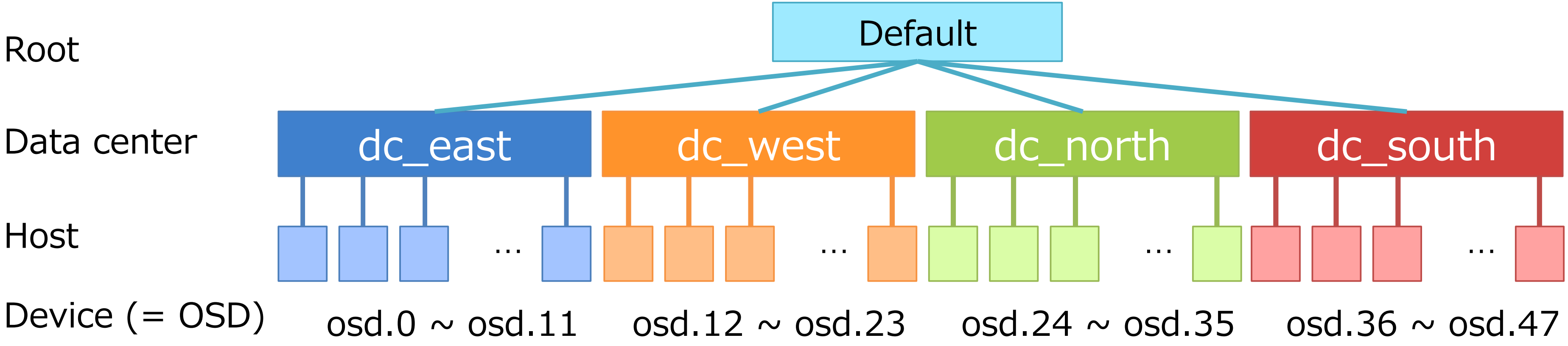
9+3 Erasure Coded Pool on 16 OSDs



Ceph CRUSH Map

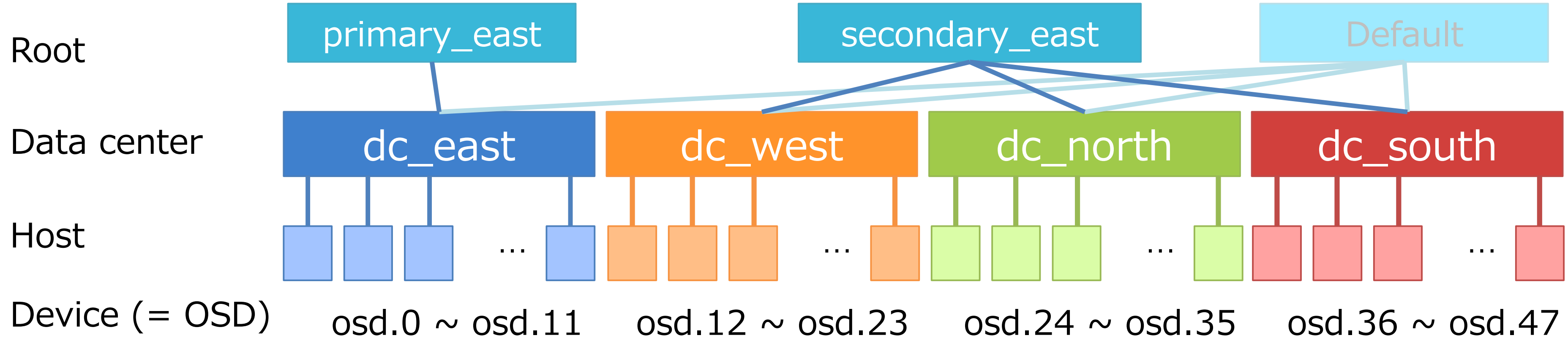
HITACHI
Inspire the Next

- Ceph provides CRUSH (Controlled Replication Under Scalable Hashing) map
- Define hierarchy of multiple layers
- Define "rule sets" for each pool to retrieve "OSD"s from hierarchy in recursive way to meet requirements of replications:
  - In x3 Replication, 3 OSDs required to be chosen.
  - In 9+15 Erasure Code, 24 OSDs required to be chosen.



Ceph CRUSH Map for EC with Primary Affinity

- We define two different kinds of “root” for East DC as primary DC
  - “primary_east” includes only “dc_east”
  - “secondary_east” includes the other three DCs.



Ceph CRUSH Map for EC with Primary Affinity

HITACHI
Inspire the Next

- Define ruleset for 9+15 EC:
  - take first 9 chunks from different hosts under “primary_east”
  - take 3 DCs from “secondary_east”, then take 5 hosts under each DC.

```
root primary_east {
    id -54
    alg straw
    hash 0
    item dc_east weight 12
}
root secondary_east {
    id -55
    alg straw
    hash 0
    item dc_west weight 12
    item dc_north weight 12
    item dc_south weight 12
}
```

```
rule primary_ec_ruleset {
    ruleset 2
    type erasure
    min_size 9
    max_size 48
    step set_chooseleaf_tries 5
    step take primary_east
    step chooseleaf indep 9 type host
    step emit
    step take secondary_east
    step choose firstn 3 type datacenter
    step chooseleaf indep 5 type host
    step emit
}
```

Experiments of placement with crushtool

Tests of CRUSH Map with crushtool

HITACHI
Inspire the Next

- Ceph provides “crushtool”, which enables users to test user-defined CRUSH maps without actual Ceph cluster environment.
- Automatically produce 1024 patterns (default) of object placement, and show statistics or bad-mappings.

```
$ crushtool -c test-crushmap.txt -o test-crushmap.bin
$ crushtool -i test-crushmap.bin --test --rule 2 --num_rep 24
  --output_csv
```

crushtool test Results: Placement Information

Chunk Order →

k = 9 (data chunks) (chunks 0–8)   m = 15 (parity chunks) (chunks 9–23)

Placement Group# ↓

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	3	2	9	10	11	8	4	7	0	19	14	22	16	23	30	35	27	34	28	42	43	47	45	41
1	4	11	3	0	5	10	1	8	7	38	46	43	41	36	13	19	21	20	23	26	25	33	31	27
2	2	5	7	11	3	0	1	4	10	46	41	38	39	43	29	28	32	25	24	19	14	18	16	23
3	8	3	1	4	11	0	7	2	6	25	31	32	34	29	14	23	12	15	22	36	38	47	43	45
4	6	9	7	0	4	5	2	11	8	22	12	19	16	18	45	36	46	37	40	33	29	34	24	26
5	2	9	4	10	0	8	1	5	6	16	17	15	20	14	43	38	42	45	46	28	24	29	25	27
6	5	2	11	9	10	3	1	0	4	37	41	40	47	45	32	29	33	30	24	17	16	21	12	13
7	8	0	6	2	5	4	10	1	3	15	22	14	23	16	27	28	33	26	34	40	45	43	39	41
8	4	0	3	9	6	10	1	7	11	40	44	42	41	45	24	30	27	34	28	15	22	16	12	21
9	2	7	11	5	10	8	0	9	3	37	45	47	41	38	26	27	25	28	29	16	15	20	22	17
:																								

datacenter	dc_east	dc_west	dc_north	dc_south
Device ID	0 ~ 11	12 ~ 23	24 ~ 35	36 ~ 47

crushtool test Results: Device Utilization

- Total number of object stored for each device (OSD) in 1024 patterns.
- In symmetric distribution, 1024 * 24 / 48 = 512 is the expected value.
- First 12 devices (in East DC) has been more chosen than the others due to primary affinity.

Device ID	0	1	2	3	4	5	6	7	8	9	10	11
# of Stored	793	769	774	768	778	763	745	748	773	754	777	774
Device ID	12	13	14	15	16	17	18	19	20	21	22	23
# of Stored	456	421	425	418	432	403	414	438	434	433	441	405
Device ID	24	25	26	27	28	29	30	31	32	33	34	35
# of Stored	433	402	428	424	435	433	443	410	420	429	443	420
Device ID	36	37	38	39	40	41	42	43	44	45	46	47
# of Stored	416	419	433	466	404	423	439	439	399	429	432	421

datacenter	dc_east	dc_west	dc_north	dc_south
Device ID	0 ~ 11	12 ~ 23	24 ~ 35	36 ~ 47

Experiments of I/O traffic with iostat

Experiments overview

HITACHI
Inspire the Next

Target

- To confirm Ceph's activity of reading erasure codes in normal condition.
  - Whether parity chunks are always read or not.

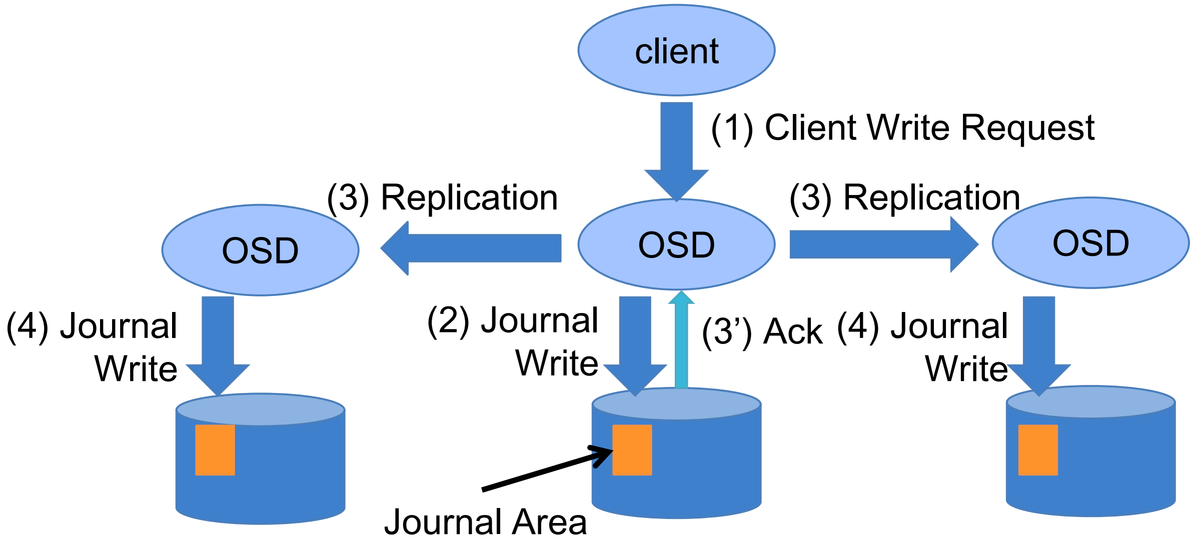
Method

- To aggregate "iostat" of volumes on each physical host.
- Write 50MB single object to 9+3 erasure coded pool (on VM).
- Flush VM caches (from both VMs and physical hosts).
- Read 50MB single object from erasure coded pool.

Ceph data write sequence (1/2)

HITACHI
Inspire the Next

In x3 replication case:

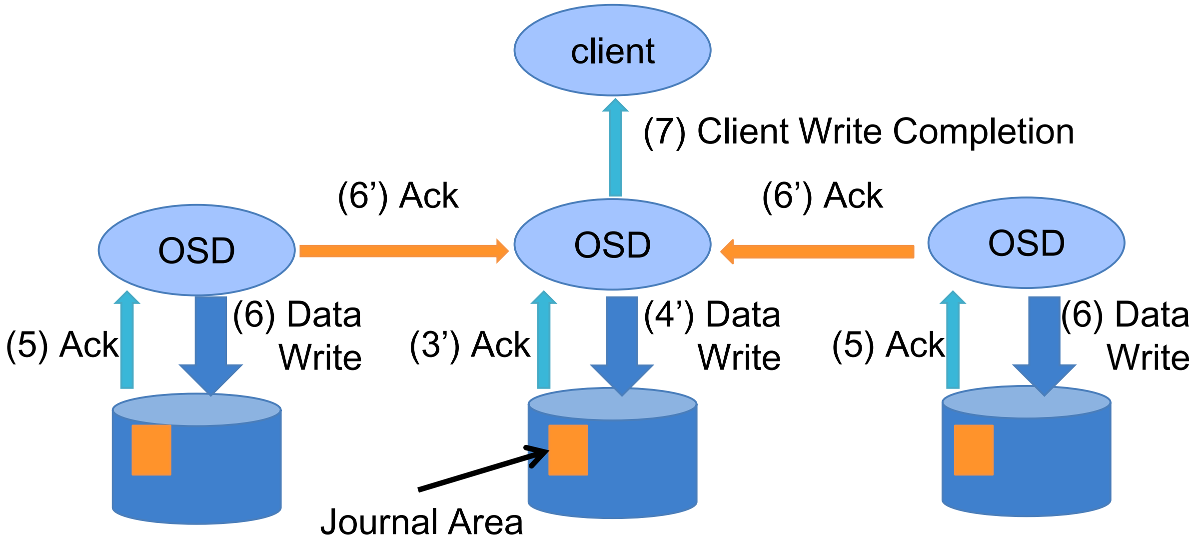


Each OSD writes to Journal prior to Data,
to reduce write latency with keeping durability.

Ceph data write sequence (2/2)

HITACHI
Inspire the Next

In x3 replication case: x6 write traffic occurs



(Journal + Data) x (x3 replication)

2 x 3

Experimental Results

- OSD Placement Group Map: [9, 5, 13, 1, 11, 2, 10, 14, 4, 15, 12, 3]
- Expected Write Amount: 50MB * (9+3) / 9 * 2 (Data + Journal) = 133.3 MB
- Expected Read Amount: 50MB (if only data chunks are read)
  or 66.7MB (if parity chunks are always read)

	Data Chunks									Parity Chunks			Non Related OSDs				Total
osd.id	9	5	13	1	11	2	10	14	4	15	12	3	0	8	6	7	
osdec	4b	2b	3c	1c	4a	1d	2a	3b	2d	3a	3d	1b	1a	2c	4c	4d	
write [MB]	10.83	10.84	10.75	10.74	10.74	10.83	10.84	10.83	10.85	10.82	10.83	10.81	0.1	0.1	0	0.1	130.01
read [MB]	5.98	5.93	5.68	5.99	6	6	5.93	5.81	5.96	0	0	0	0	0	0	0	53.28

- Writes are almost equally distributed to data and parity chunks OSDs.
- All of reads are from data chunks OSDs, no parity chunks.

Conclusion

Conclusion and Future Work

HITACHI
Inspire the Next

Conclusion

- From the experimental results, our proposed erasure code could be applied to satisfy both high availability and improvement of read performance.

Future Work

- To deploy a large storage system in four geographically-distant data centers based on the proposed erasure code scheme.

HITACHI
Inspire the Next