Cooling Down Ceph

Exploration and Evaluation of Cold Storage Techniques

Cold Storage
Ceph

Cooling Down Ceph
Object Stubs
Striper Prefix Hashing

"Cold storage is an operational mode or a method operation of a data storage device or system for **inactive data** where an explicit trade-off is made, resulting in data **retrieval response times beyond what may be considered normally acceptable** to online or production applications in order to archive significant capital and operational **savings**"

– IDC Technology Assessment: Cold Storage Is Hot Again - Finding the Frost Point (2013)

- Facebook photos [1]:
  82% reads to 8% stored data

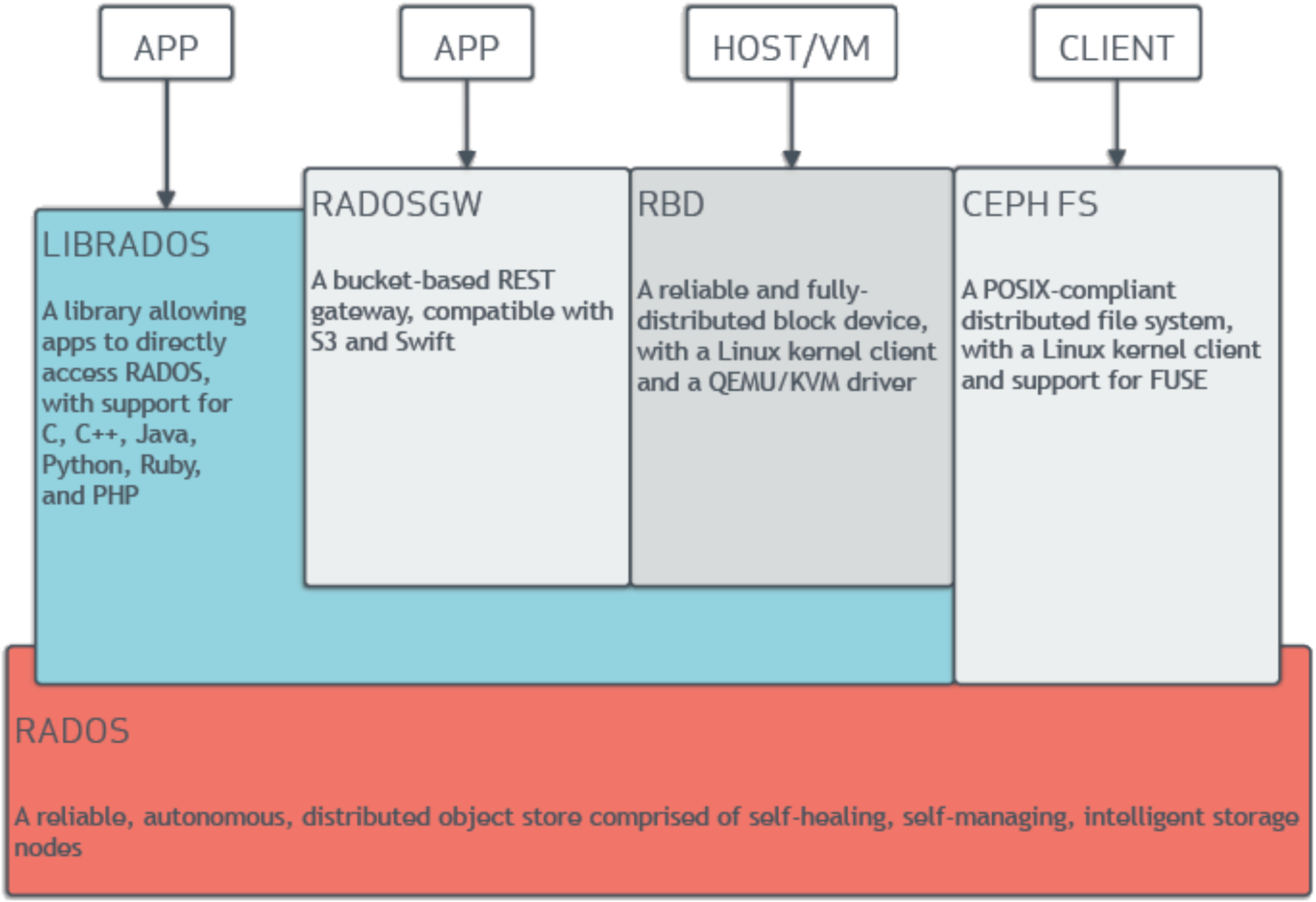
- Scientific data system [2]:
  50% reads to 5% stored data

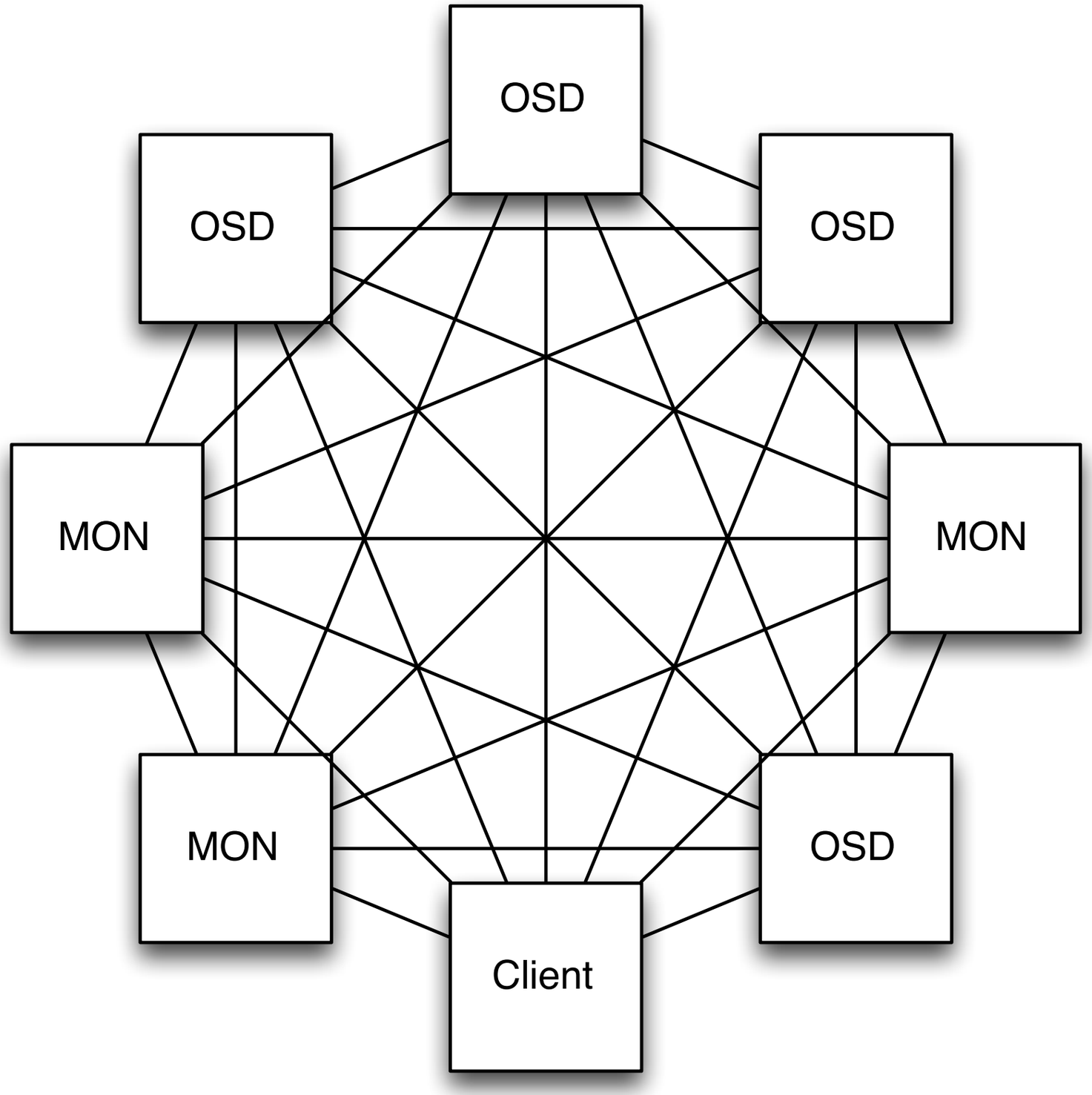
[1] T. P. Morgan. Facebook Rolls Out New Web and Database Server Designs. http://www.theregister.co.uk/2013/01/17/open_compute _facebook_servers/, 2013

[2] M. Grawinkel, L. Nagel, M. Mäsker, F. Padua, A. Brinkmann, and L. Sorth. Analysis of the ECMW Storage Landscape. *Proc. of the 13th USENIX Conference on File and Storage Technologies (FAST)*, 2015

ceph

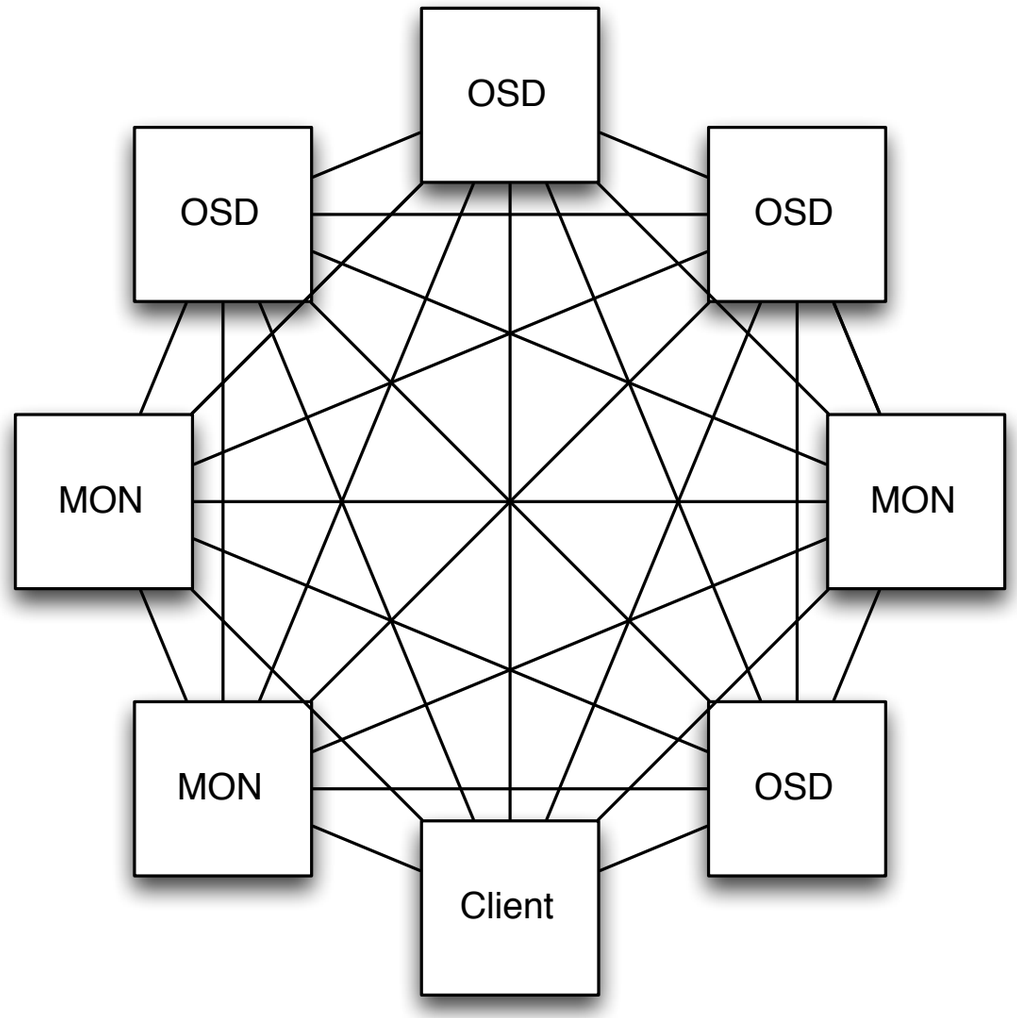
- Distributed storage system
- No single point of failure
- Horizontal scaling
- Run on commodity hardware


APP
APP
HOST/VM
CLIENT
LIBRADOS
A library allowing apps to directly access RADOS, with support for C, C++, Java, Python, Ruby, and PHP
RADOSGW
A bucket-based REST gateway, compatible with S3 and Swift
RBD
A reliable and fully-distributed block device, with a Linux kernel client and a QEMU/KVM driver
CEPH FS
A POSIX-compliant distributed file system, with a Linux kernel client and support for FUSE
RADOS
A reliable, autonomous, distributed object store comprised of self-healing, self-managing, intelligent storage nodes


OSD
OSD
OSD
MON
MON
MON
OSD
Client

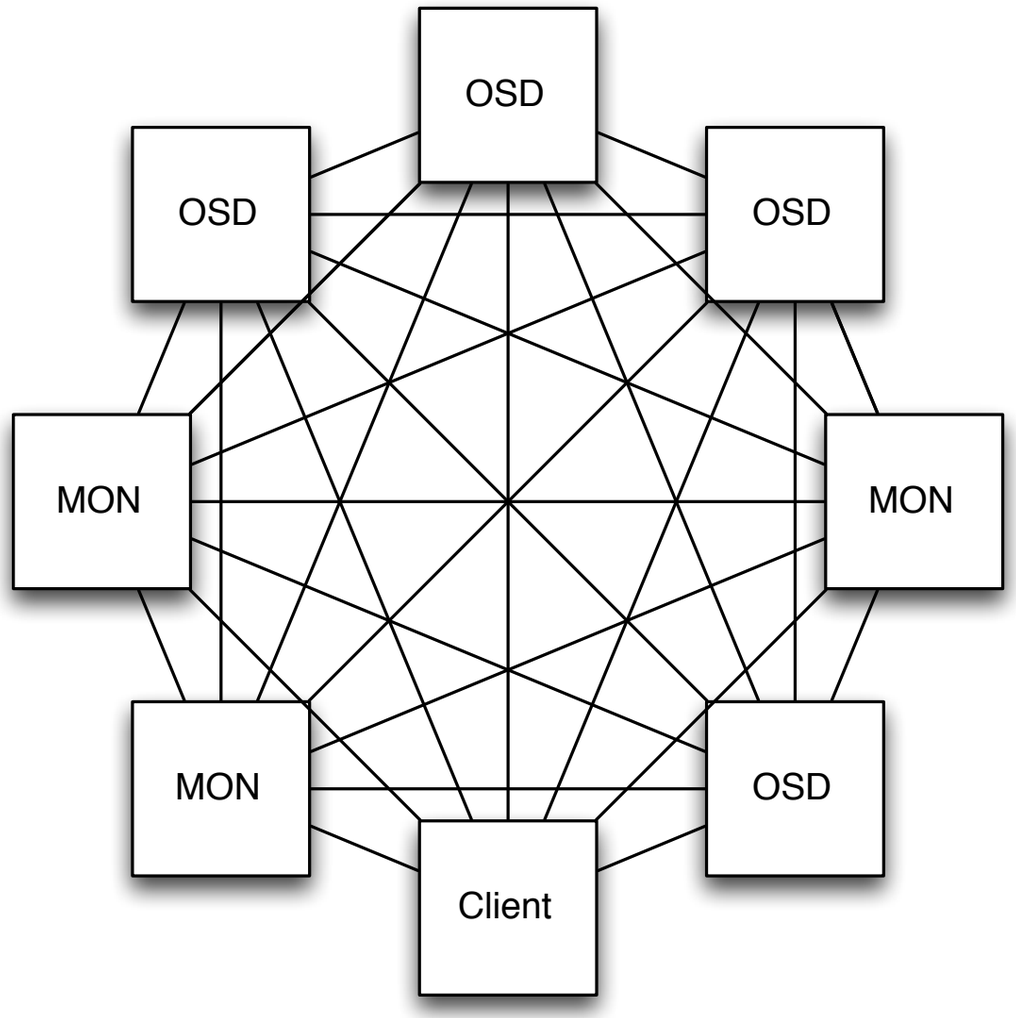
Monitor

- Keeps the Cluster Map
- Distributed Consensus
- Not in data path



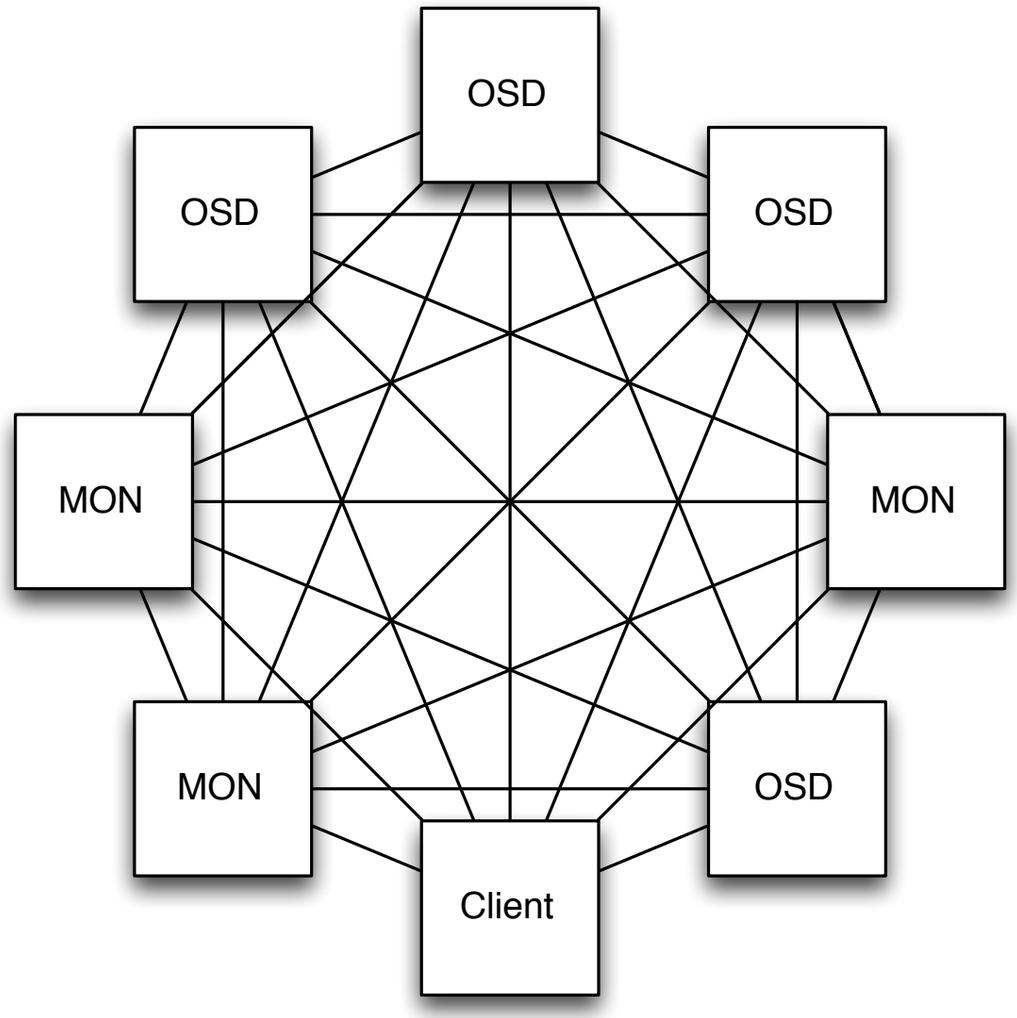
Client

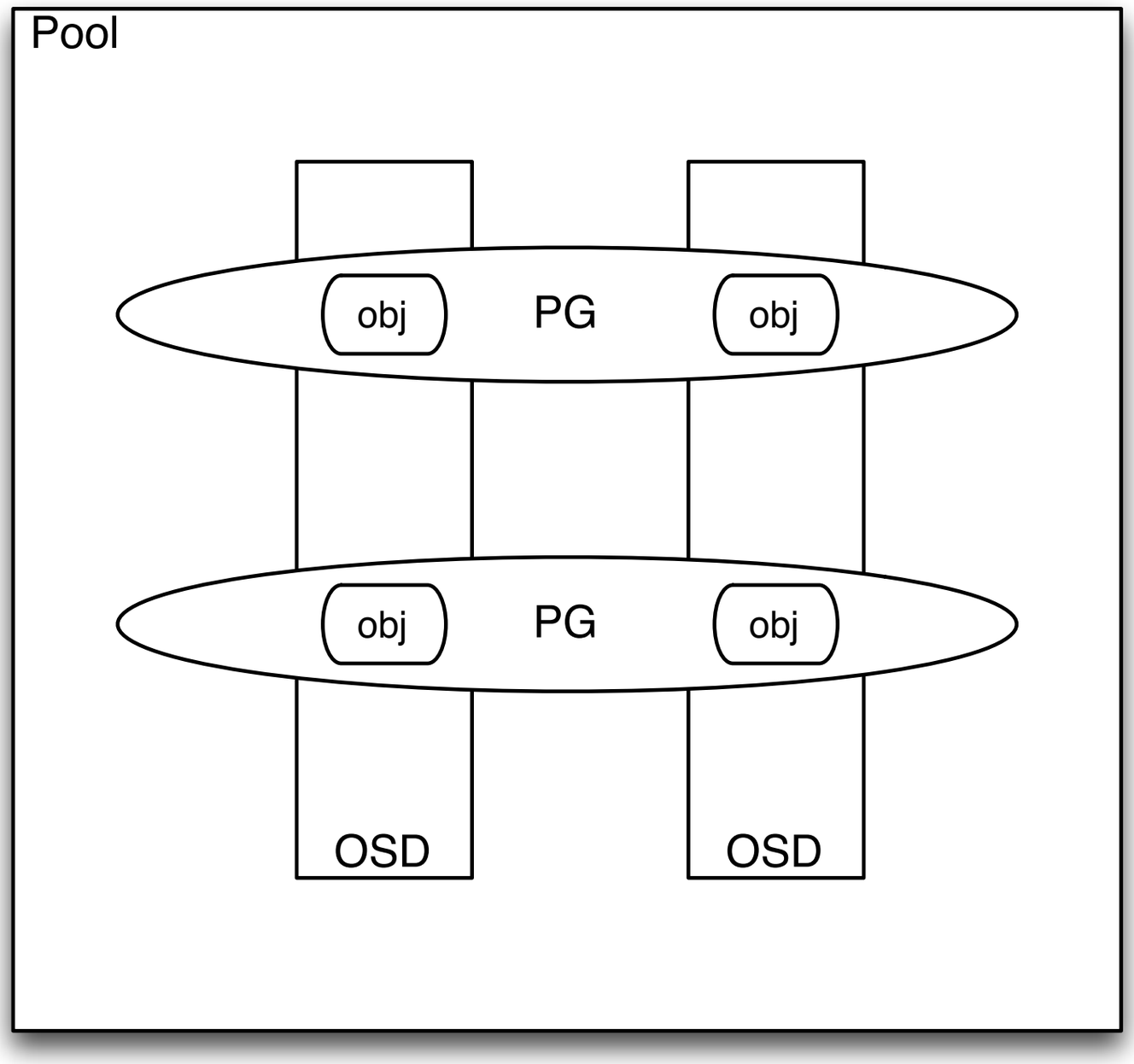
- Computes placement based on Cluster Map
- Directly accesses OSDs and MONs



OSD

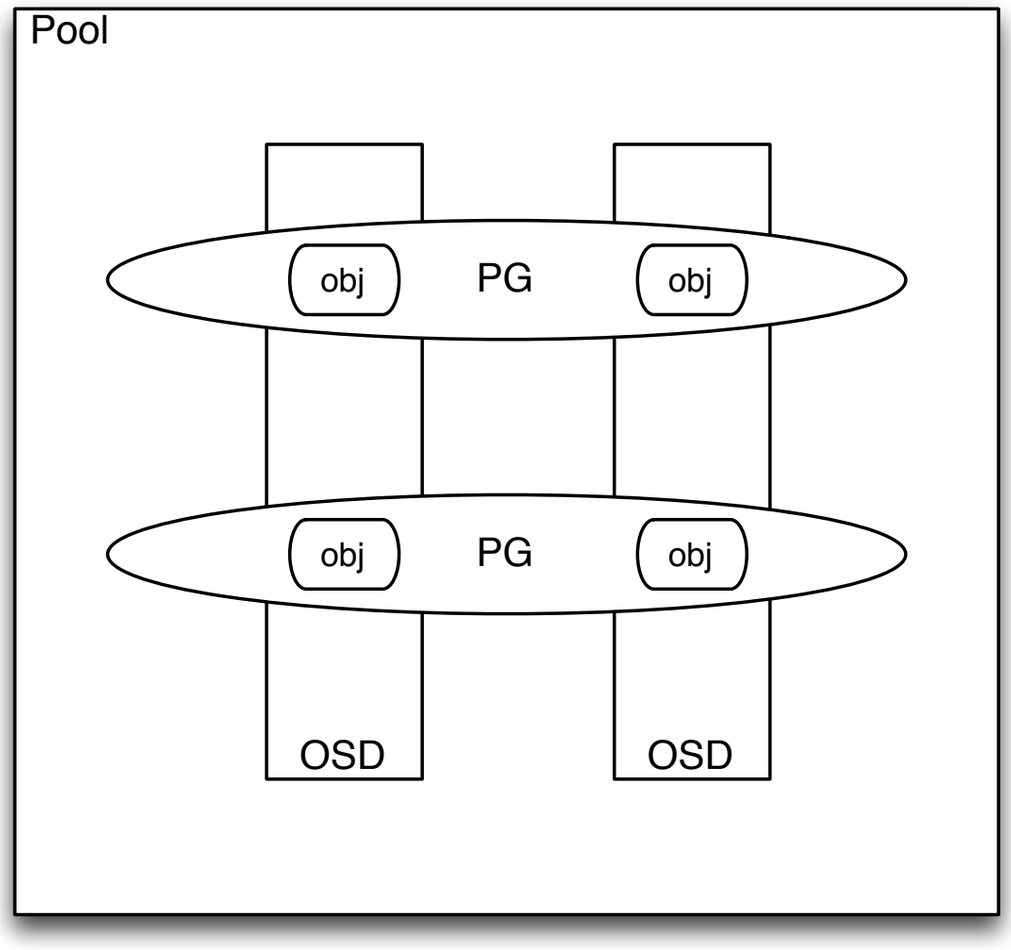
- Stores Objects
- Manages replication
  - Placement
- OSD ~ Disk
- Backends
  - Filesystem
  - Key/Value Store
  - Ethernet drives




Pool
obj
PG
obj
obj
PG
obj
OSD
OSD

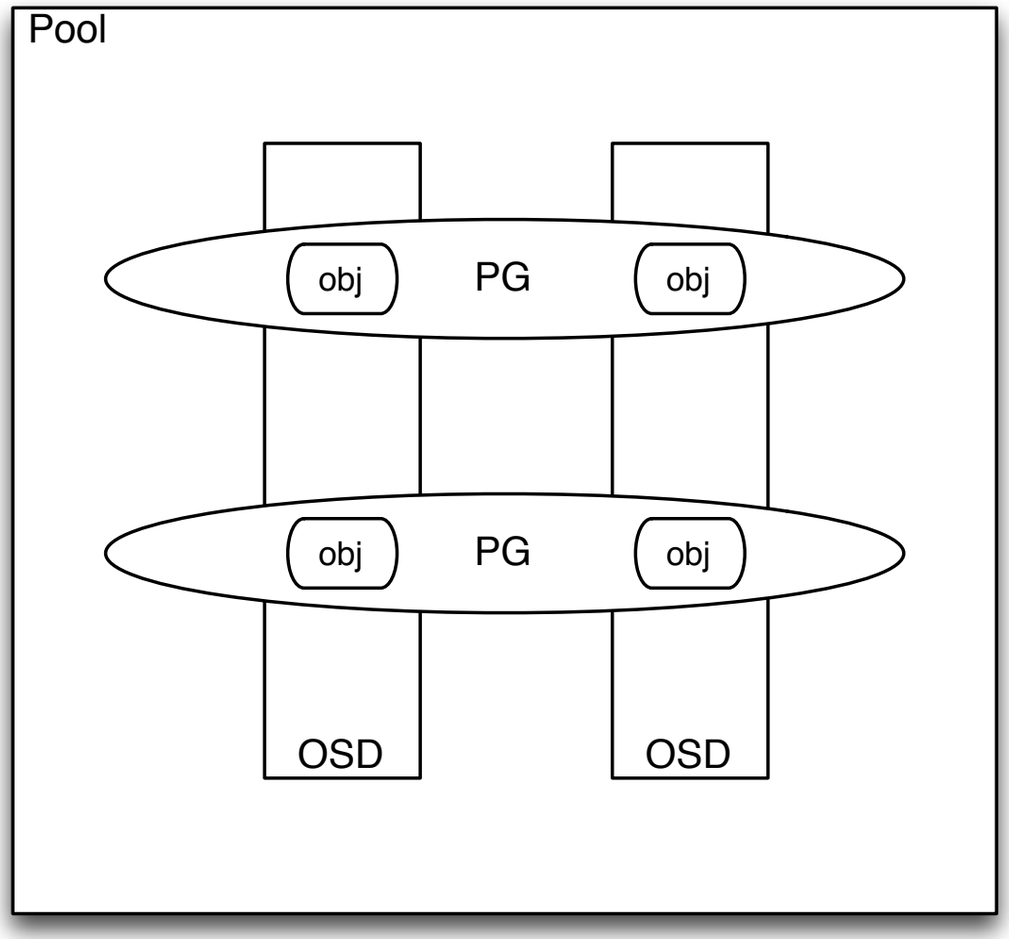
Pool

- OSDs
  - Buckets
    - Type
      - Rack, Server, Disk, …
- Type
  - Replicated
  - Erasure Coded



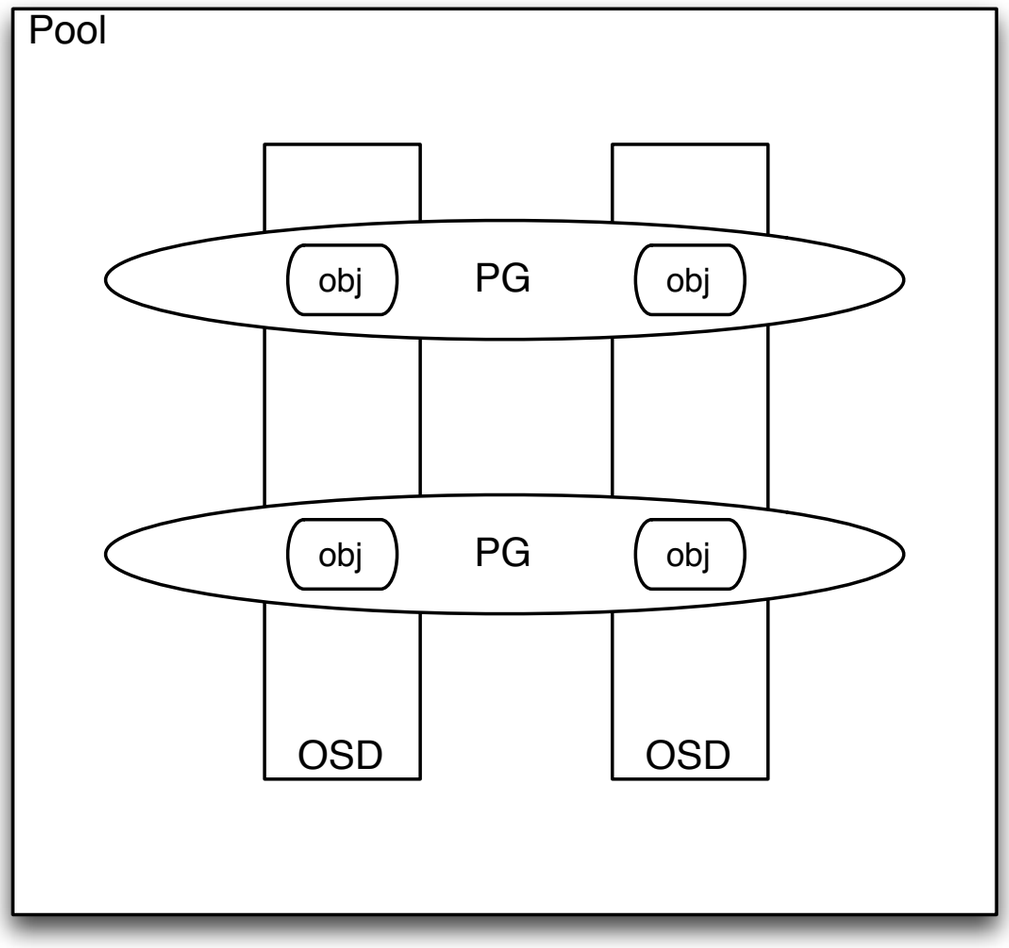
Object

- Data
  - 4 MiB
- Name
- Xattrs
- Object Map

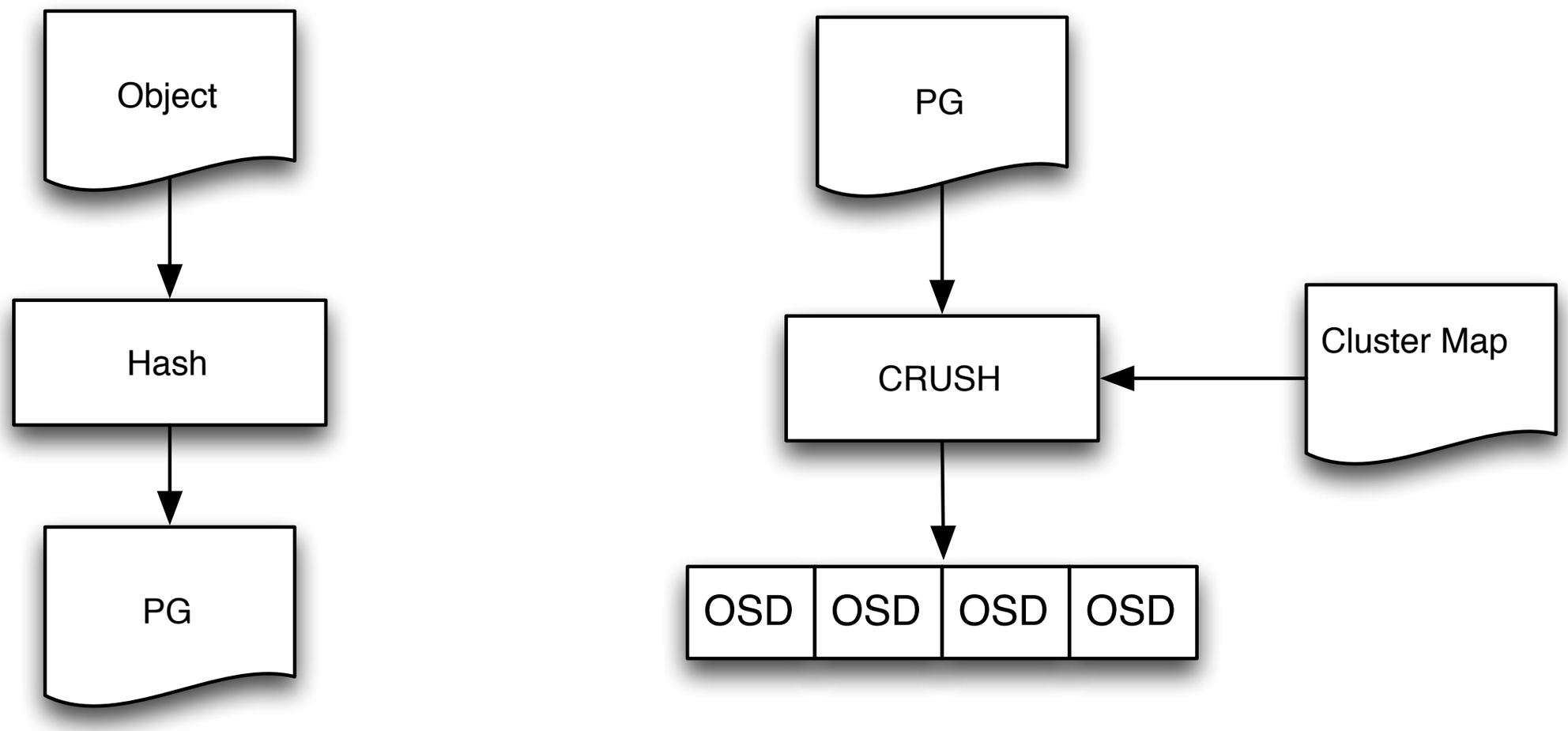


Placement Groups

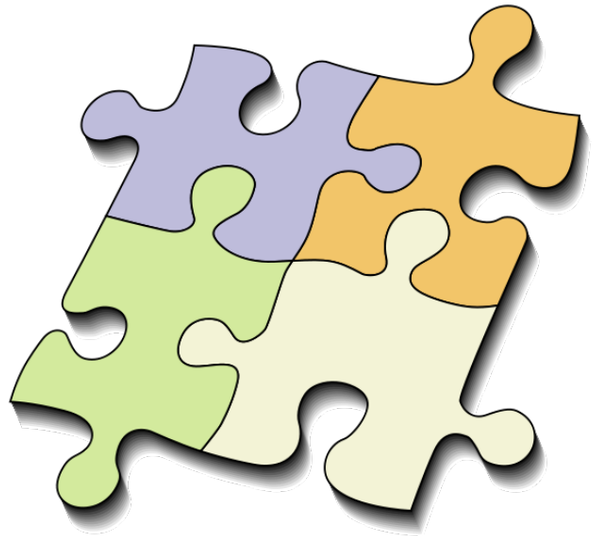
- Abstraction for placement computation
- ~ 100 per OSD



Placement



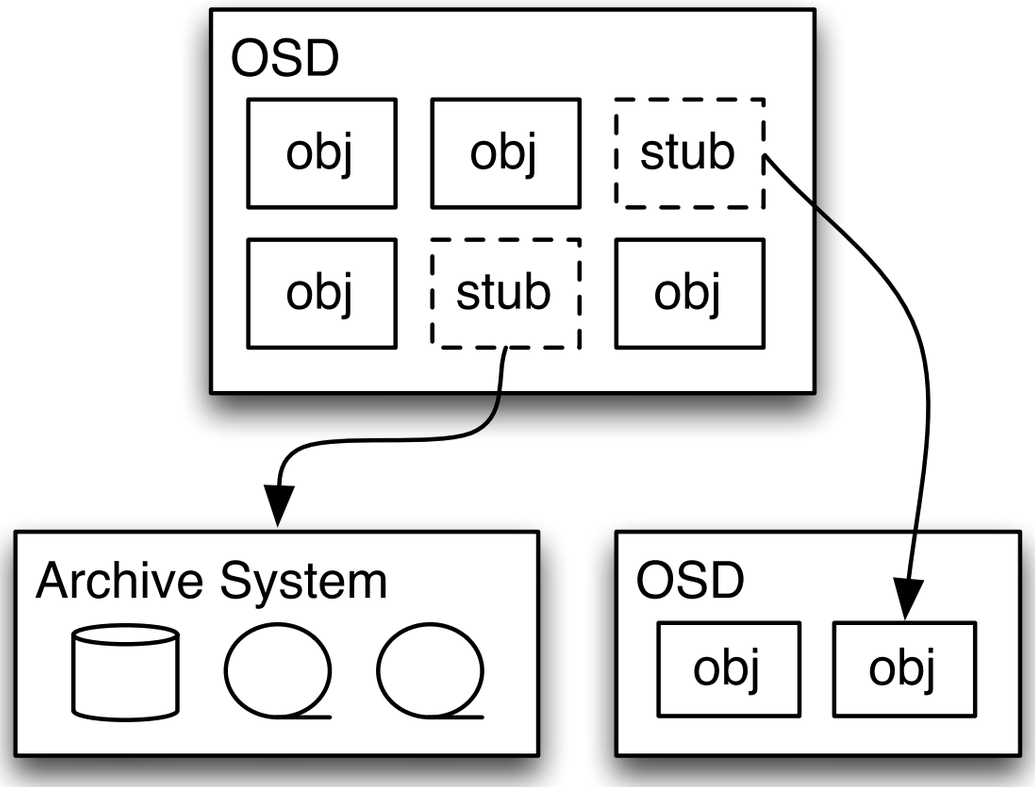
Cooling Down Ceph



- Ceph's Cold Storage Features
  - Cache Tiering
  - Erasure Coding
- Metadata-aware clients
  - Semantic Pool Selection
  - Metadata for Later
- Placement
  - Striper Prefix Hashing
  - Extra Placement Information
- Redirection and Stubbing
  - Object Redirects
  - Object Stubs
- Object Store
  - Backend to Archive System
  - Journal Cache

- Ceph's Cold Storage Features
  - Cache Tiering
  - Erasure Coding
- Metadata-aware clients
  - Semantic Pool Selection
  - Metadata for Later
- Placement
  - **Striper Prefix Hashing**
  - Extra Placement Information
- Redirection and Stubbing
  - Object Redirects
  - **Object Stubs**
- Object Store
  - Backend to Archive System
  - Journal Cache

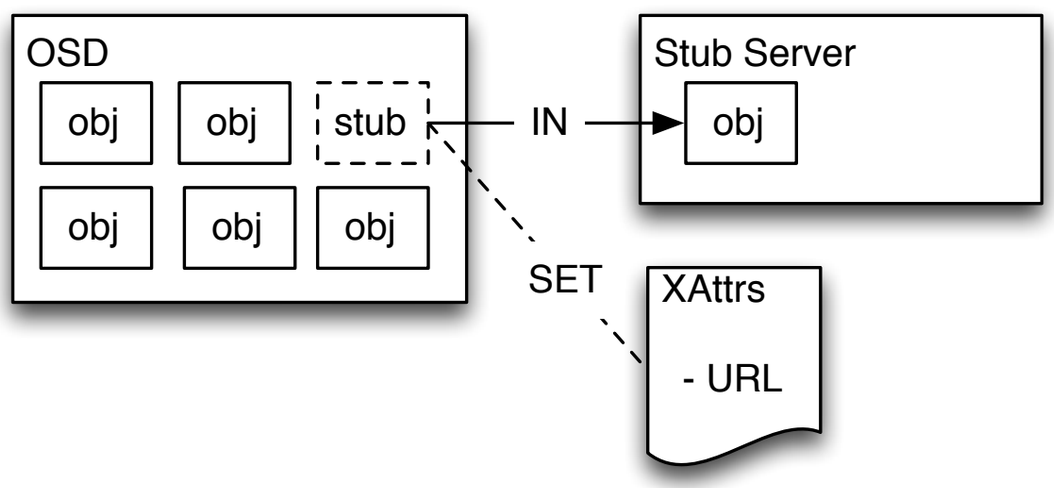
Object Stubs

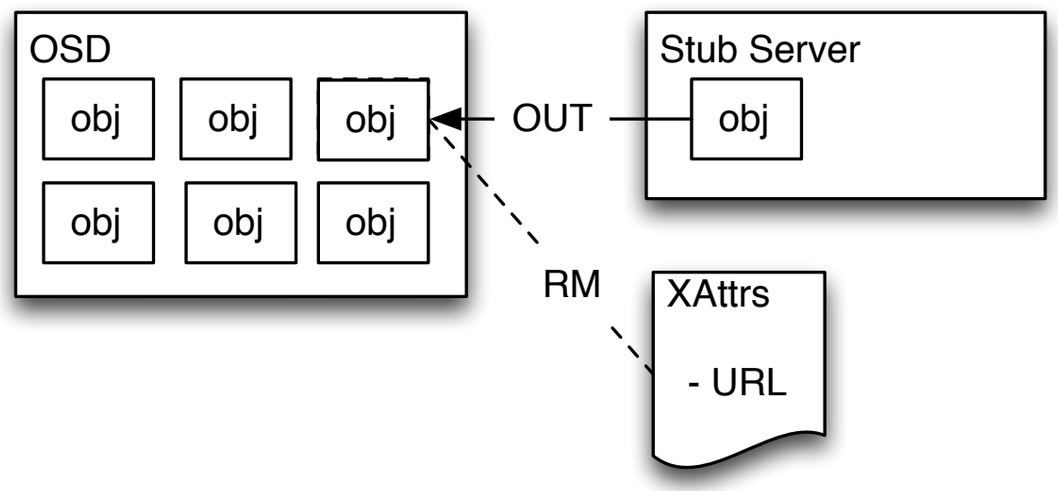


Implementation

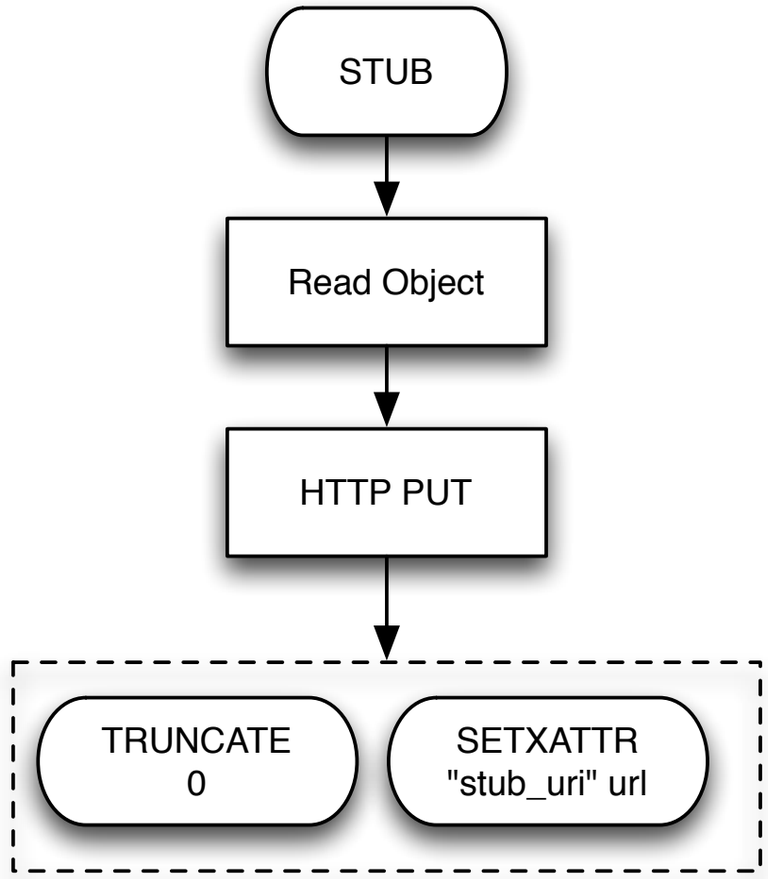
- As part of the OSD
- Transparent to the clients
- New RADOS Operations
  - Stub
  - Unstub
- Implicit unstub

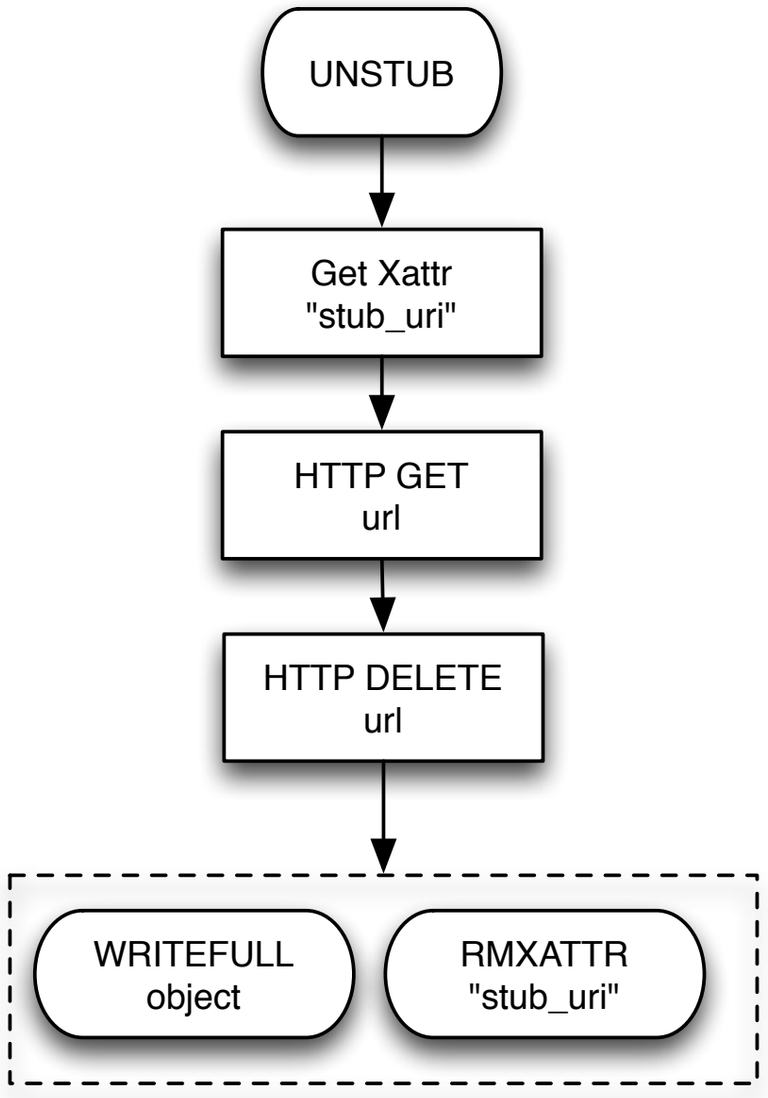
New RADOS Ops: Stub, Unstub





Implementation: Stub, Unstub





Implicit Unstub

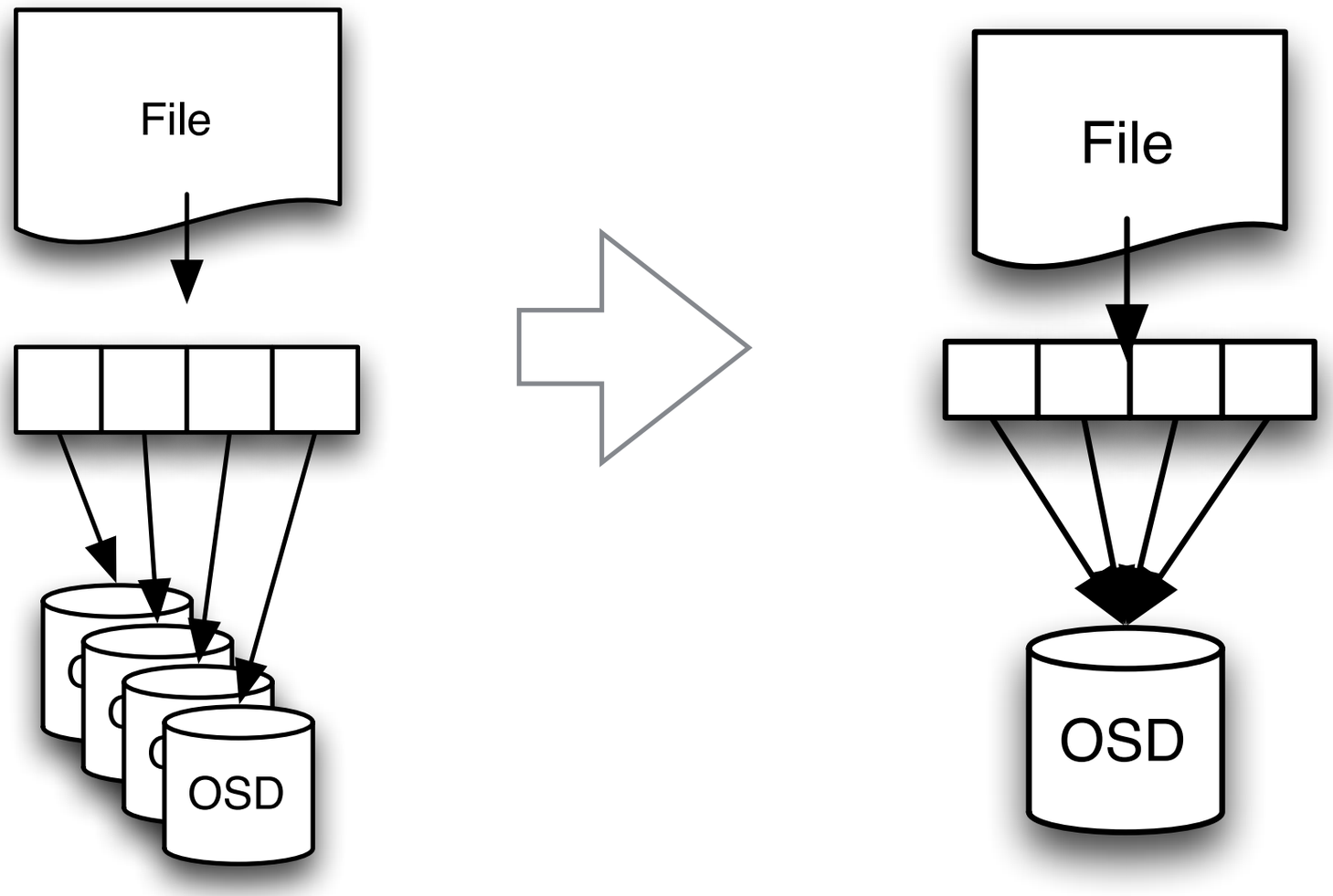
- Scan operation lists for ops that need data

- Prepend Unstub

Benefits

- Supports links to external storage systems
- Stubbed Snapshots => Backup

Striper Prefix Hashing



Implementation

```
uint32_t pg_pool_t::hash_key(const string& key,
                             const string& ns) const
{
  string n = make_hash_str(key, ns);
  return ceph_str_hash(object_hash, n.c_str(), n.length());
}
```

```
uint32_t pg_pool_t::hash_key(const string& inkey,
                             const string& ns) const
{
  string key(inkey);

  if (flags & FLAG_HASHPSONLYPREFIX) {
    string::size_type n = inkey.find(".");

    if (n != string::npos) {
      key = inkey.substr(0, n) ;
    }
  }

  string n = make_hash_str(key, ns);
  return ceph_str_hash(object_hash, n.c_str(), n.length());
}
```

Methodology

- ECMWF ECFS HPSS dump [1]
  - 137 million files
  - 14.8 PiB
  - 10% random sample
- Simulator

[1] M. Grawinkel, L. Nagel, M. Mäsker, F. Padua, A. Brinkmann, and L. Sorth. Analysis of the ECMW Storage Landscape. *Proc. of the 13th USENIX Conference on File and Storage Technologies (FAST)*, 2015

Simulator

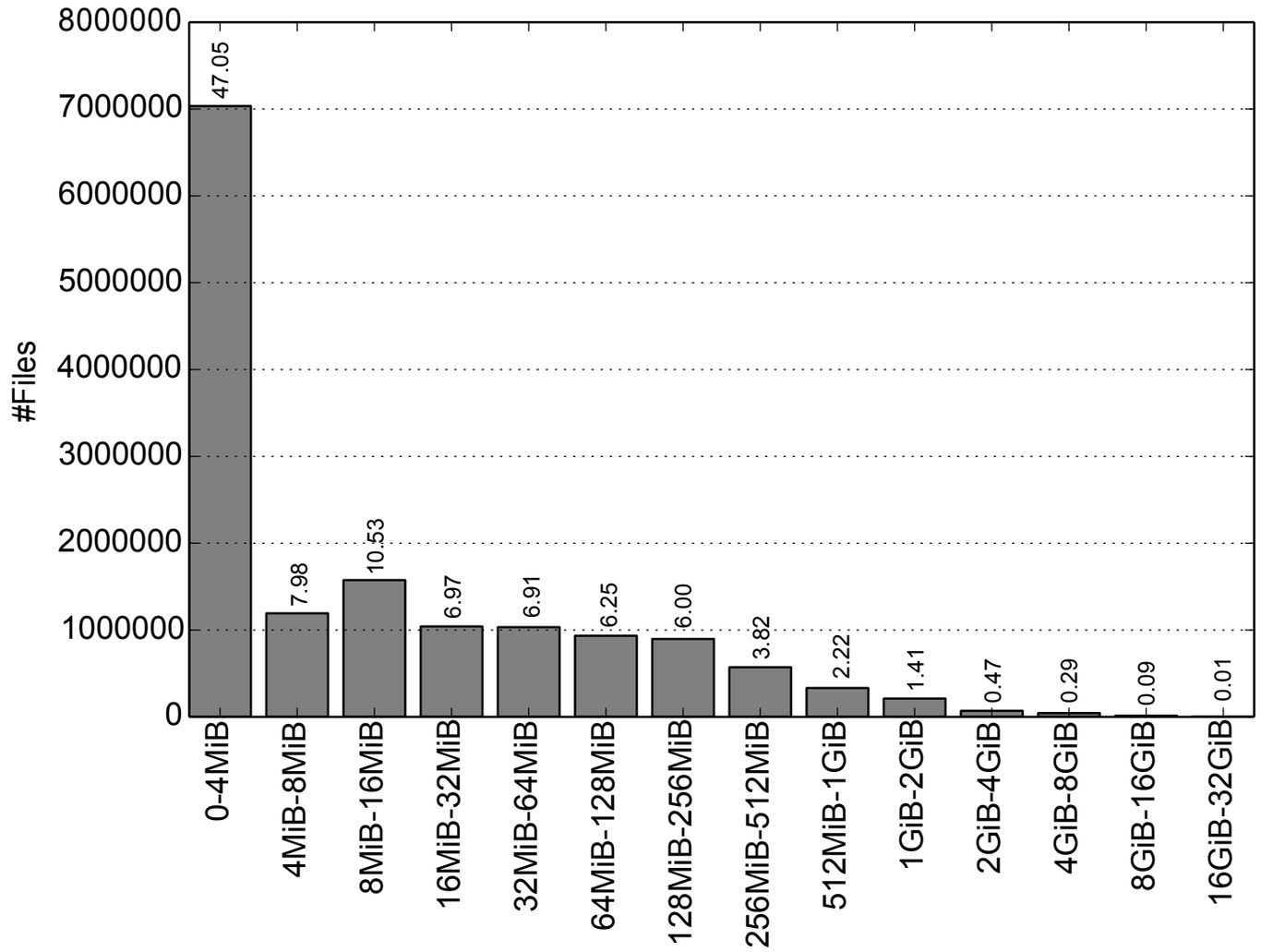
- 600 OSDs
- 38400 PGs
- 45 minutes on 32 cores

	Hash Algorithm	Prefix Hash Enabled?
1	RJenkins	No
2	Linux	No
3	RJenkins	Yes
4	Linux	Yes

Workload

ECFS HPSS 10% random sample

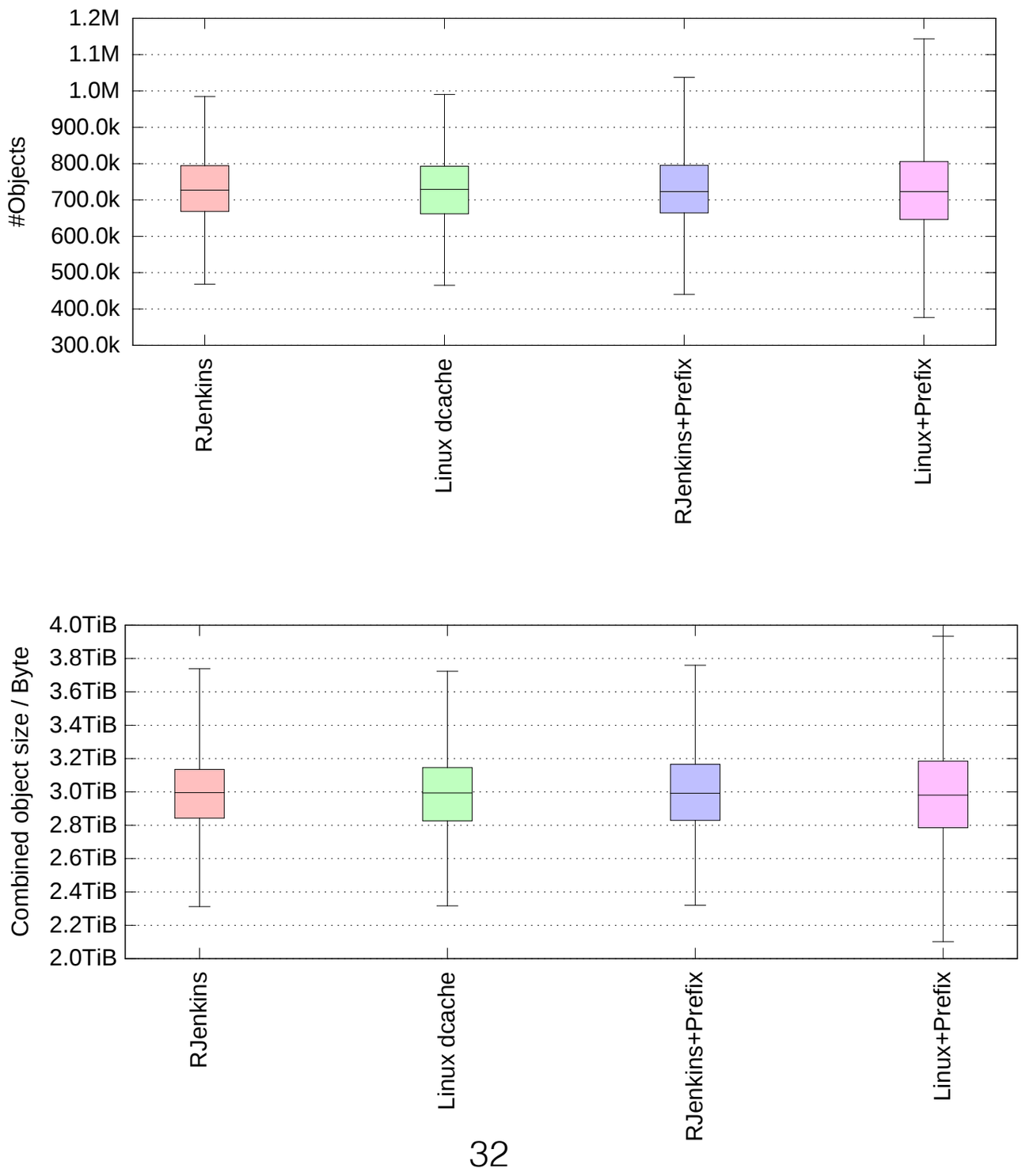
Total #files	~ 15 mil
Total used capacity	1.595 PiB
Max file size	32 GiB
Size of Objects ≤ 4 MiB	5.495 GiB
Size of Objects > 4 MiB	1.590 PiB



Distinct OSDs per File or: Does it work?

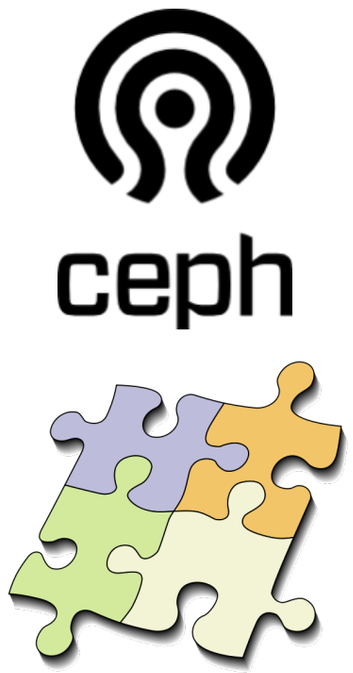
Statistic	RJenkins	Linux dcache	RJenkins+Prefix	Linux+Prefix
Min.	1	1	1	1
Q_1	3	3	1	1
Median	9	9	1	1
Q_3	35	35	1	1
Max.	600	600	1	1

Balance

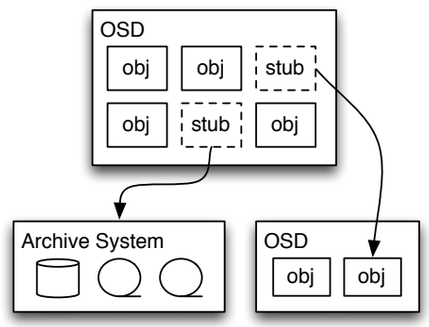

#Objects
1.2M
1.1M
1.0M
900.0k
800.0k
700.0k
600.0k
500.0k
400.0k
300.0k
RJenkins
Linux dcache
RJenkins+Prefix
Linux+Prefix
Combined object size / Byte
4.0TiB
3.8TiB
3.6TiB
3.4TiB
3.2TiB
3.0TiB
2.8TiB
2.6TiB
2.4TiB
2.2TiB
2.0TiB
RJenkins
Linux dcache
RJenkins+Prefix
Linux+Prefix

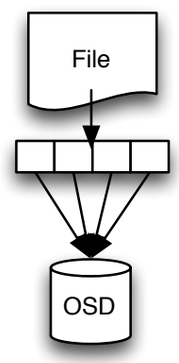
Racap

- Ceph
- Cooling Down Ceph



- Implementation and Evaluation
  - Object Stubs
  - Striper Prefix Hashing



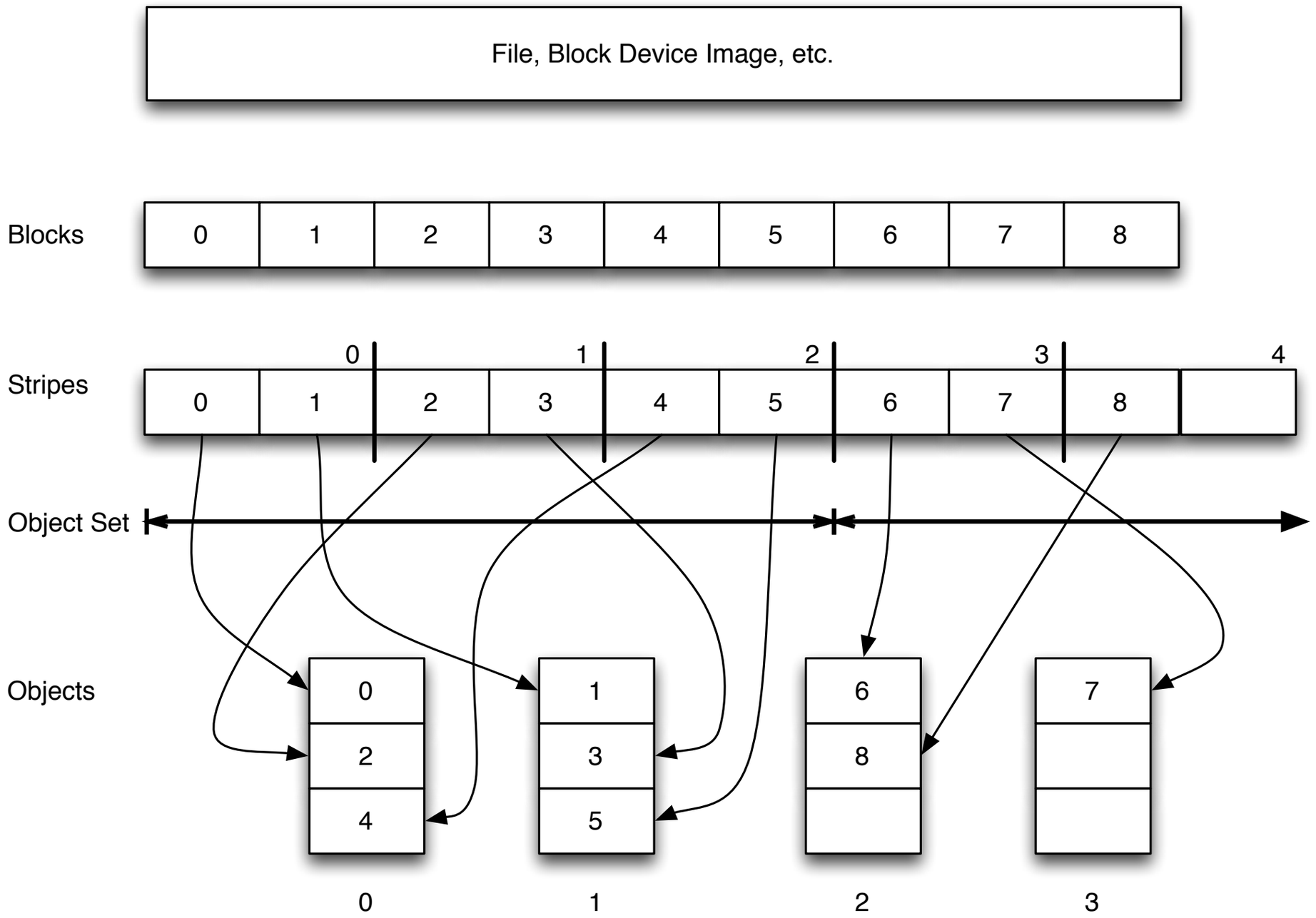


Cooling Down Ceph

Exploration and Evaluation of Cold Storage Techniques

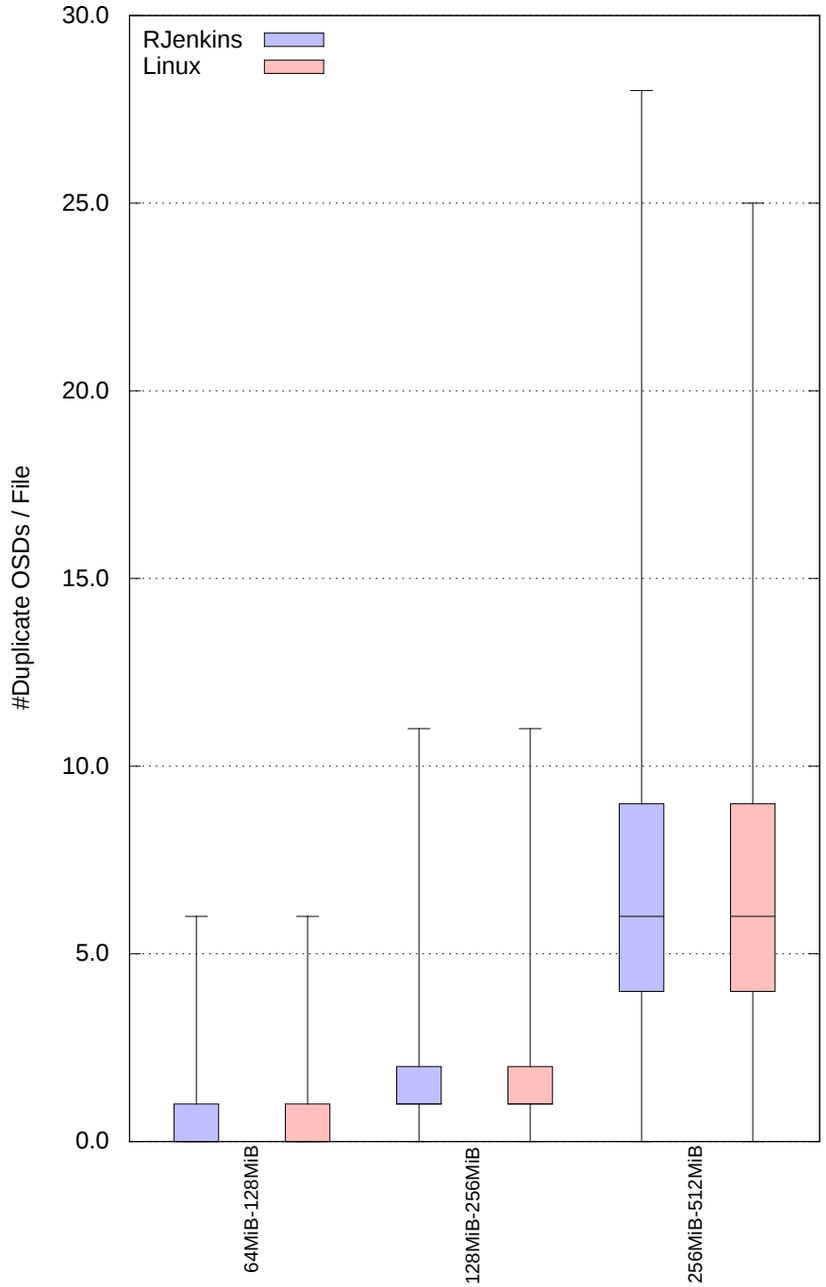
Bonus Slides

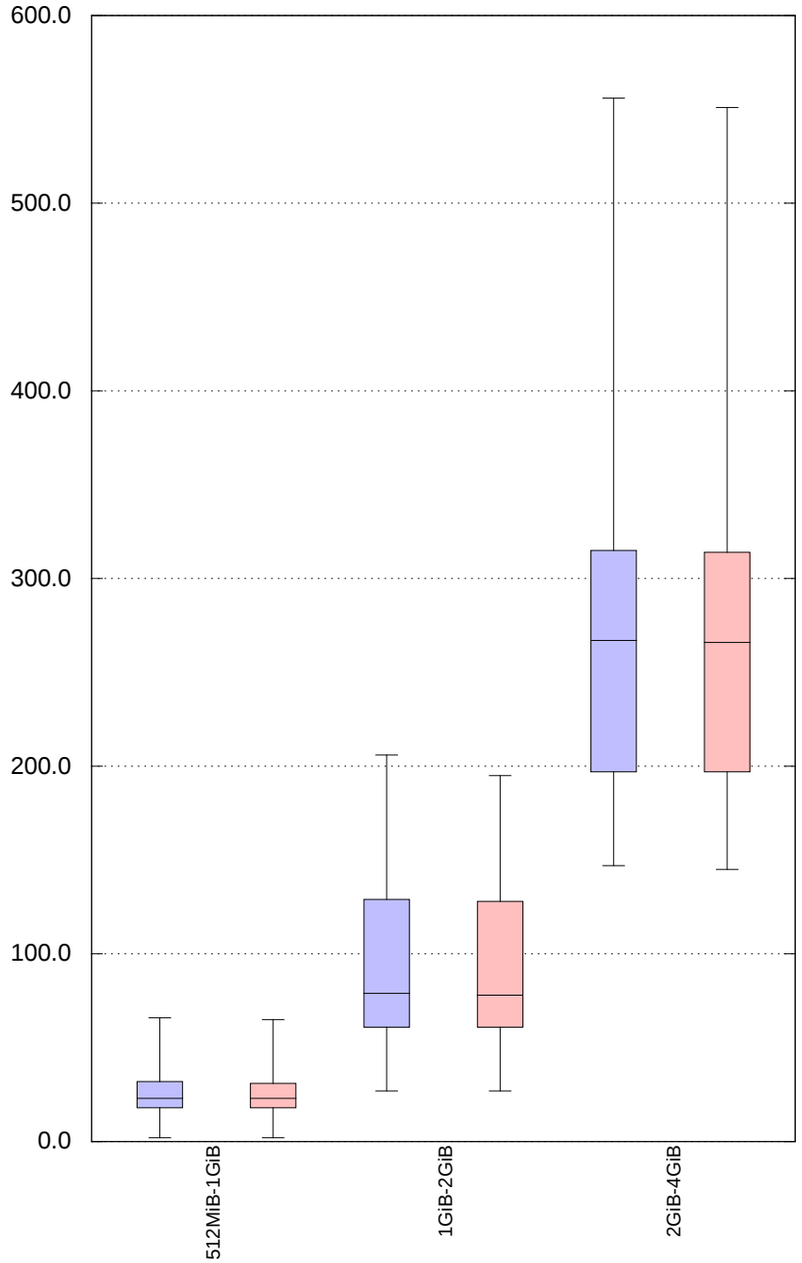
Striper

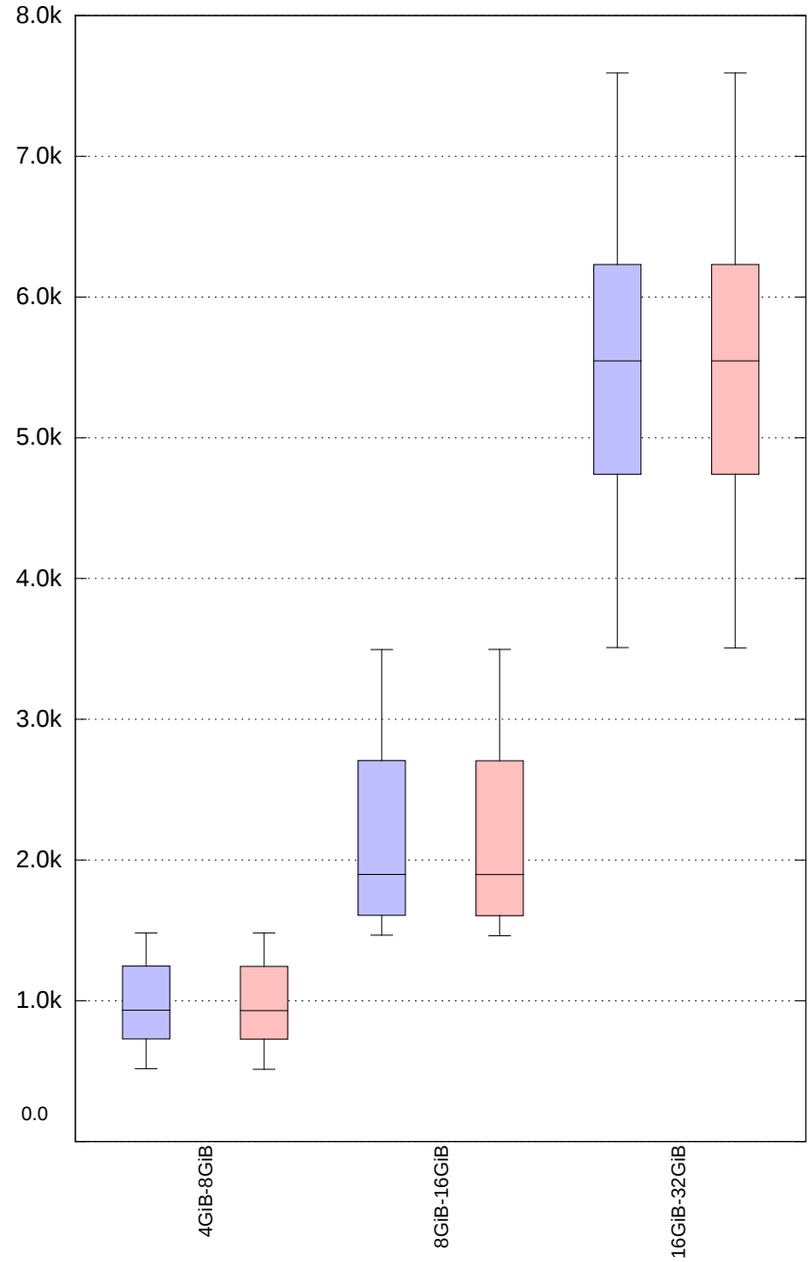

File, Block Device Image, etc.
Blocks
0
1
2
3
4
5
6
7
8
Stripes
0
1
2
3
4
0
1
2
3
4
5
6
7
8
Object Set
Objects
0
2
4
1
3
5
6
8
7
0
1
2
3

Implicit Unstub Ops

read	✓	cache-flush	✓	tmapup	✓	unbalance-reads	✓
stat		cache-evict	✓	tmapput	✓	scrub	✓
mapext	✓	cache-try-flush	✓	tmapget	✓	scrub-reserve	✓
masktrunc	✓	tmap2omap	✓	create	✓	scrub-unreserve	✓
sparse-read	✓	set-alloc-hint		rollback	✓	scrub-stop	✓
notify		redirect		watch		scrub-map	✓
notify-ack		unredirect		omap-get-keys		wrlock	
assert-version		clonerange	✓	omap-get-vals		wrunlock	
list-watchers		assert-src-version		omap-get-header		rdlock	
list-snaps		src-cmpxattr		omap-get-vals-by-keys		rdunlock	
sync_read	✓	getxattr		omap-set-vals		uplock	
write	✓	getxattrs		omap-set-header		dnlock	
writefull	✓	cmpxattr		omap-clear		call	
truncate	✓	setxattr		omap-rm-keys		pgls	
zero	✓	setxattrs		omap-cmp		pgls-filter	
delete		resetxattrs		copy-from	✓	pg-hitset-ls	
append	✓	rmxattr		copy-get-classic	✓	pg-hitset-get	
startsync		pull	✓	undirty		pgnls	
settrunc		push	✓	isdirty		pgnqls-filter	
trimtrunc	✓	balance-reads	✓	copy-get	✓		


RJenkins
Linux
#Duplicate OSDs / File
30.0
25.0
20.0
15.0
10.0
5.0
0.0
64MiB-128MiB
128MiB-256MiB
256MiB-512MiB


600.0
500.0
400.0
300.0
200.0
100.0
0.0
512MiB-1GiB
1GiB-2GiB
2GiB-4GiB


8.0k
7.0k
6.0k
5.0k
4.0k
3.0k
2.0k
1.0k
0.0
4GiB-8GiB
8GiB-16GiB
16GiB-32GiB