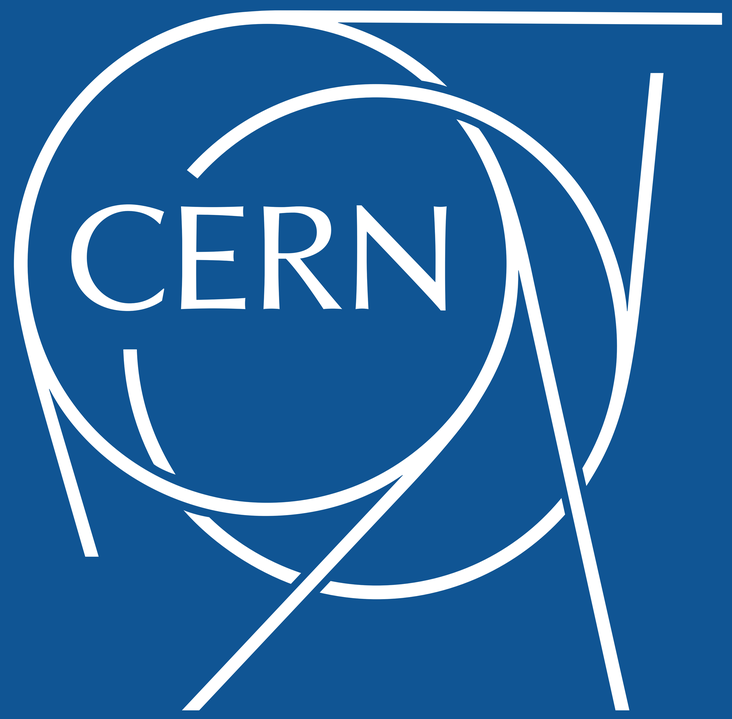

CERN

Ceph-based storage services for Run2 and beyond

Dan VAN DER STER, Massimo LAMANNA, Luca MASCETTI,
Andreas PETERS, Hervé ROUSSEAU (all CERN IT-DSS)

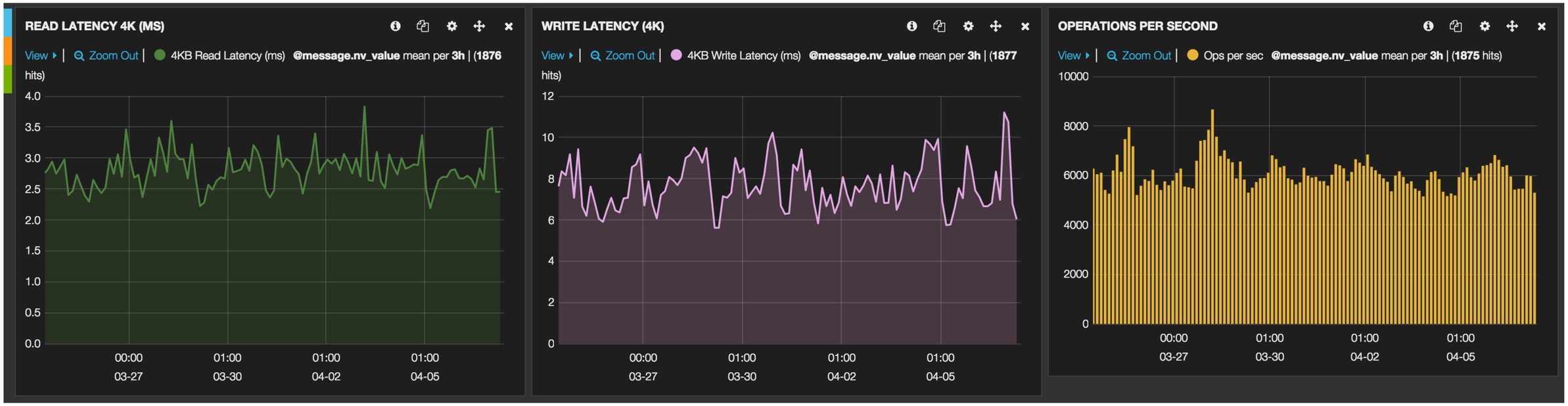
CHEP 2015 – Okinawa – 16 April 2015

Status

- 3 petabyte cluster in production for ~18 months
  - OpenStack Cinder volumes and Glance images
  - 500TB used (including triple replication)
  - ~3000 block devices: 1200 volumes, 1800 images
  - 2 availability zones at CERN, 1 coming in Budapest
- Plenty of useful ops experience
  - Performance tuning in software and with SSDs
  - Host failures, power outages, network incidents
- Ceph for physics data storage?
  - Ongoing development projects
  - Large scale (10's PBs) tests early results

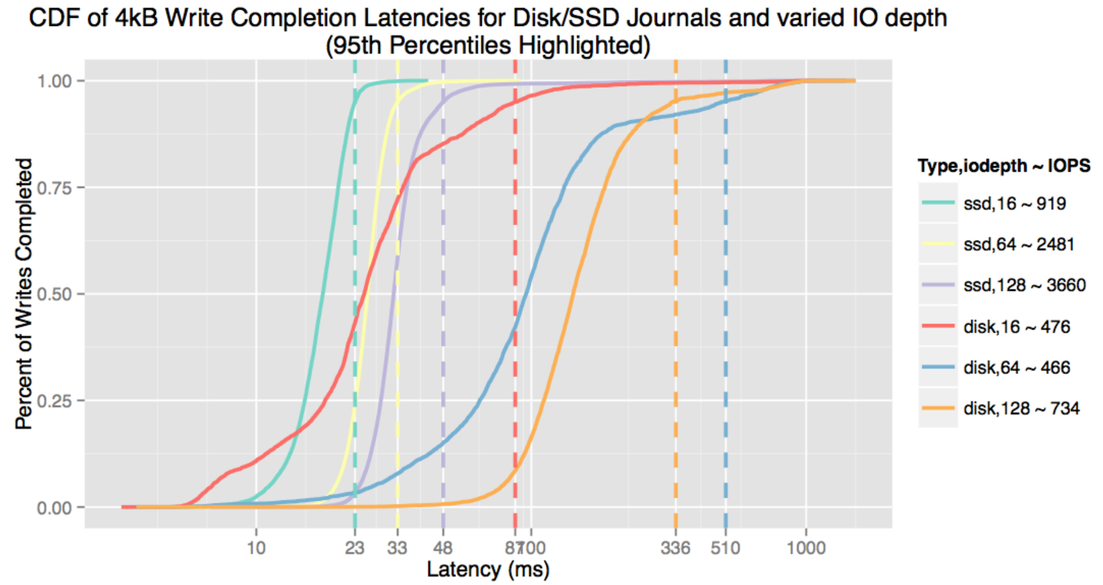
Storage for OpenStack

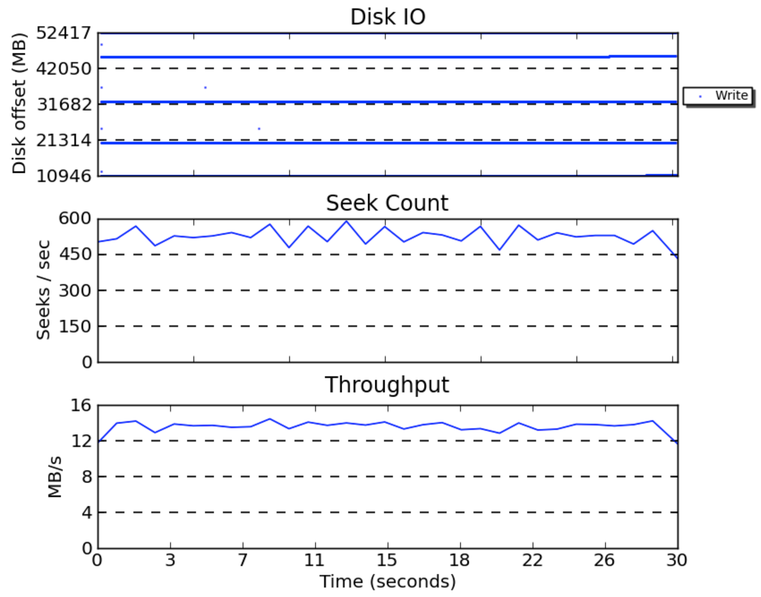
- Basic model: user boots a VM and attaches an arbitrarily large block device
- On-demand storage enables cool applications:
  - Horizontal scalability for central VO services (e.g. ATLAS PanDA, CMS GlideinWMS, CMS Webtools…)
  - Close to 100TB of ZFS-based virtual NFS servers
  - CVMFS squids and (soon) stratum zero
  - Many applications we don't follow closely: ~1000 VMs have an attached Ceph volume
- Tunable performance: IOPS is configurable (per attached volume)
- Cost efficiency with thin provisioning: we have successfully replaced a NetApp and other expensive custom storage servers using this model



IOPS - Do we need SSDs?

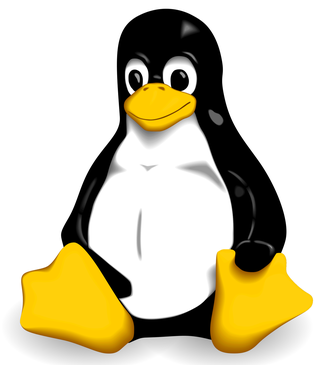
- Ceph uses write-ahead journaling to guarantee write durability
  - Client sees write ack after it is persistent on all replicas
- Two general use-cases:
  - Block: Slow spinning disks will result in unacceptable latency
  - Object: throughput is more important than latency, so SSDs not needed
- 5-10x increase in 4kB write IOPS capacity after installing SSD journals
  - Plus decreased small write latency from ~50ms to 5-10ms (see upper plot)
- Which SSDs to buy?
  - You need high endurance and stable random write performance (see lower plot)
  - We use the 200GB Intel DC S3700 (with 5 journals per SSD)





Ceph (and Linux) Tuning

- The ceph-mon's LevelDB can become very busy during backfilling
  - SSDs are absolutely required on your ceph-mon!!!
- Scrubbing can be very IO intensive
  - Scrubs triggered periodically (default=weekly) – often all at the same time every week (similar to a thundering herd…)
  - You should manually smear the scrubs across the week
- Scrub IOs compete with user IOs:
  - Use cfq on the data disks and ionice (idle) the OSD scrub thread
- Handling many small files
  - Hosting 100's of thousands of files puts immense cache pressure on inode/dentry cache (check vm.vfs_cache_pressure)
  - NUMA zone reclaim is a disaster (on RHEL 6). To disable it: vm.zone_reclaim_mode = 0
- Ceph uses many many threads (and sockets)
  - Servers need kernel.pid_max = 4194303
  - Clients need ulimit -n > 4096
- Disable updatedb for /var/lib/ceph

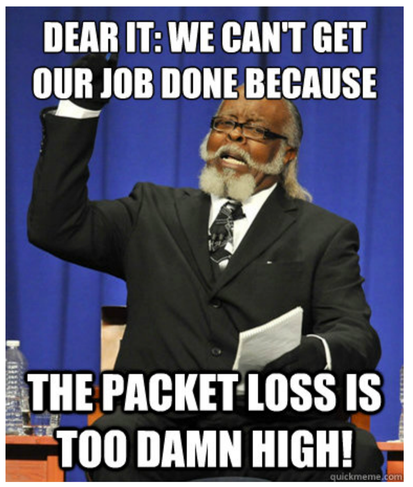


Interesting Incidents

- Disk & host failures
  - Disks fail ~monthly in the cluster: completely transparent to our clients
  - SSD journals have been very reliable: no failures yet →→→→→→→→→
  - Twice lost a whole host: in both cases the host could be recovered by a human. We now disabled automatic backfilling in this case:

    mon osd down out subtree limit = host

- Power cut 16 October 2015
  - 3 out of 5 Ceph monitors lost power => no quorum
  - Cluster down for ~18 minutes as a result
  - VM IOs were blocked during this time => no corruptions reported
    - *virtio-blk* has this feature. *virtio-scsi* would have timed out IOs



- Router card failure 13 March 2015
  - Random packet loss led to OSDs going up/down over many hours
  - A few OSDs (~20) reached a heartbeat timeout and committed suicide
  - Random network badness, lots of backfilling: intermittent volume access
  - No data corruption reported & confirmed. No data scrub inconsistencies.

30 petabyte test

- We borrowed 150 servers – 192TB each – for a short lived Ceph scale test
  - General summary: it works! and performance was pretty good.
- … what's the catch?

- We found a scalability limitation:
  - An osdmap is a blob which contains the structure of the cluster, the directory of OSDs, the CRUSH map, etc…
  - With 7200 OSDs the osdmap was ~4MB in size
  - Each OSD caches up to 500 previous versions of the osdmap – even with some deduplication each OSD consumed 3-4GB RAM. 48 OSDs per server ➔ ~200GB RAM needed just to cache the osdmaps!!!! (And we feel this is wasted RAM, since old osdmaps are mostly not useful)
  - We found a workaround (cache fewer maps) but there must be a better solution!

- See backup slides for performance details, if time permits.

Use-cases beyond OpenStack

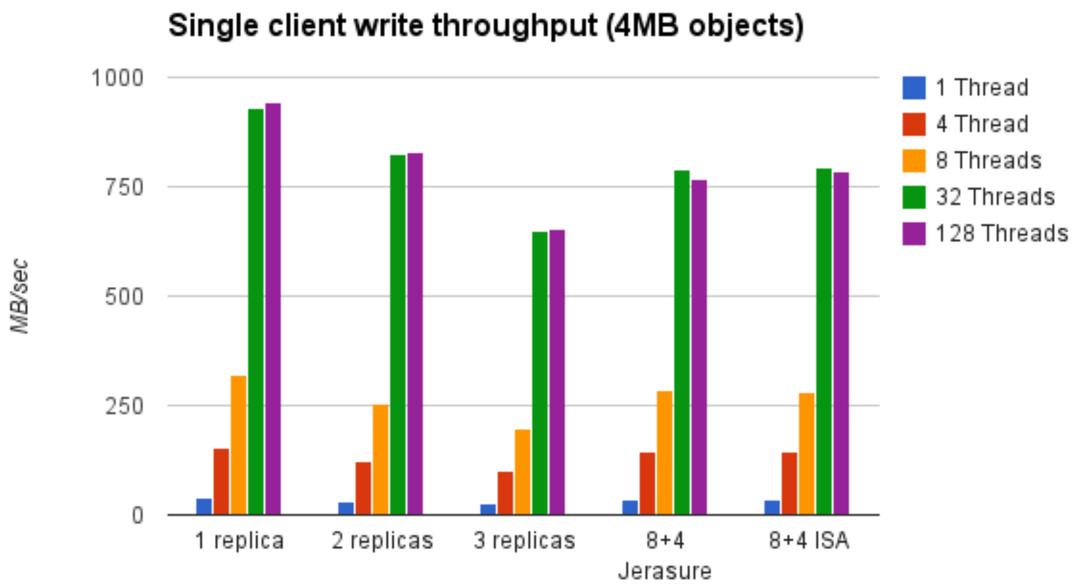
- RADOSGW running in pre-prod for S3:
  - Horizontally scalable with civetweb and haproxy
  - Use-cases: CVMFS stratum zero, BOINC work-units uploads with pre-signed URLs
  - 35 million objects

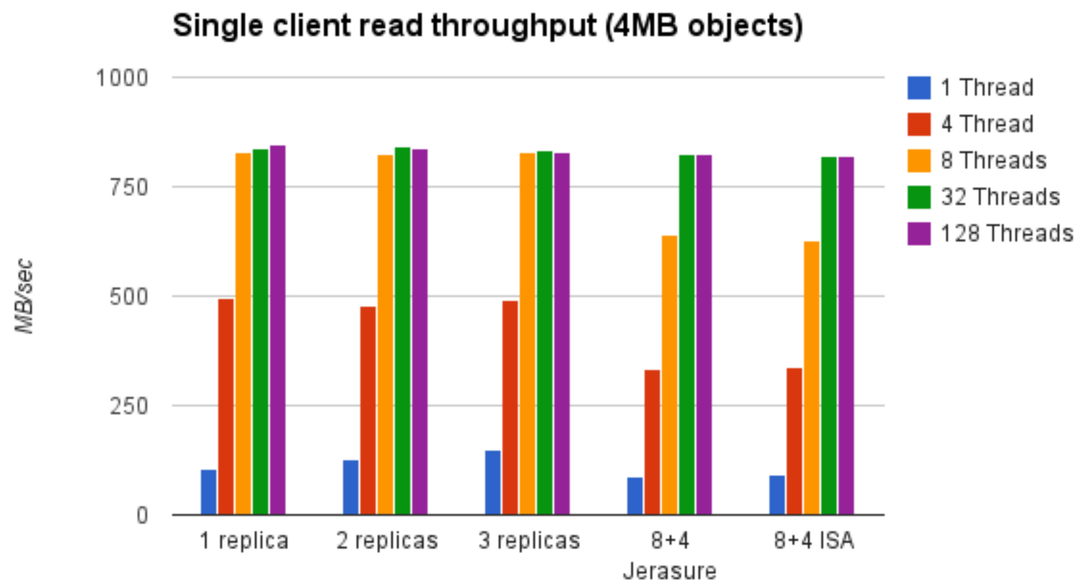
- CASTOR tape buffer
  - V2.1.15 adds RADOS support for striping object storage

- EOS DIAMOND R&D, see:
  - https://indico.cern.ch/event/304944/session/3/contribution/297

www.cern.ch

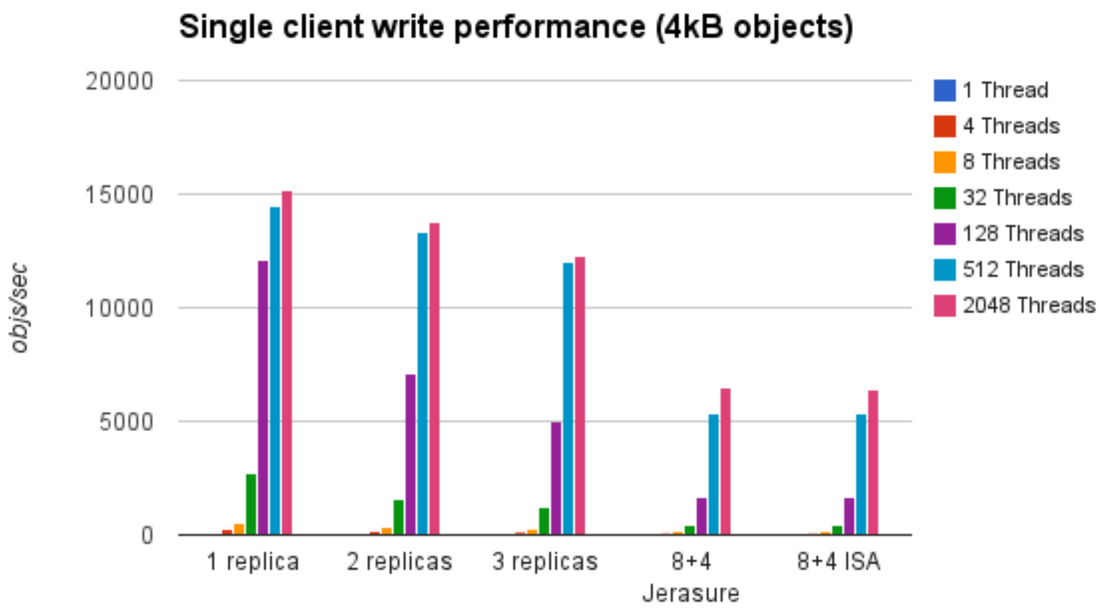
30PB test: single 10GbE client

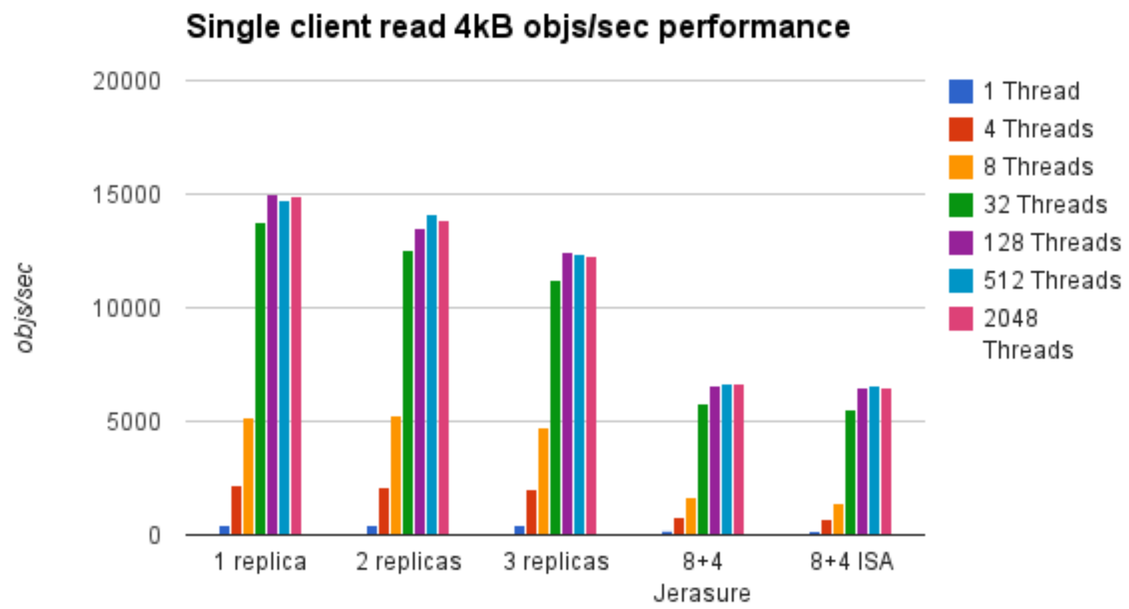




- Saturates at ~900MBps with 32 threads for writing, 8 threads for reading
- Little difference between jerasure and ISA coding

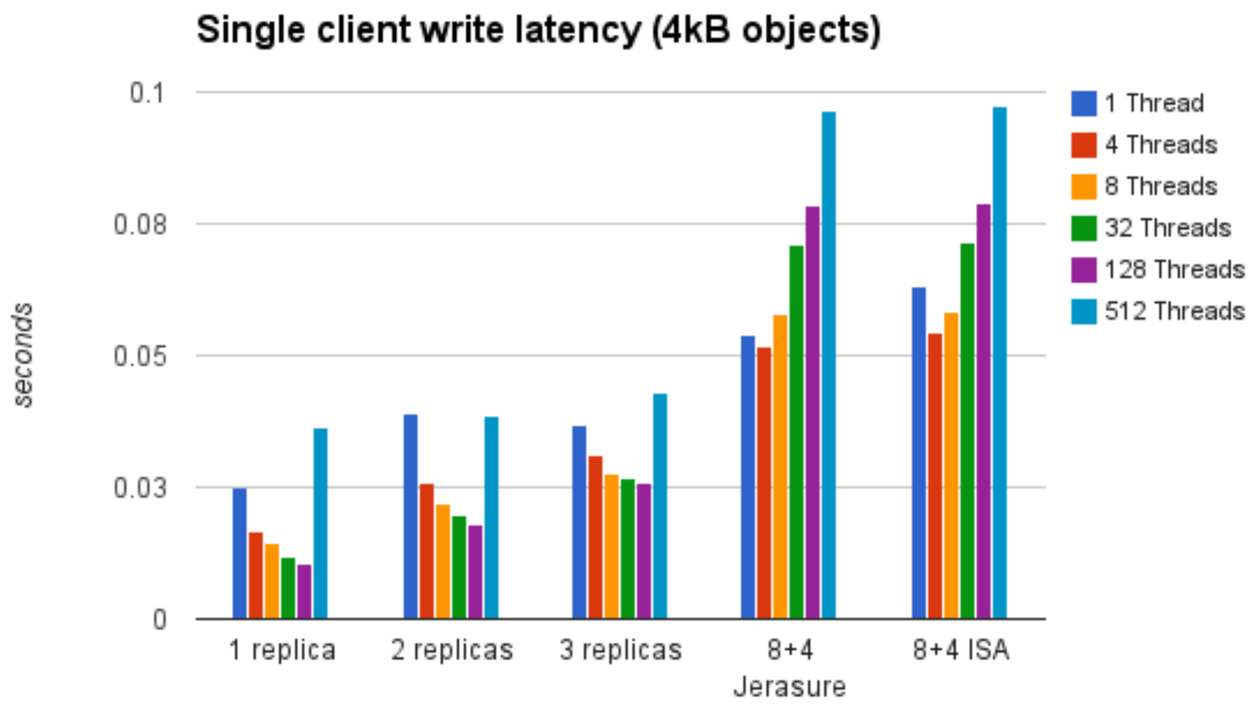
30PB test: single client IOPS





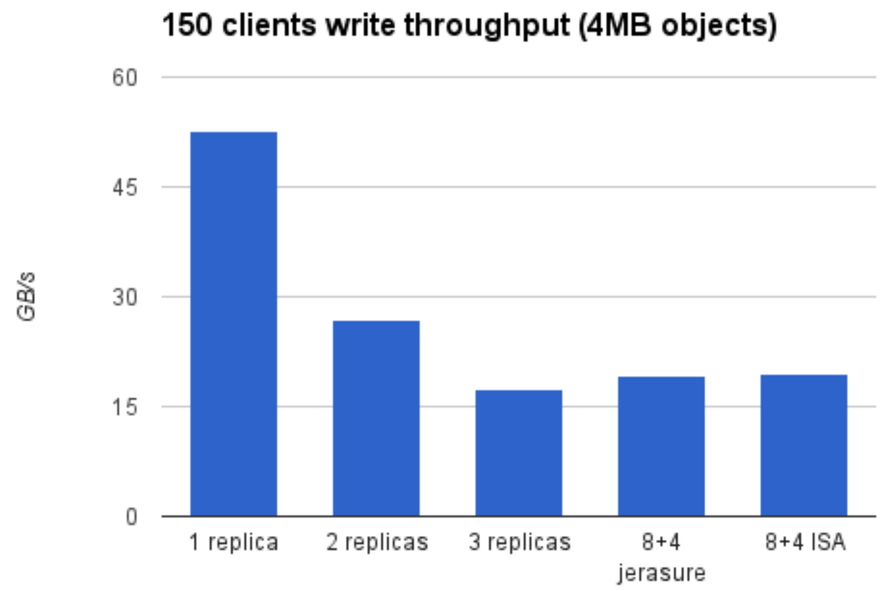
- One client can write/read at ~15000 IOPS to single replica pool. More replicas ➔ fewer IOPS
- Erasure coding leads to roughly halved IOPS

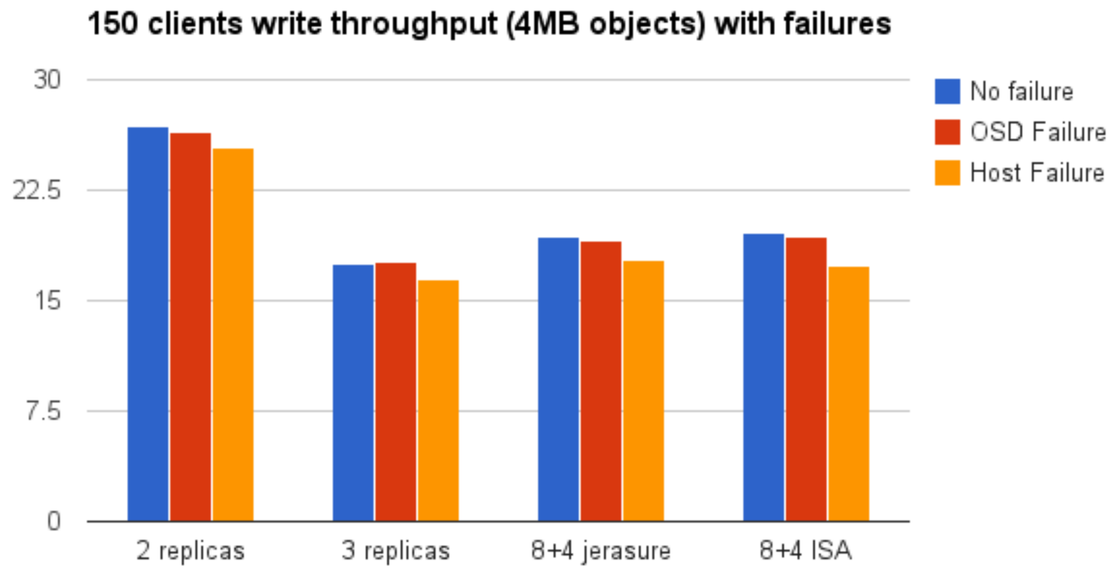
30PB test: single client latency



- 4kB write latency scales from ~12ms for replicated pools to 50-100ms for erasure pools
- (Remember: no SSD journals used for this test)

30PB test: 150 client perf.





- Up to 52GBytes/sec writing to a 1 replica pool. Halved with 2 replicas, thirded with 3 replicas.
  - ISA still roughly equal to jerasure (didn't measure CPU consumption)
- Throughput during failures and backfilling: little impact for all profiles