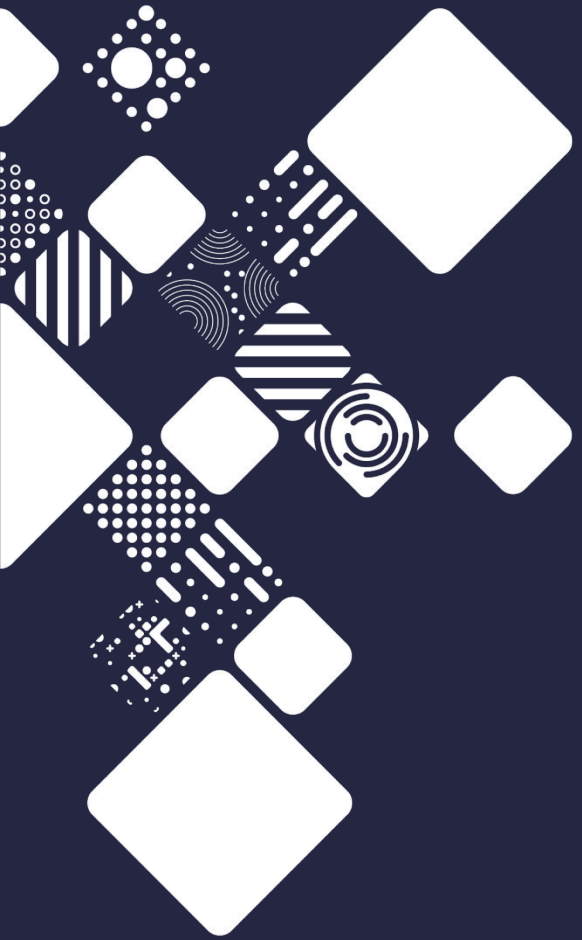


Oops, we ran out of IOPS! Adding NVMe devices to an overworked S3 service.

Dave Holland
Wellcome Sanger Institute

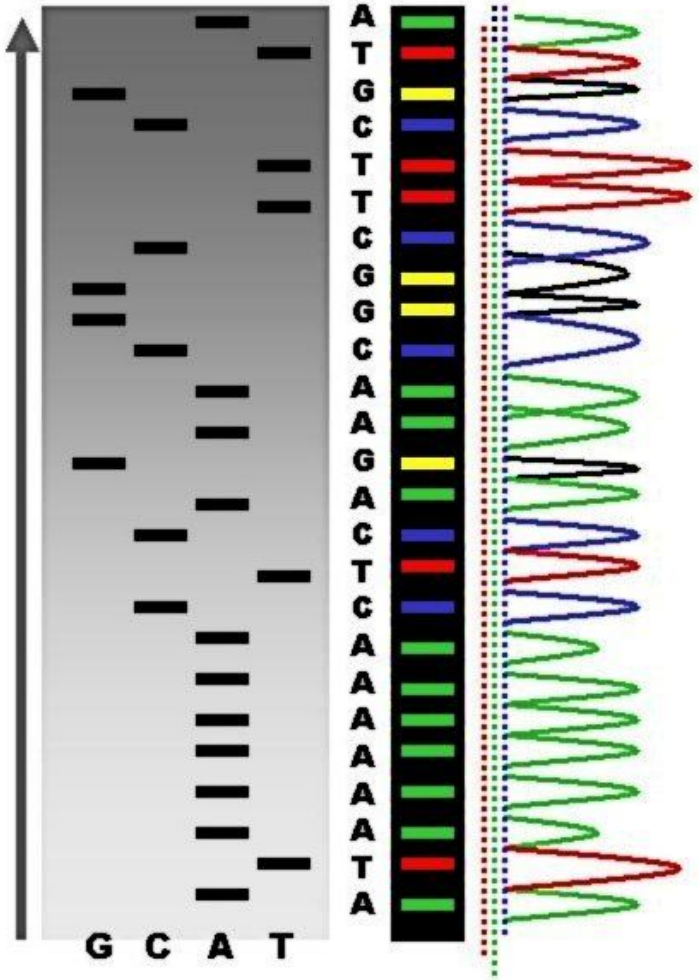
wellcome sanger institute

What I'll talk about

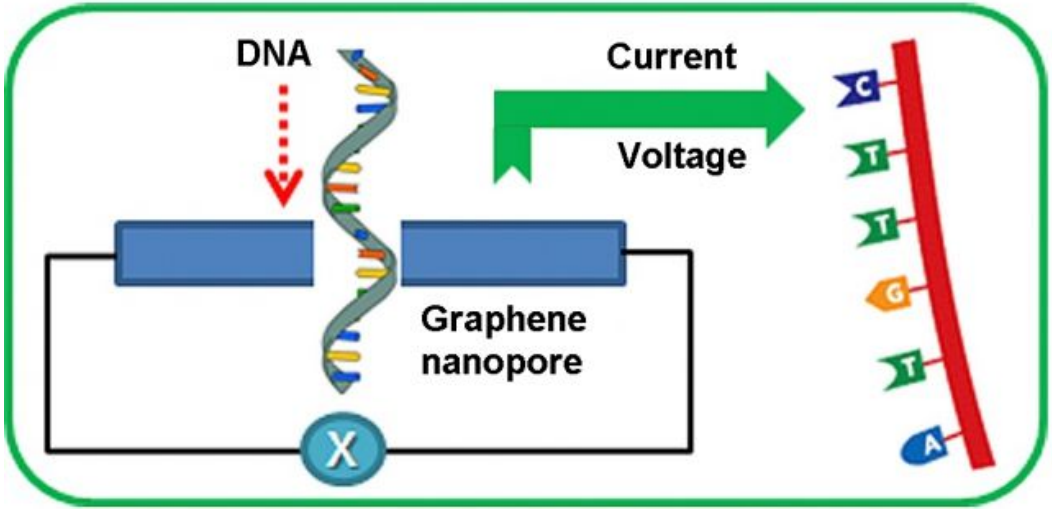
- What the Sanger Institute does
- Where Ceph fits in
- What went wrong
- How we fixed it

wellcome sanger institute

wellcome sanger institute



DNA sequencing (the 30-second summary)



wellcome sanger institute

Then...

ABI 3700

~2Mbase per day

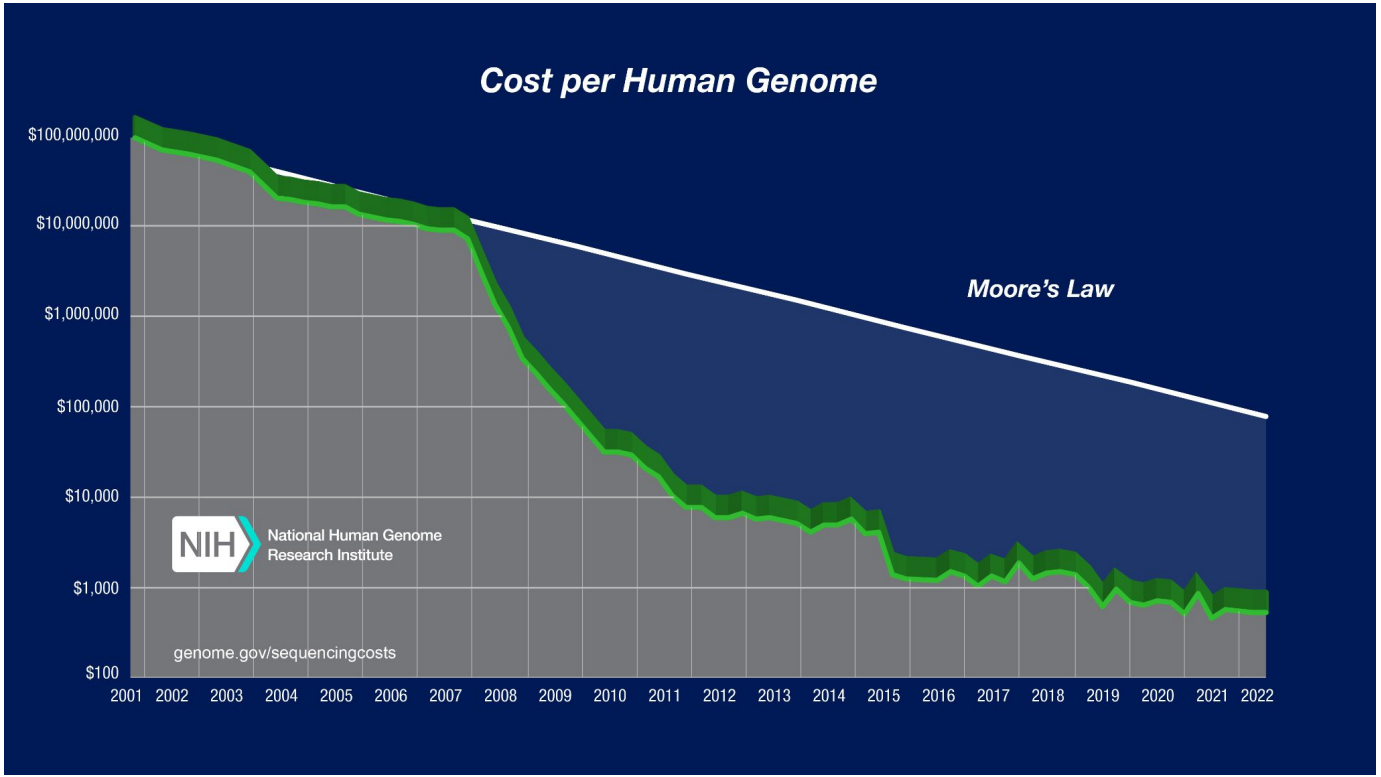
wellcome sanger institute

...and now

Illumina NovaSeq 6000

~3Tbase per day

wellcome sanger institute



wellcome sanger institute

Oxford Nanopore
MinION and SmidgION
up to 20Gbase / 48 hours

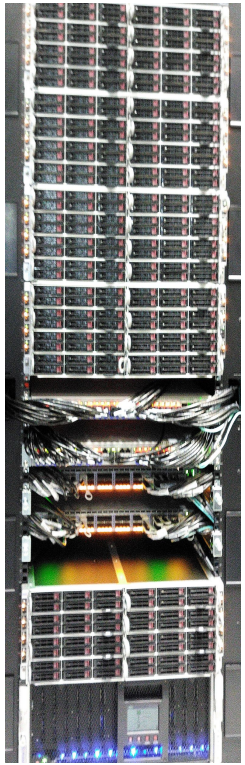
wellcome sanger institute

wellcome sanger institute

Our Ceph journey

wellcome sanger institute

Initial deployment



- 2017: first production system, 1PB usable (3-way replication)
  - 9 Supermicro systems, each with 60x 6TB HDD + 2x 2TB NVMe for journal, 2x 100GbE
- Arista 7060CX switches, spine/leaf

wellcome sanger institute

Build it and they will come

- The system was designed to support OpenStack (RBD for images and volumes) with “a little” S3 use
- It quickly became obvious that S3 was the killer feature from the users’ perspective
  - Dataset hosting
  - “Web site” hosting
  - Collaborator data transfer (including HM Government) 🏛️
- Now >75% of data stored in Ceph is radosgw/S3

wellcome sanger institute

Ceph growth

- 2018 expansion: 1PB → 4.5PB usable
  - 9 servers → 51 servers (adding a 4th failure domain)
  - 540 OSDs → 3060 OSDs
- 2019: added 6 dedicated radosgw servers with a HAproxy layer for resilience
- Upgrades: version 10 (Jewel) to 12 (Luminous) to 14 (Nautilus) to 16 (Pacific)
- Ceph is generally hugely robust! 💪

wellcome sanger institute

wellcome sanger institute

What went wrong

Hitting the edges

- In early 2022 we experienced occasional but repeated waves of OSD crashes and suicides
- systemd initially papered over the issue by restarting OSDs
- Two 6+ hour outages focused our attention 😓
- With help from Canonical support, backtraces initially pointed to rgw GC corruption or bug
  - Tried recreating default.rgw.gc pool
- A workaround was to disable GC
  - But that means the GC list can grow without bound

The cause

- Deletion of a large number of objects via S3 or aborting multipart uploads involves a garbage collection workload that can overwhelm a HDD OSD
  - The metadata (GC/index objects) has a much higher read/modify/write workload than the data
- The OSD is marked as down then out by its peers as it is too busy to respond to heartbeats
- This triggers data migration to recover loss of redundancy
- The extra I/O on other OSDs causes an avalanche of OSD failures 💥

The plan

Move the rgw metadata to “something faster” 🏃

- Purchase 8x 2TB NVMe devices to be installed in selected nodes (2 per failure domain) - ample for <100GB metadata
- Define NVMe device class and CRUSH rules
- Configure the rgw metadata pools to use NVMe rules instead of the HDD rules
- …
- Profit!

wellcome sanger institute

Nuts and bolts

```
ceph osd crush class create nvme

ceph osd crush set-device-class nvme 123

ceph osd crush rule create-replicated replicated_nvme \
    default rack nvme

ceph osd pool set default.rgw.meta crush_rule replicated_nvme
```

wellcome sanger institute

Testing, testing

- Canonical support verified the plan
- Used a smaller test cluster for familiarity and reassurance
  - It's HDD-only so made a "fakenvme" device class on a few HDDs
- Adjusted osd-max-backfills and osd-recovery-max-active to control the mayhem
- The process was expected to be transparent to users/clients

wellcome sanger institute

Wrinkles

- Naively adding the NVMe devices would cause the default replication rule to place other data on them
  - Define HDD device class and CRUSH rules, and configure all pools to use HDD rules before adding NVMe devices
- Changing a CRUSH rule from “use any OSD” to “use any HDD OSD” unexpectedly caused data movement in testing
  - Only 0.15% of objects moved, but our production cluster has ~1 billion objects!
  - This is to do with the shadow CRUSH tree root ID changing
  - Mitigated with norebalance and CERN’s upmap-remapped.py (could also use crushtool --reclassify)

wellcome sanger institute

Glorious victory

- The plan worked in production just as it worked in testing
- No unexpected data movement
- No interruption to S3 service 🎉
- GC was re-enabled without any ill effects
- Ceph went on to be reliable and robust

Conclusion

wellcome sanger institute

Take-homes

- System usage creep is a thing
- Monitor your I/O rates
- Don’t be too quick to look for a bug
- Having a test environment is invaluable
- Changing the CRUSH map is only a bit scary