Ceph at the Flatiron Institute

Andras Pataki

September 17, 2019

Flatiron Institute

- www.flatironinstitute.org
- Internal research division of the Simons Foundation
- Mission: to advance scientific research through computational methods, including data analysis, modeling and simulation
- Organized into centers
  - Center for Computational Astrophysics (CCA)
  - Center for Computational Biology (CCB)
  - Center for Computational Quantum Chemistry (CCQ)
  - Center for Computational Mathematics (CCM)
  - Scientific Computing Core (SCC)
- Young institution - extremely fast growth

Flatiron Institute

Data of a variety of sorts ...

- Large genomics datasets from sequencers
- Astrophysics simulation outputs

Variety of computational styles ...

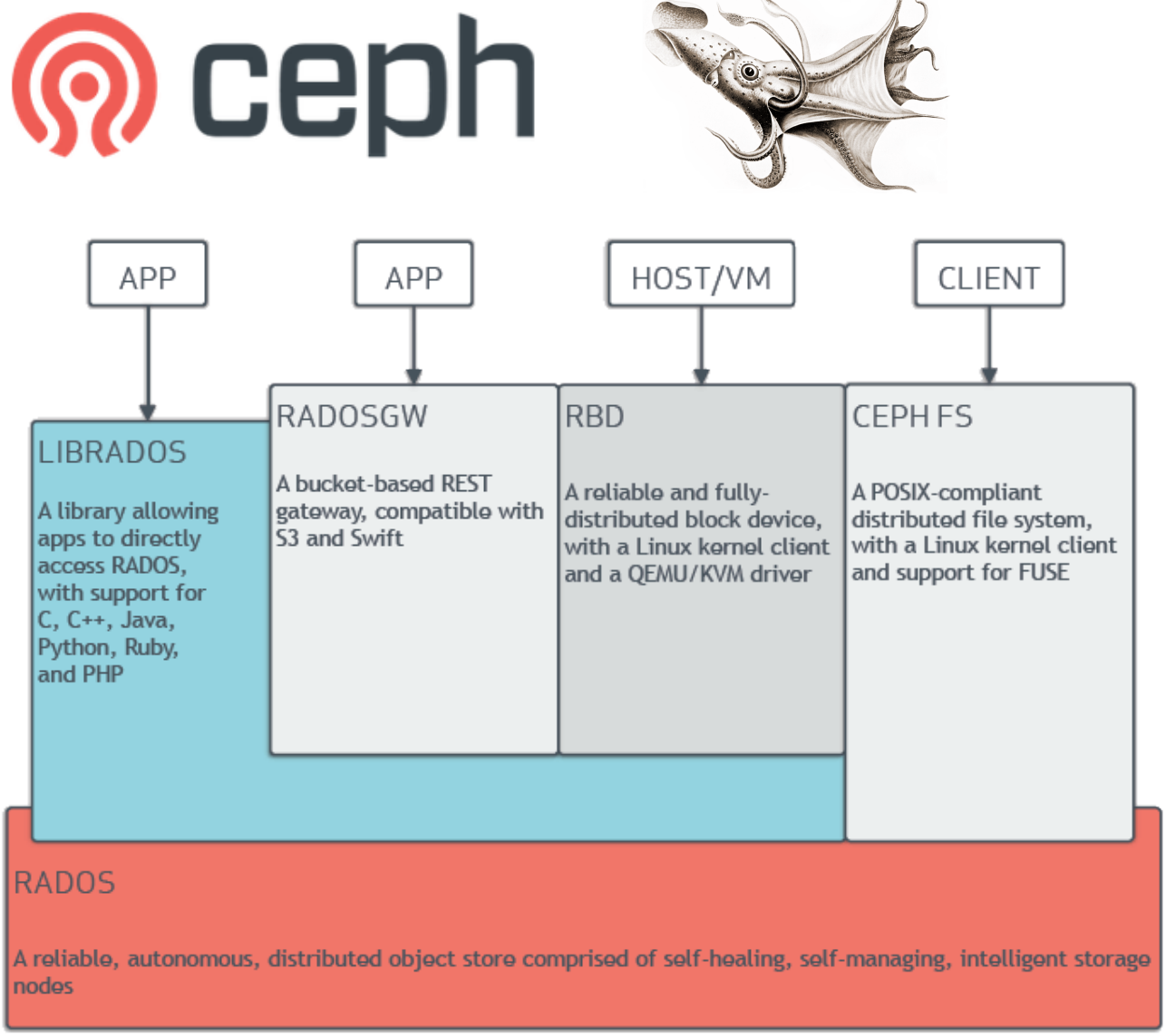
- Embarrassingly parallel: genomics pipelines
- Loosely coupled MPI: protein folding, some quantum chemistry codes
- Tightly coupled MPI: astro sims

Flatiron Institute

Computational resources

- About 40k cores of computing
  - 20k in New York (Manhattan and Brookhaven National Labs)
  - 20k at the San Diego Supercomputer Center (SDSC)
- $\approx$ 200 GPUs
- Almost 30PB of raw space in Ceph storage (Manhattan)
- Also GPFS and Lustre storage

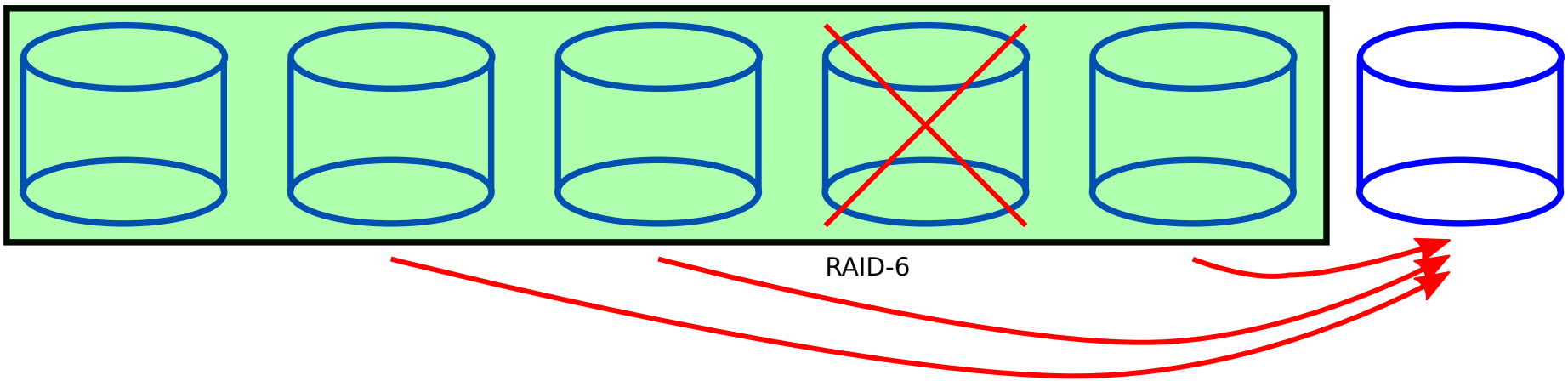
Ceph


ceph
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

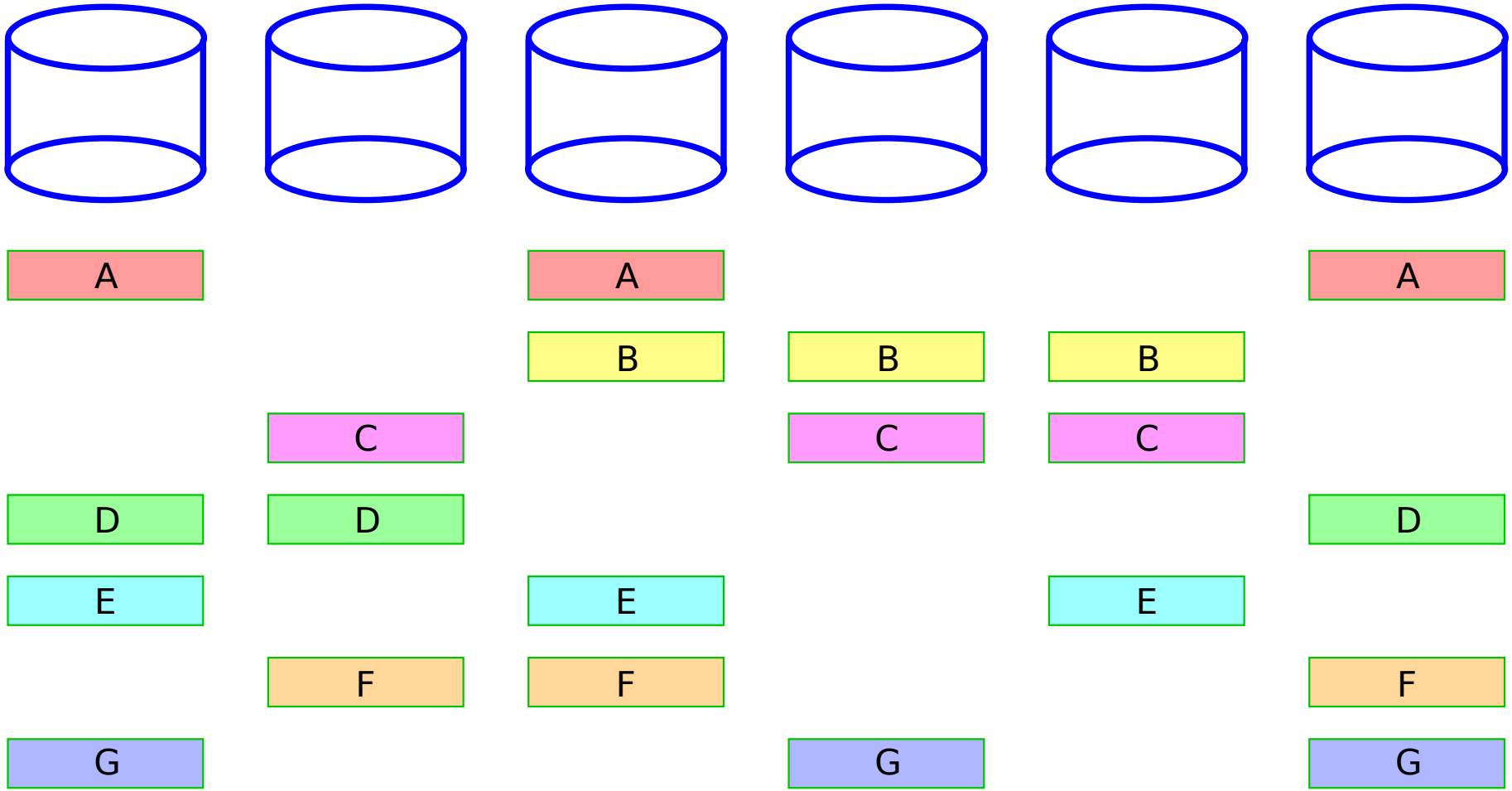
Recovery - Ceph vs. traditional RAID

RAID recovery

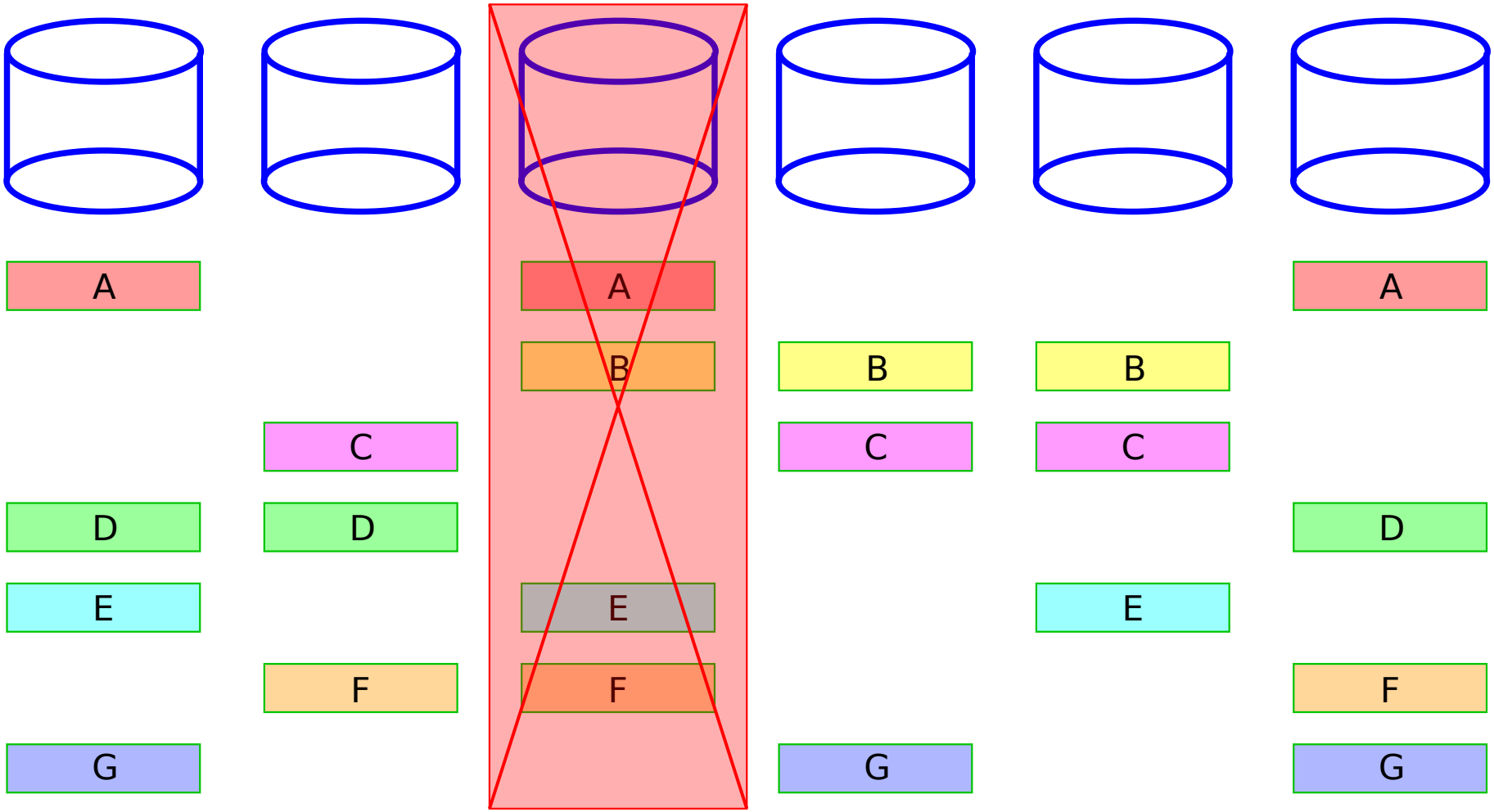
- Drive failure - rebuild requires reading other drives in full
- Secondary drive failures more likely



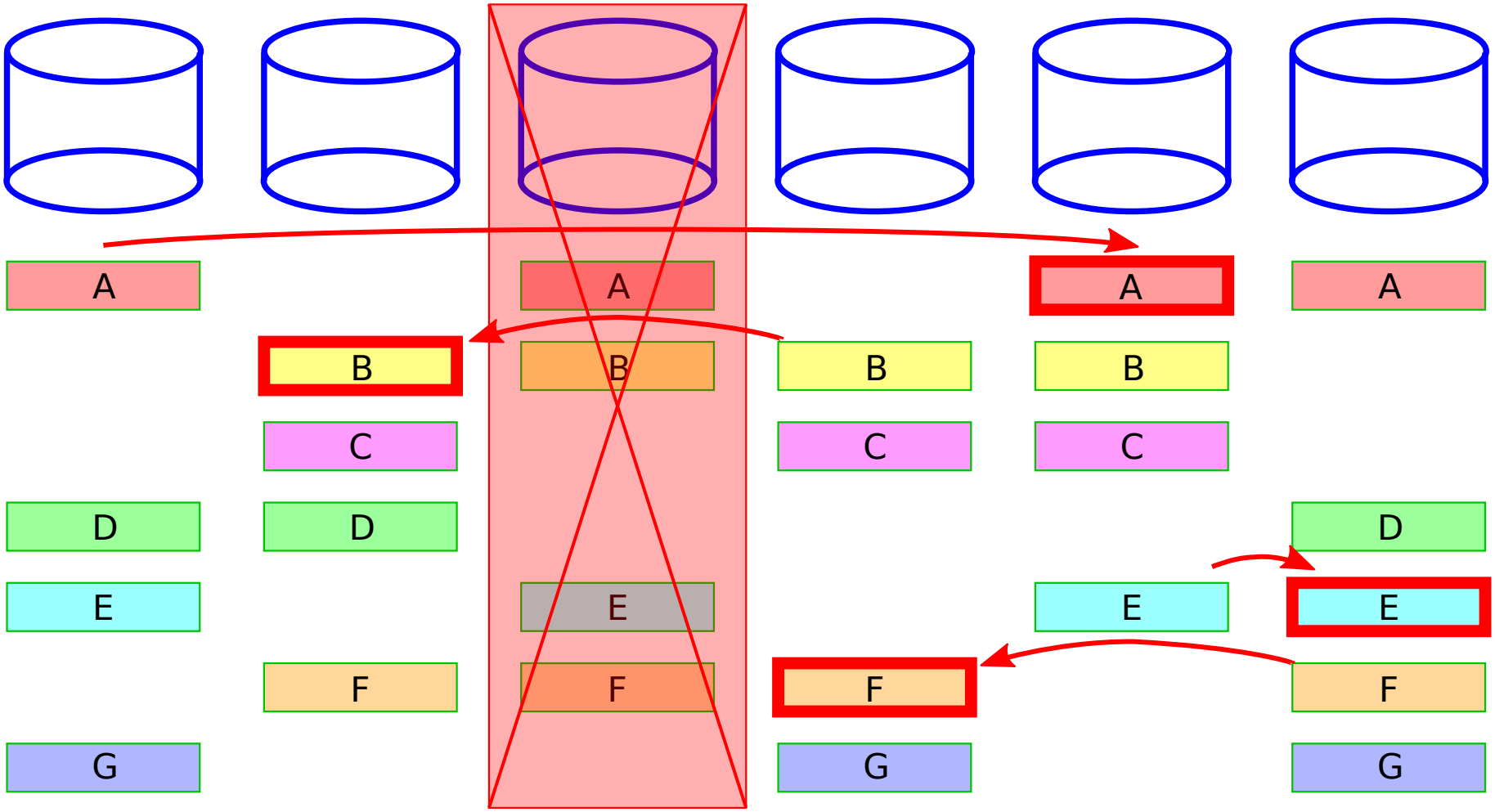
Declustered Placement



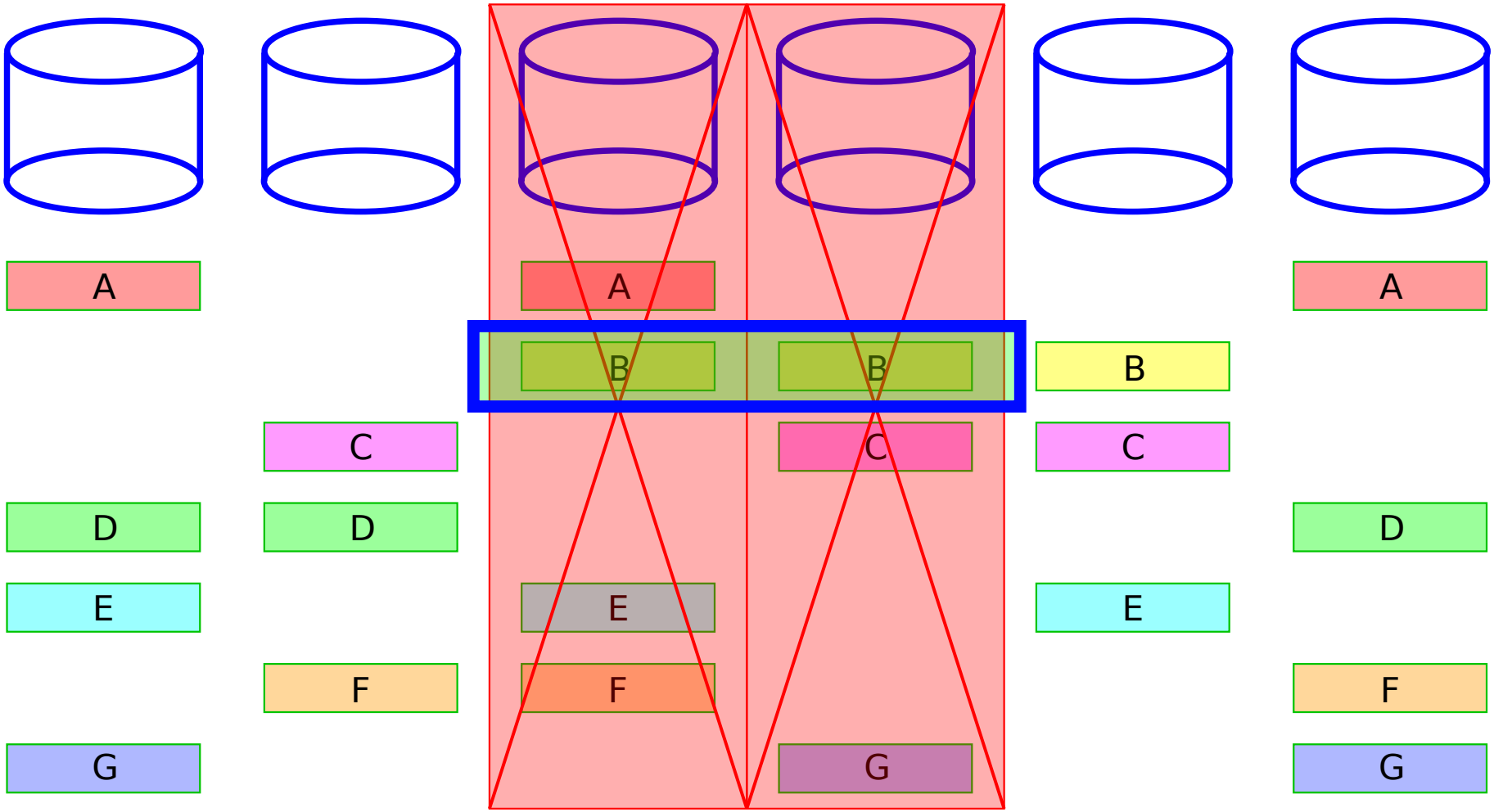
Declustered Placement - Single disk failure



Declustered Placement - Single disk failure



Declustered Placement - Dual disk failure



History of Ceph at Flatiron

Before Ceph

- Used small HDFS installation before - served a dozen or so users
  - Not POSIX compliant
- Evaluated alternatives: Lustre, GlusterFS, GPFS, Ceph

Initial Ceph setup

- Flatiron CephFS birthday: 4/10/2015
- Hammer (8th) release
- Small scale - half a dozen Dell T630 servers
  - 18 x 6TB spinning drives per node, no flash
  - collocated management processes (mon, mds)

Ceph - Scaling out

Research group grew - Ceph scaled well

- Space oriented design
- Using Dell DSS-7500 building blocks
  - 45 large capacity spinning drives (8TB, lately 12TB)
  - Two high performance NVMe drives for journaling (P3700, Optane)
  - Dual 40Gbps Ethernet connectivity
- Separate metadata storage - 1U R630 nodes with NVMe storage
- Separate monitor nodes

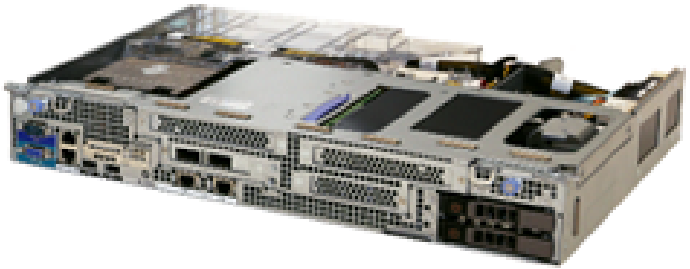
Current setup:

- 42 DSS-7500 nodes, about 30PB raw space
- 5 R640 monitor nodes
- 6 R630 metadata storage nodes

Flatiron Ceph Implementation

Dense storage nodes from Dell (DSS-7500)

- 90 x 8TB 7200rpm SAS drives, 2 x 28 core Broadwell servers in 4U dual 40Gbps ethernet



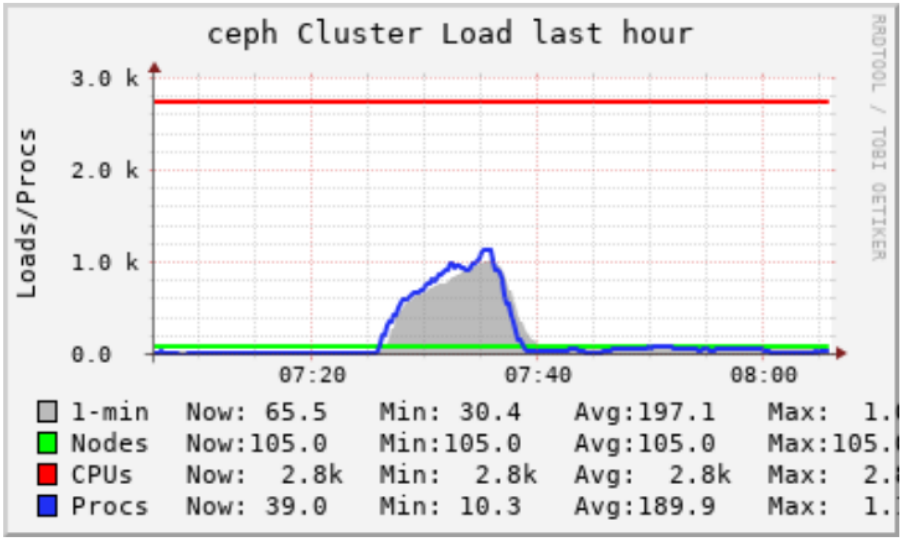
Flatiron Ceph Implementation

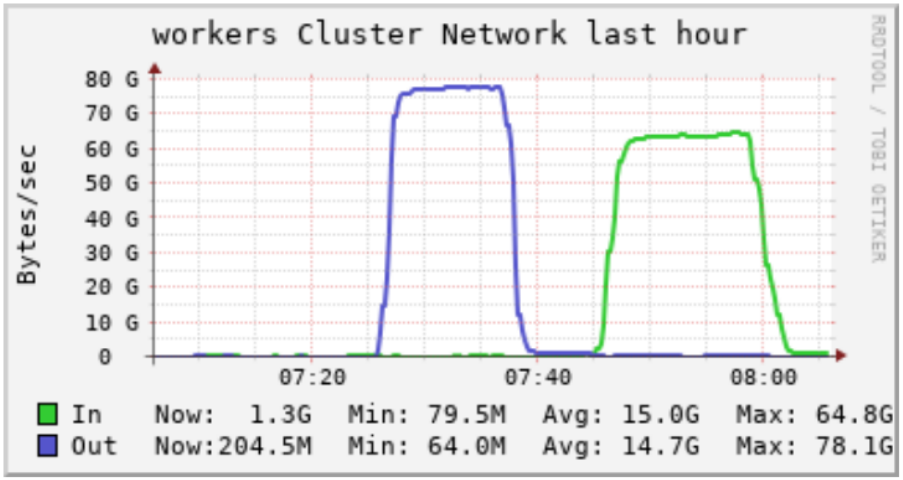
Flatiron Ceph Implementation

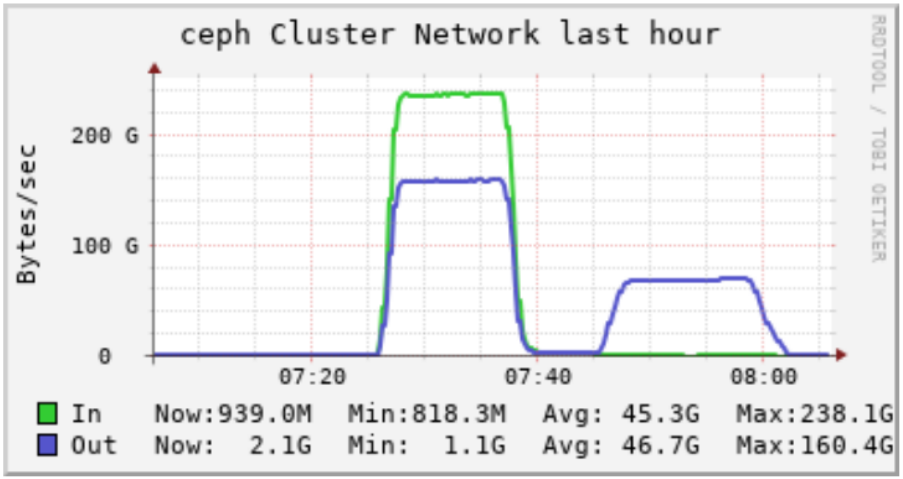
CephFS Performance - 1

Test setup:

- 36 DSS-7500 servers
- 64 client nodes with 10GbE
- Triple replicated pool
- Sequential write/read



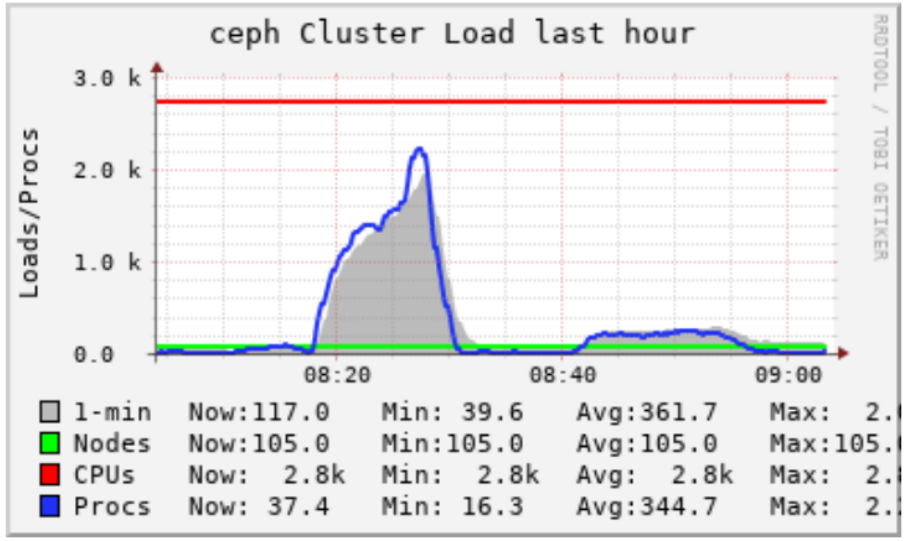


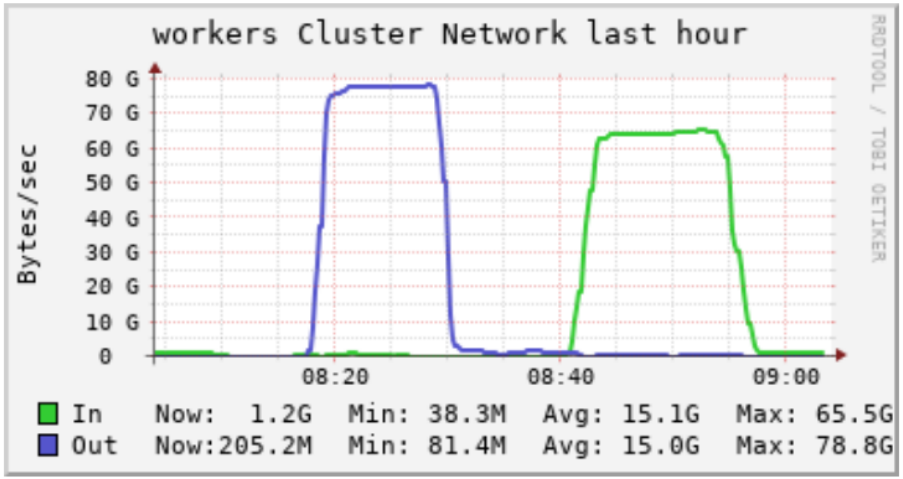


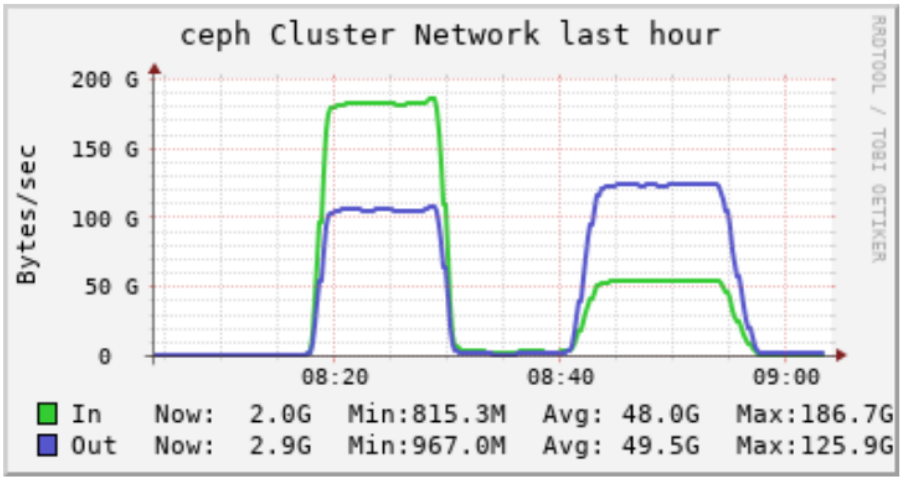
CephFS Performance - 2

Test setup:

- 36 DSS-7500 servers
- 64 client nodes with 10GbE
- 6+3 EC profile pool
- Sequential write/read



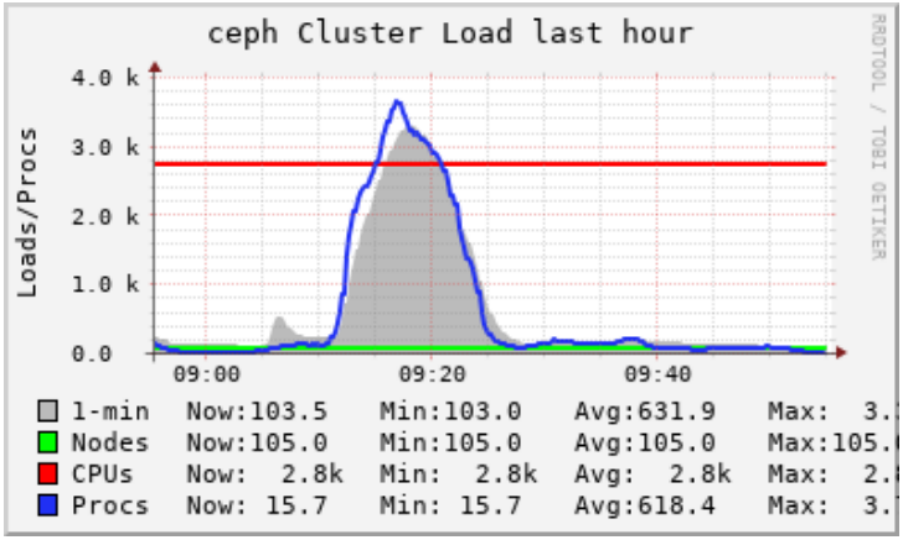


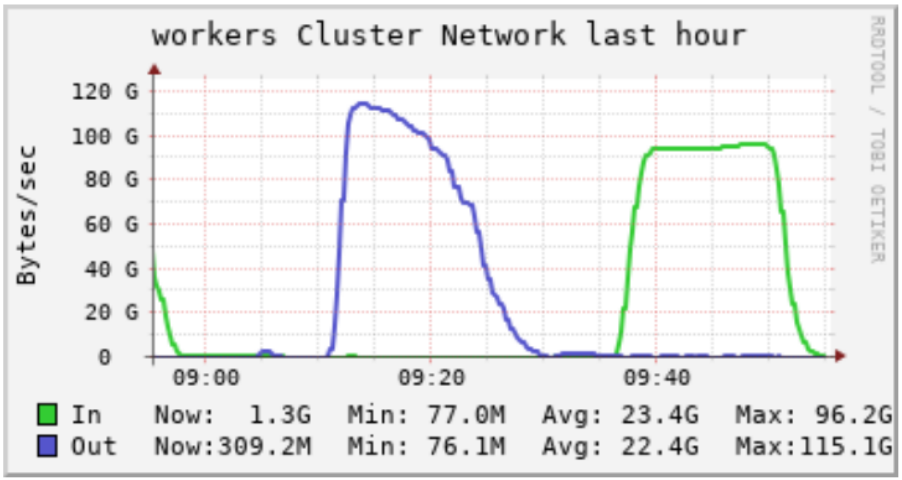


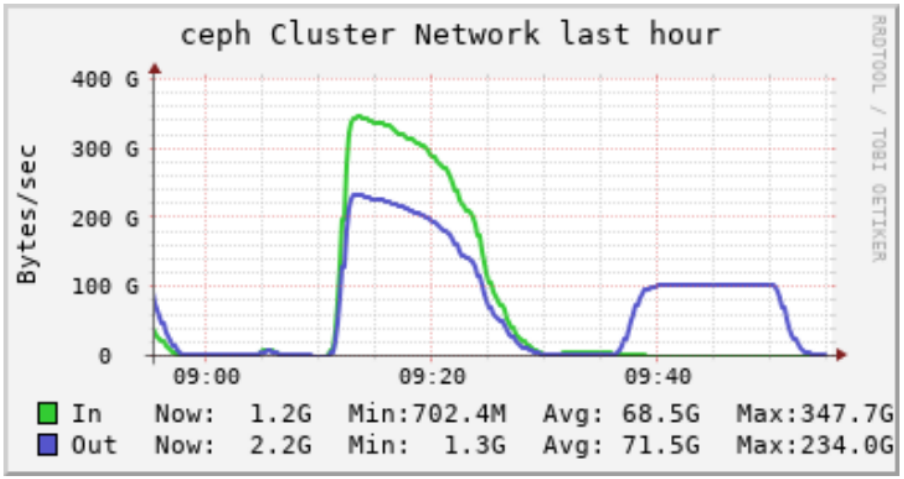
CephFS Performance - 3

Test setup:

- 36 DSS-7500 servers
- 96 client nodes with 10GbE
- Triple replicated pool
- Sequential write/read



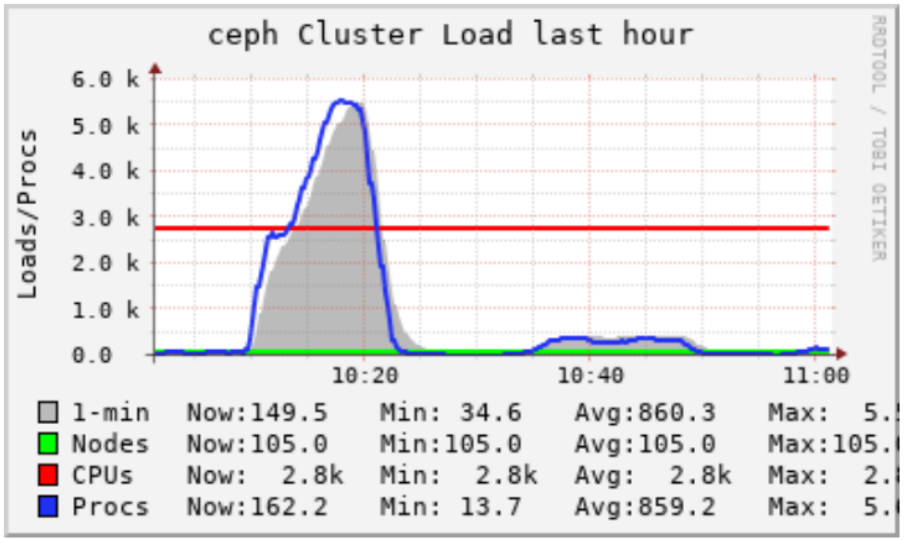


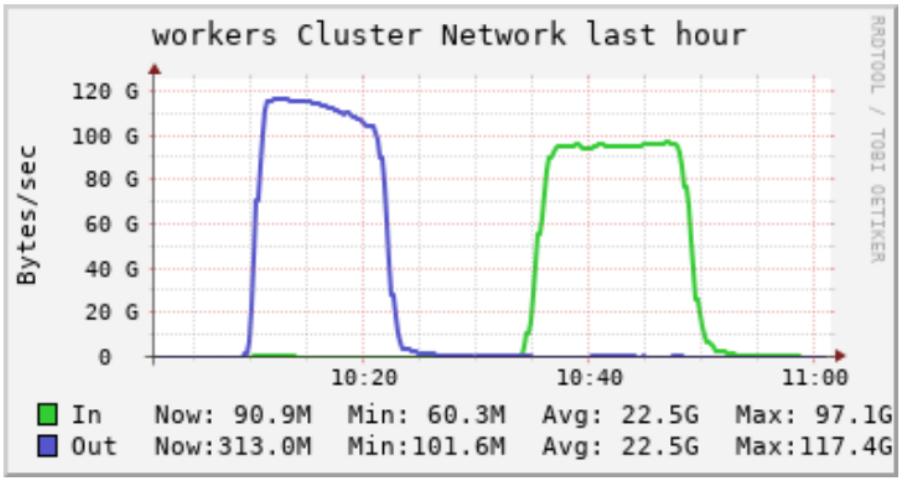


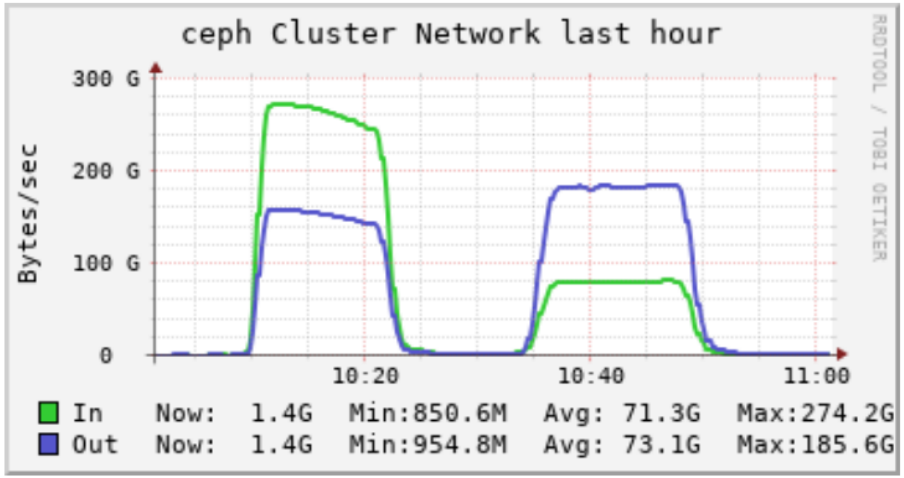
CephFS Performance - 4

Test setup:

- 36 DSS-7500 servers
- 96 client nodes with 10GbE
- 6+3 EC profile pool
- Sequential write/read







Flatiron Ceph Data Placement

Unique challenge of Flatiron Data Center

- In basement - on an island a few feet above sea level

Distributed Ceph storage nodes around the building

- Configured data placement using the flexibility of CRUSH
- No failure of a single area (room) will result in a loss of data
- Building is divided into 3 regions
  Loss of one region (such as the basement) results in no data loss
- It has a theoretical overhead of 50 percent

Encoding

- Triple replication - used for metadata and small files
- Erasure coding (6+3) - used for large files
- Actual overhead - very close to theoretical

Lessons learned - over years

Hardware

- Drives - some NAS grade drives corrupt data silently
  - Ceph detected and warned us about data corruption
- Networking - packet drops on 40Gbps interfaces
- NVMEs/SSDs
  - Consumer grade devices - lower performance than spinning drives

Software

- Ceph versions - some more stable than others
  - We started with Hammer (0.94.x), then Jewel (10.2.x), Luminous (12.2.x)
  - Recently upgraded to Mimic (13.2.6)
  - Upgrading to Nautilus (14.2.x) in 2019
- Kernel versions - issues with CentOS kernels
  - Running custom built kernel on Ceph storage nodes
- Mellanox firmware/software stack

Failures and Resilience

Typical failures:

- Single drive failures, read/write errors - Ceph handles automatically
- Sometimes node crashes - mostly hardware reasons
  - Manual intervention - no automatic recovery
- DIMM replacements
- Rarely - NVMe failure, SAS controller failure

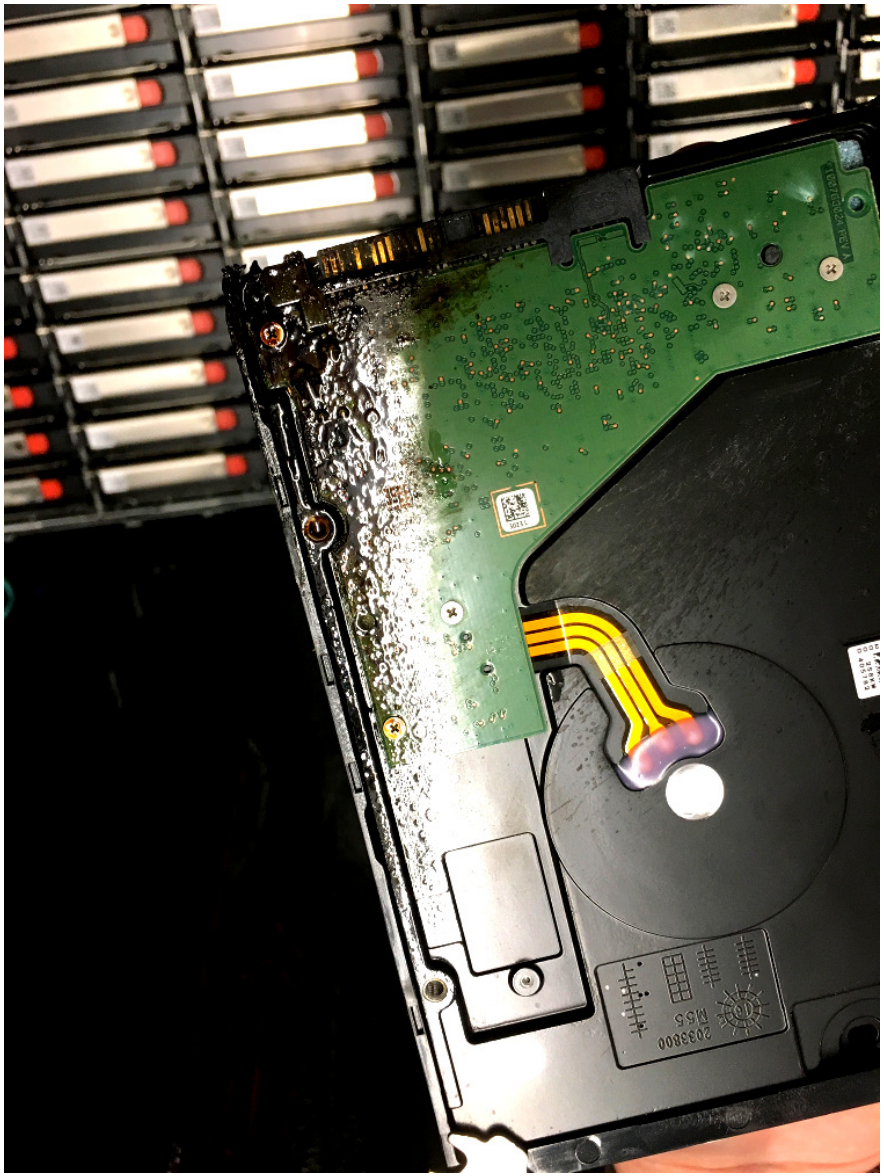
Availability:

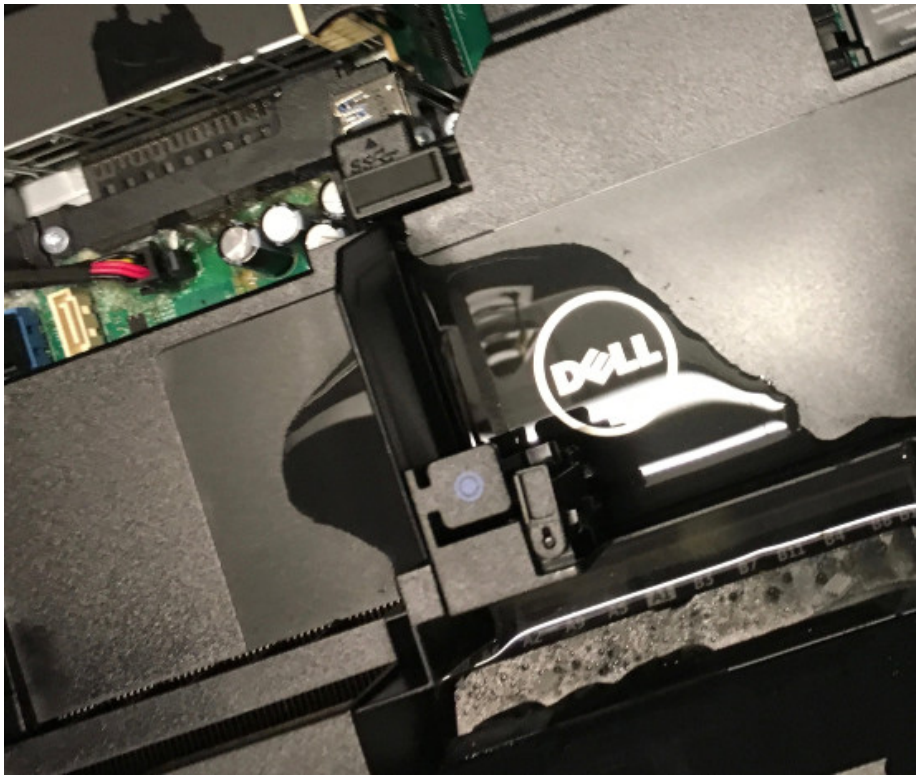
- With one exception - we had no unplanned outages on the DSS-7500 setup
- We moved Ceph data to our new building without interrupting availability
- We have done upgrades of ceph without a shutdown

Real Disaster Recovery:

- Tenant above one of our floors left the water running over a weekend
- One of the data closets got flooded with a 90 drive ceph node in it
- Water + electricity $\longrightarrow$ trouble
- However - we lost no data

Flooded ceph node + electrical fire





Flatiron Customizations

Usage monitoring

- Real time usage monitoring is a challenge with most distributed FS's
- We run a modified Ceph client - collects real time usage statistics
- Makes it possible to identify problematic jobs
  - examples: opening thousands of files a second, doing small I/O

Custom patches for issues/enhancements

- Every so often - testing pre-release features, bug fixes, enhancements
- Open source development model makes bug fix cycle much shorter

Erasure coding conversion

- All files written as triple replicated originally
- Periodic parallel file system scan
  - identifies eligible files and converts them to EC 6+3
- EC files: $< 5\%$ by count, $> 95\%$ by space
- Talked to developers about moving files across pools in the MDS

Flatiron Customizations

Ceph current performance

Aggregate by: user Averaging timescale: 10 seconds

Columns | Refresh | Auto-refresh: OFF Last updated: 2019-09-09 10:48:22 Search:

host	user	file	open (files/s)	read (MB/s)	write (MB/s)	total (MB/s)
	[illegible]		0.007	0.000	0.000	0.000
	a[illegible]		1.057	478.136	17440.979	17919.114
	a[illegible]		0.000	0.000	0.001	0.001
	c[illegible]		0.000	0.000	0.000	0.000
	c[illegible]		0.004	0.000	0.063	0.063
	c[illegible]		0.104	0.001	41.660	41.661
	c[illegible]		2.196	280.640	0.000	280.640
	d[illegible]		0.000	0.000	0.001	0.001
	e[illegible]		920.015	0.267	0.860	1.126
	g[illegible]		0.003	1024.881	0.000	1024.881
	jl[illegible]		0.083	0.000	19.306	19.306
	j[illegible]		0.001	0.000	6.317	6.317
	k[illegible]		0.003	0.000	0.043	0.043
	k[illegible]		0.427	0.019	0.000	0.019
	n[illegible]		0.122	0.000	0.000	0.000
	n[illegible]		3.701	0.000	60.887	60.887
	n[illegible]		0.000	0.000	0.000	0.000
	s[illegible]		0.000	88.117	0.000	88.117
	v[illegible]		0.002	0.000	0.000	0.000
	v[illegible]		0.000	0.000	0.001	0.001
	y[illegible]		0.000	0.000	0.000	0.000
total	**total**	**total**	**927.742**	**1872.060**	**17570.118**	**19442.177**

Current Challenges

Small file performance

- Use kernel client
  - Stability
  - Kernel version dependence
  - Usage monitoring instrumentation
- Ceph Octopus planned improvements - small file creation/removal
  - Testament to flexible design of Ceph

Future ceph building blocks

- DSS-7500 is a bit disk heavy - has not seen any architecture updates
- Flash storage node
  - Especially when small file performance improves
  - Ceph has a project (Crimson) - OSD optimized for low latency flash

Longer term

- HSM like functionality - moving old data to tape or colder storage

Questions