NETFLIX

C* Upgrades at Scale

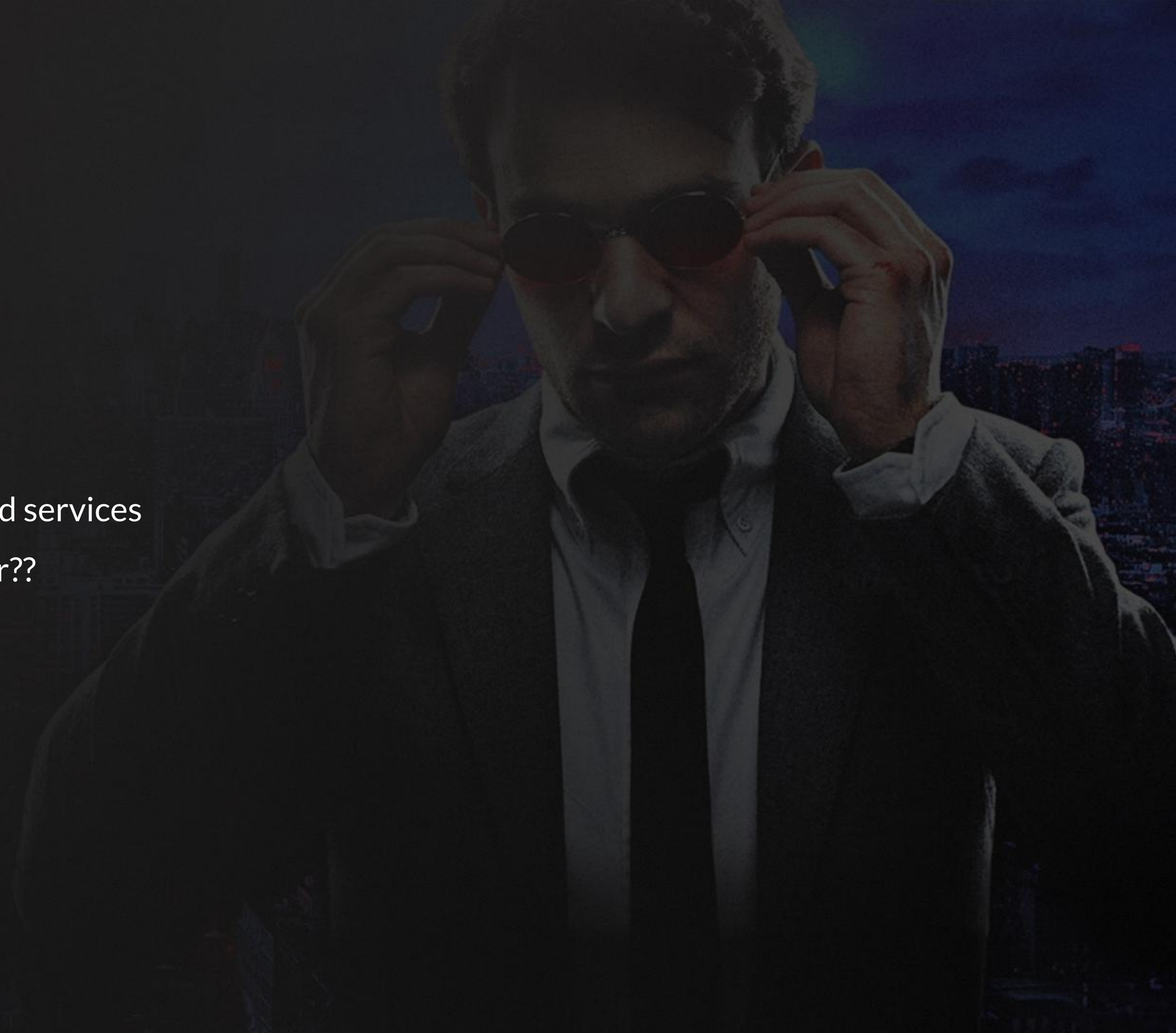
Ajay Upadhyay & Vinay Chella
Data Architects

NETFLIX

Who are We??

- Cloud Database Engineering [CDE]
- Providing Cassandra, Dynomite, Elastic Search and other data stores as a service
- Ajay Upadhyay
  - Data Architect @ Netflix
- Vinay Chella
  - Data Architect @ Netflix
  - https://www.linkedin.com/in/vinaykumarchella

Agenda



- About Netflix
- Cassandra @ Netflix
- What is our C* scale ??
- What do we upgrade, Why?
- How does it impact applications and services
- What makes C* upgrades smoother??
- Infrastructure for upgrade
- Online Upgrade process
- Pappy framework
- Migration Hiccups/ Roadblocks

NETFLIX

About Netflix

- Started as DVD rental
- Internet delivery/ streaming of TV shows and movies directly on TVs, computers, and mobile devices in the United States and internationally

Cassandra @ Netflix

- 90% of streaming data is stored in Cassandra
- Data ranges from customer details to Viewing history to streaming bookmarks
- High availability, Multi-region resiliency and Active-Active

What is our scale (C* Footprint)

235

NETFLIX

What is our scale (C* Footprint)

Continue...

7000+

NETFLIX

Duration of the upgrade???

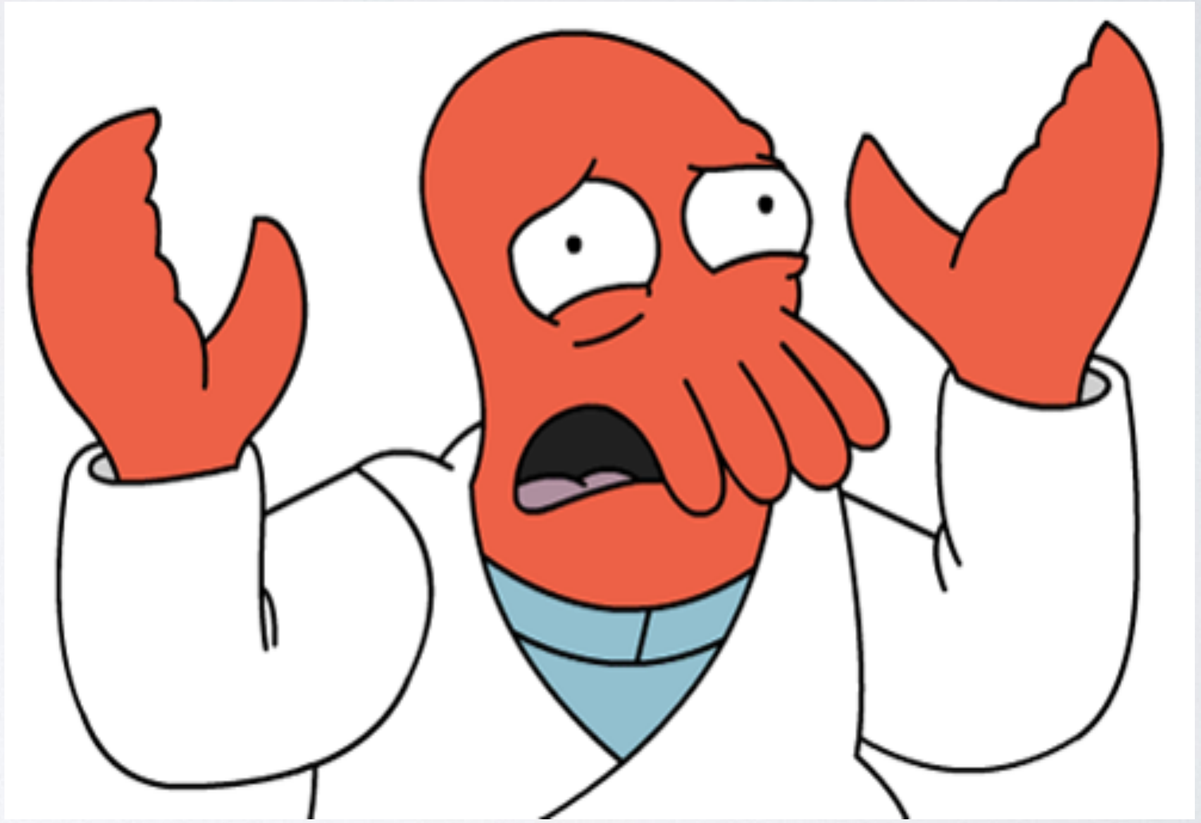
What do we upgrade??

- Software
  - Cassandra
  - Priam
  - OS
  - Other sidecars
- Hardware
  - AWS instance types

What do we upgrade, Why?

Continue...

Why new release (c* 2.0)? What's wrong with (1.2)?



What do we upgrade, Why? continues...

- Typically nothing wrong with current release (in this case 1.2…)
- Pretty stable and few clusters are still running current release without any issues
- Still supported by Datastax but new features and bug fixes are not being done any more
- End of life cycle

What do we upgrade, Why? continues...

- Good to be on latest and stable release
- Lots of great features
- Bug fixes and enhancements / new features are being only added to the latest release

Application and service level impact/change

- Before Migration
- During migration
- After migration

Application and service level impact/change

- None, if app is using latest Netflix OSS components (Astyanax/ BaseServer)

During the upgrade???

NETFLIX

What makes C* upgrades smoother and transparent

How do we certify a binary??

NETFLIX

Functionality….

NETFLIX

Is any existing functionality broken?

How do we certify a binary for the final upgrade

Simulate PROD

- Use pappy framework to simulate prod traffic on test clusters
- Soak it, make sure traffic pattern is same in terms of R/W OPS, storage and AWS instance type

How do we certify a binary for the final upgrade

Typical usage patterns:

- Read Heavy
- Write Heavy
- STCS
- LCS
- Aggressive TTLs
- Variable Payloads

How do we certify a binary for the final upgrade

continues...

During Upgrade:

- Start upgrading one of the clusters to new release, keeping same traffic
- Any issues, get new patch, redo whole migration
- Process is fully automated, takes few minutes to couple of hours to redo entire test

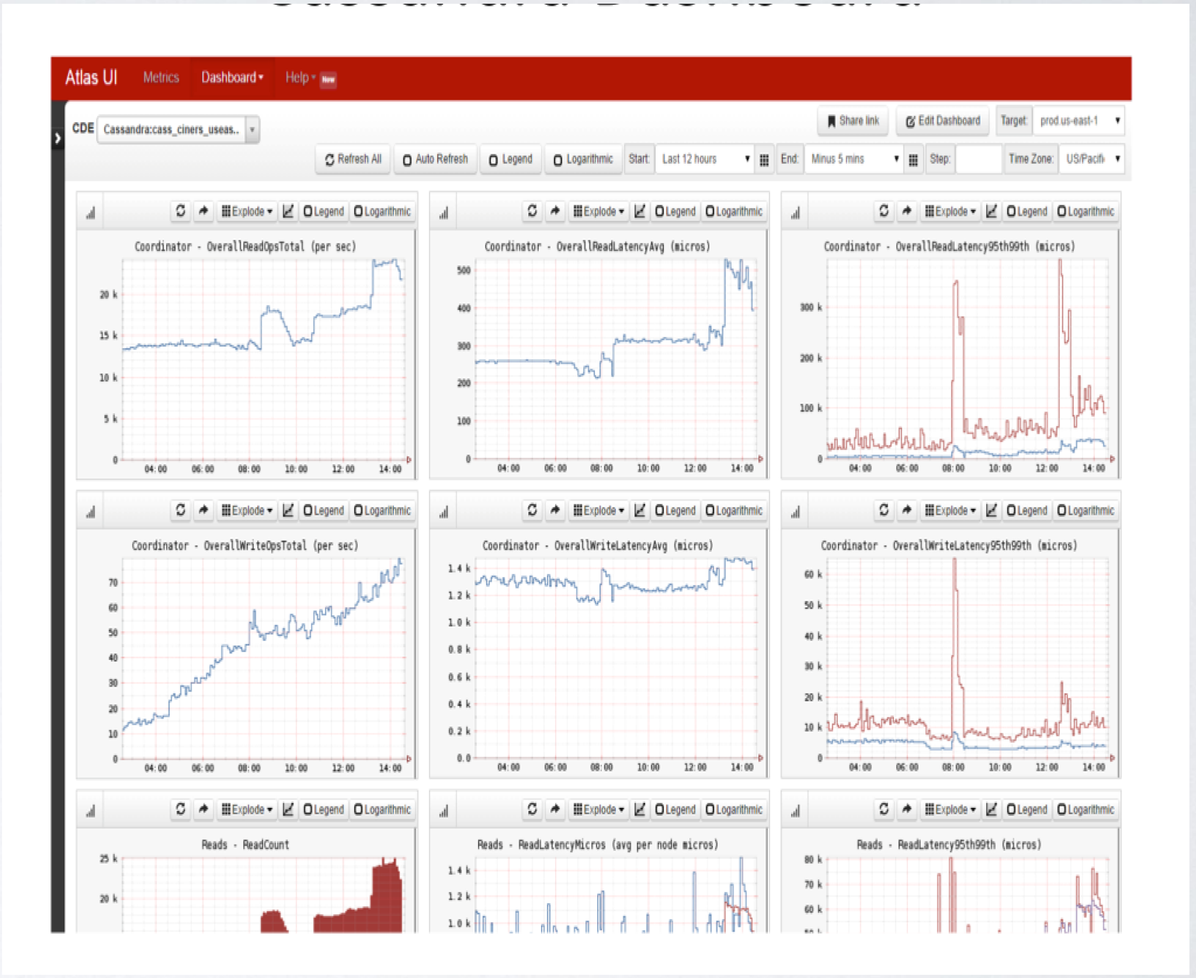
How do we certify a binary for the final upgrade

continues...

Summary

- Functional validation
- Performance testing
- Load test with routine maintenance tasks
  - Repair and Compaction
  - Replacement and Restarts
- Key Metrics Used
  - 95th and 99th Latencies
  - Dropped Metrics on C*
  - Exceptions on C*
  - Heap Usage on C*
  - Threads Pending on C*

Infrastructure for upgrade...



Infrastructure for upgrade...

- Health Monitoring Jobs
- Regular Maintenance Jobs
  - Repairs
  - Compactions
- Upgrades
  - Sizing
  - Instance Types
  - OS
  - Sidecar
  - C*

Online upgrade process (steps)

Pre-Upgrade:

1. Check for the forgotten files
2. Check for currently running (maintenance) jobs
3. Disable all the maintenance jobs until the duration of upgrade process
   a. Compact
   b. Repair
   c. Terminate
4. Disable Monkeys
5. Download the DSE tarball
6. Make sure that the cluster is healthy before starting upgrade

Online upgrade process (steps)

Upgrade:

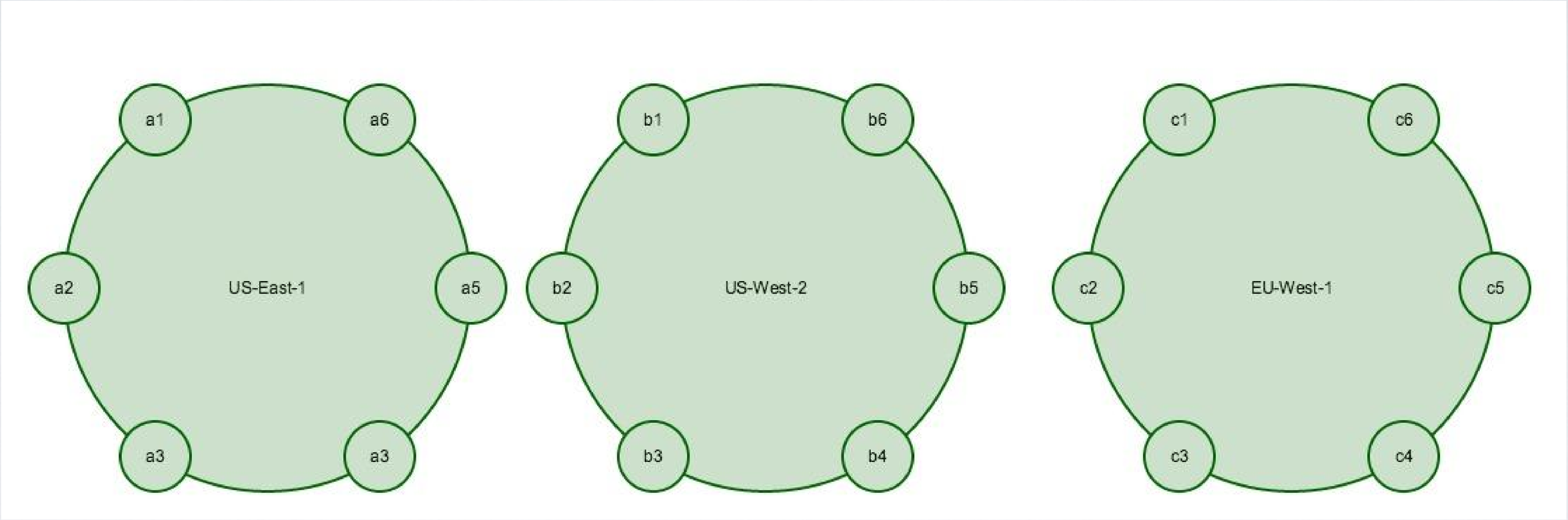
- Running binary upgrade of Priam and C* on a node
  - Continue to make sure that the cluster is still healthy
  - Disable node in Eureka
  - Disable thrift, gossip and run nodetool drain
  - Stop Cassandra and Priam
  - Install new C* package (tarball)
  - Make sure the cluster is still healthy
    - Disable/Enable gossip if new node does not see all others vice-versa
  - Verify “nodetool version”
- Make sure the cluster is still healthy before next step

Online upgrade process (steps)

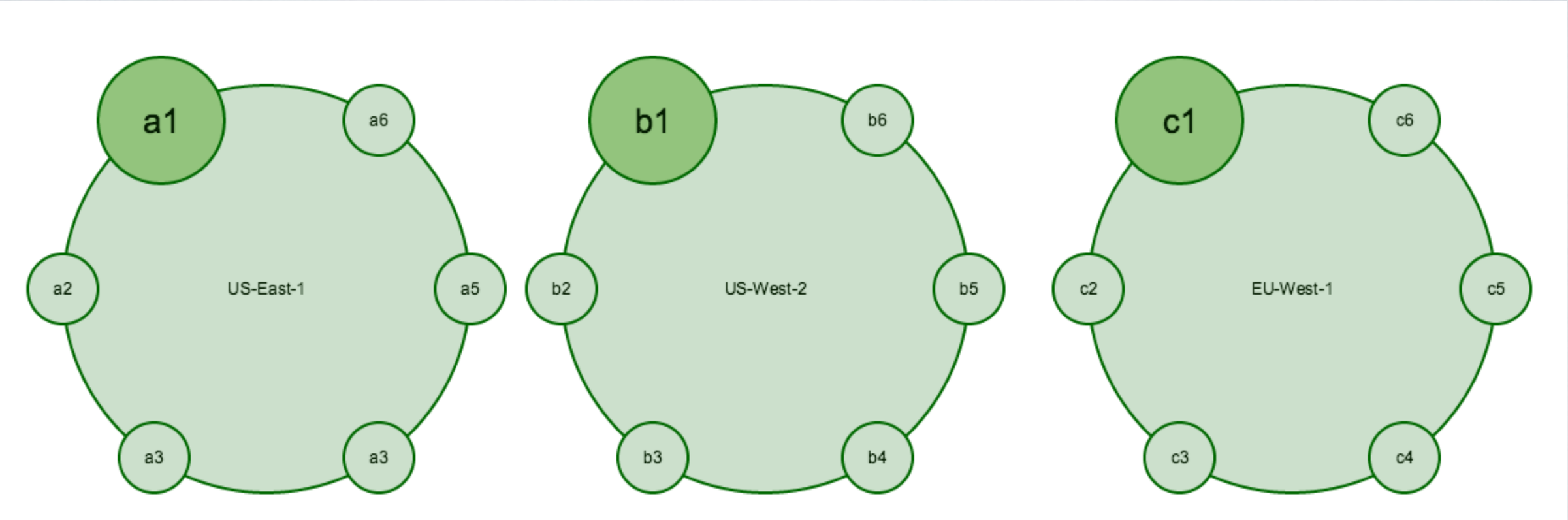
Post-Upgrade:

1. Alter CF settings (for all Keyspace/CF): speculative_retry = 'NONE' and index_interval = 256
2. Run SSTABLE Upgrade on all nodes (STCS - parallel, LCS - Serial)
3. Enable Chaos Monkey
4. Make sure the cluster is still healthy before considering the upgrade done

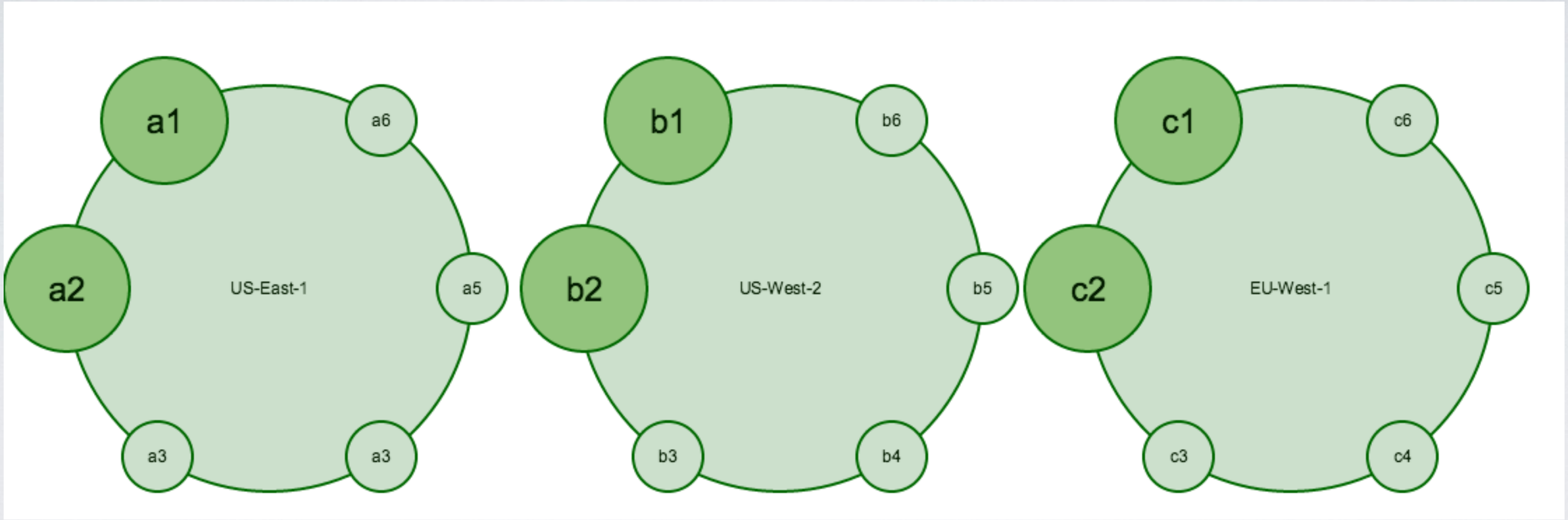
Online upgrade process



One node at a time - parallelly in all regions



One zone at a time - parallely in all regions



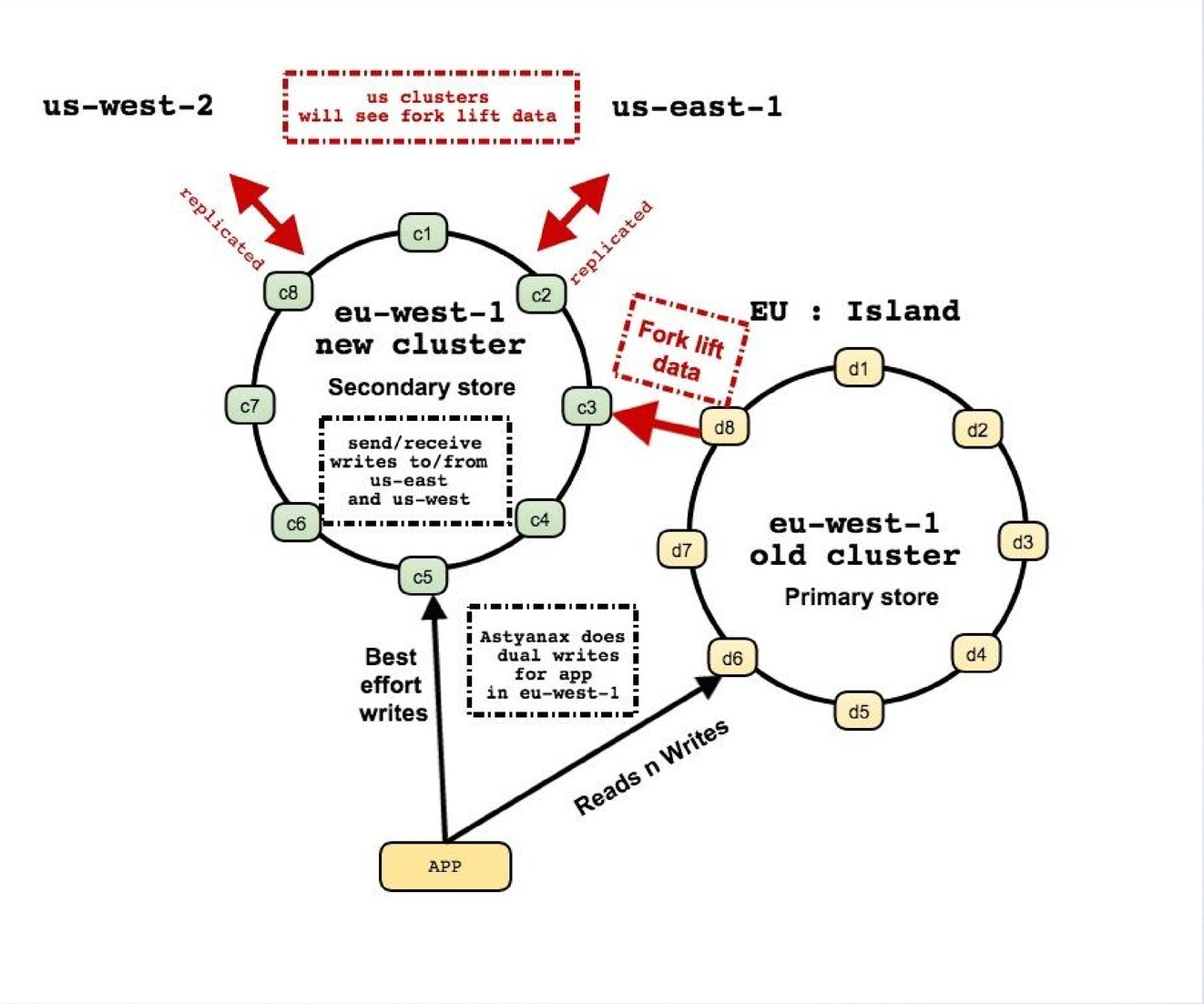
One zone at a time??

NETFLIX

Binary upgrades??
Why not replacements??

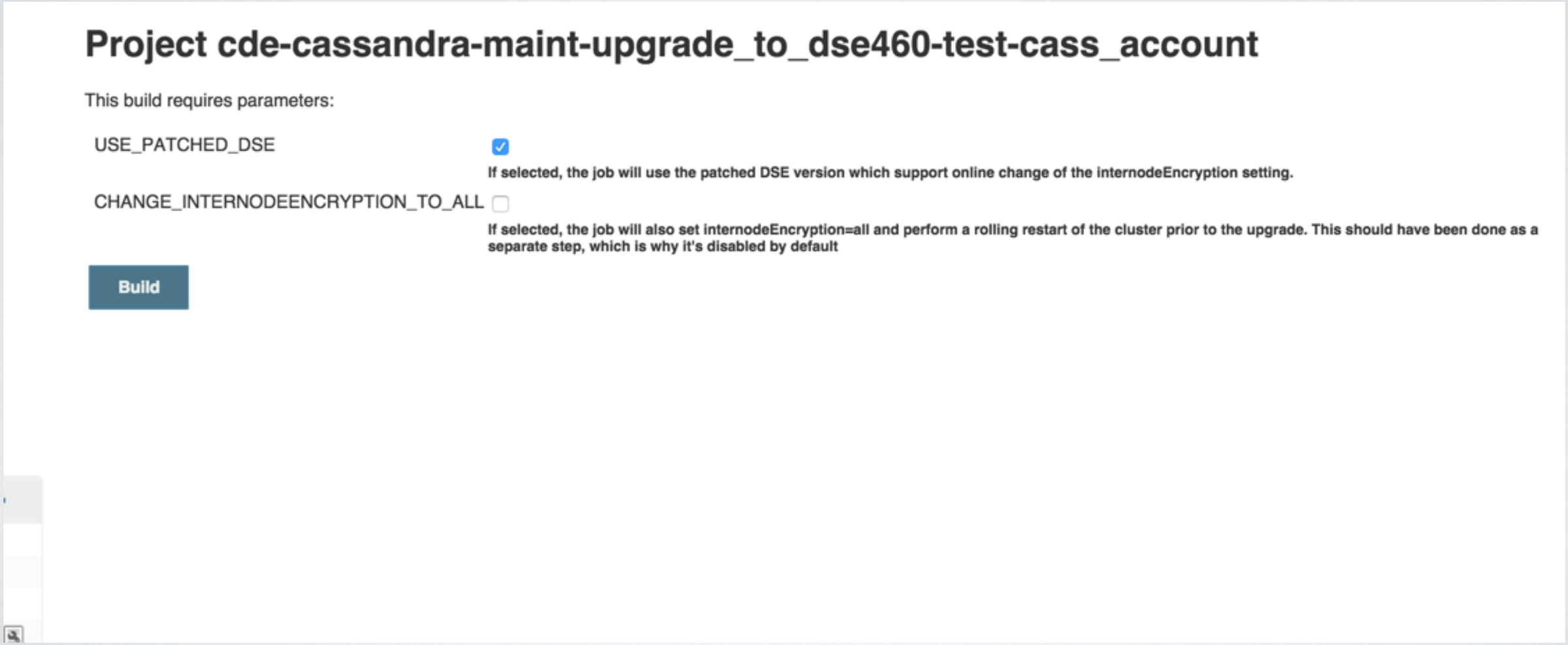
Dual Writes - Forklift (Pappy Framework)

- For short lived data and big clusters
- Astyanax - Dual Writes
- C* Forklifter - Pappy Framework



NETFLIX

Click of a button (Just 1 button….)



What about performance??

NETFLIX

Pappy Framework.....

NETFLIX

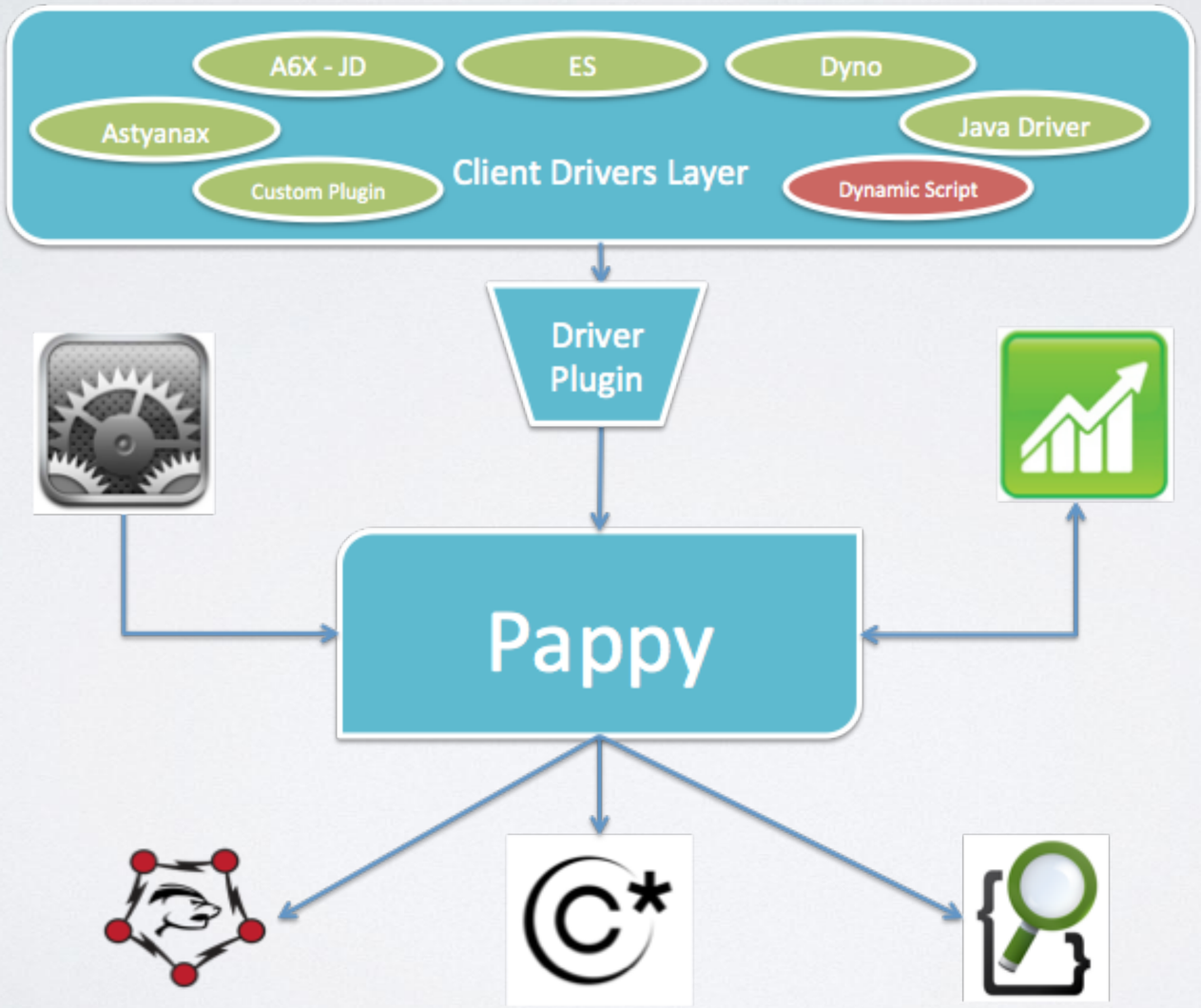
What is Pappy??

- CDE's new Test harness framework
- Helps CDE moving fast with load / performance tests
- Netflix homegrown: Governated, well integrated with netflix OSS infrastructure (Archaius, Servo, Eureka and Karyan)
- Side by side comparison of different performance runs

High level features

- Collects metrics for
  - Various Instance types comparison
  - Varying data models in terms of payload, shard and comparators
  - Different driver(s) versions
- Pluggable architecture enables working with various datastores (Cassandra, Dyno, ES etc.,)

Architecture



Performance testing...

- More than 10 rounds of performance testing
- Yaml tuning
  - Read heavy - Write heavy
  - Write heavy
  - Read heavy

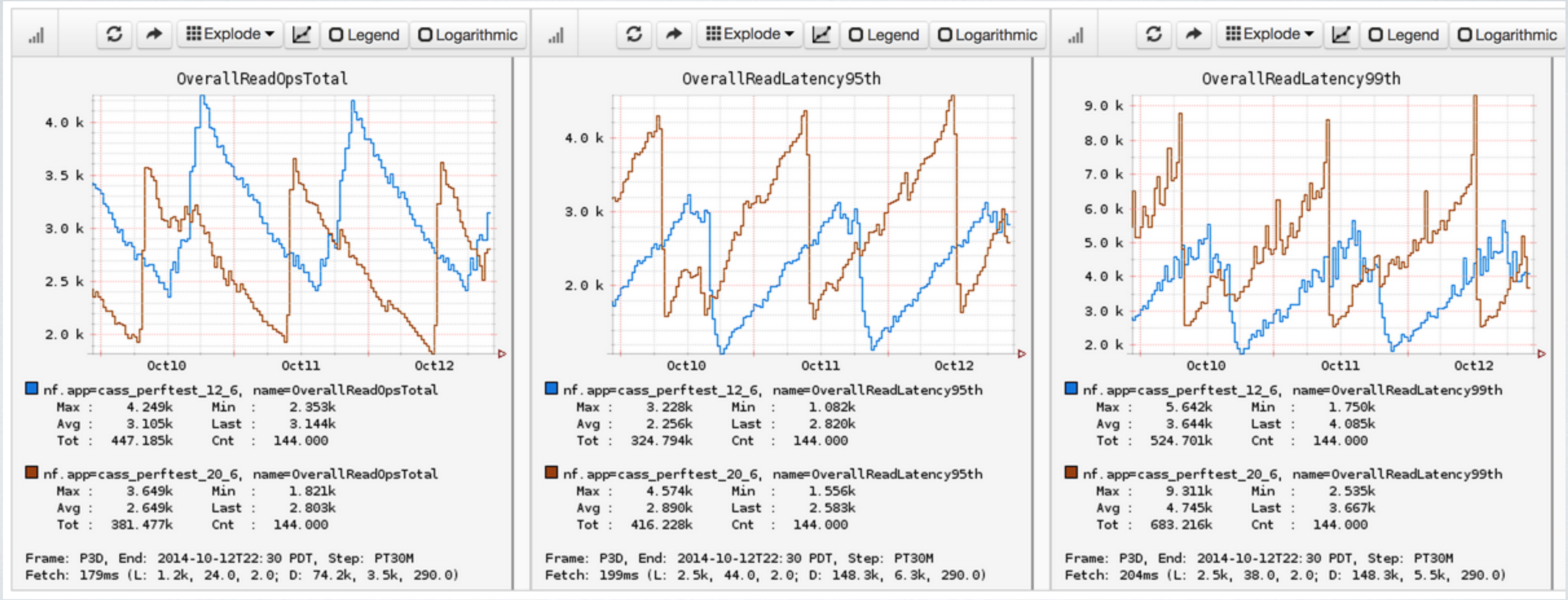
Test Setup #1

Row Size:	1KB (size-100*cols-10)
Data Size:	30GB (Keys: 20,000,000)
Read:	4K (3.6 K)
Writes:	1 K (700)
Instance:	i2.xl (4 CPU, 30 GB RAM)
Nodes:	6
Region:	Us-East
Read CL	CL_ONE
Write CL	CL_ONE
Heap	12 GB / Newgen: 2GB

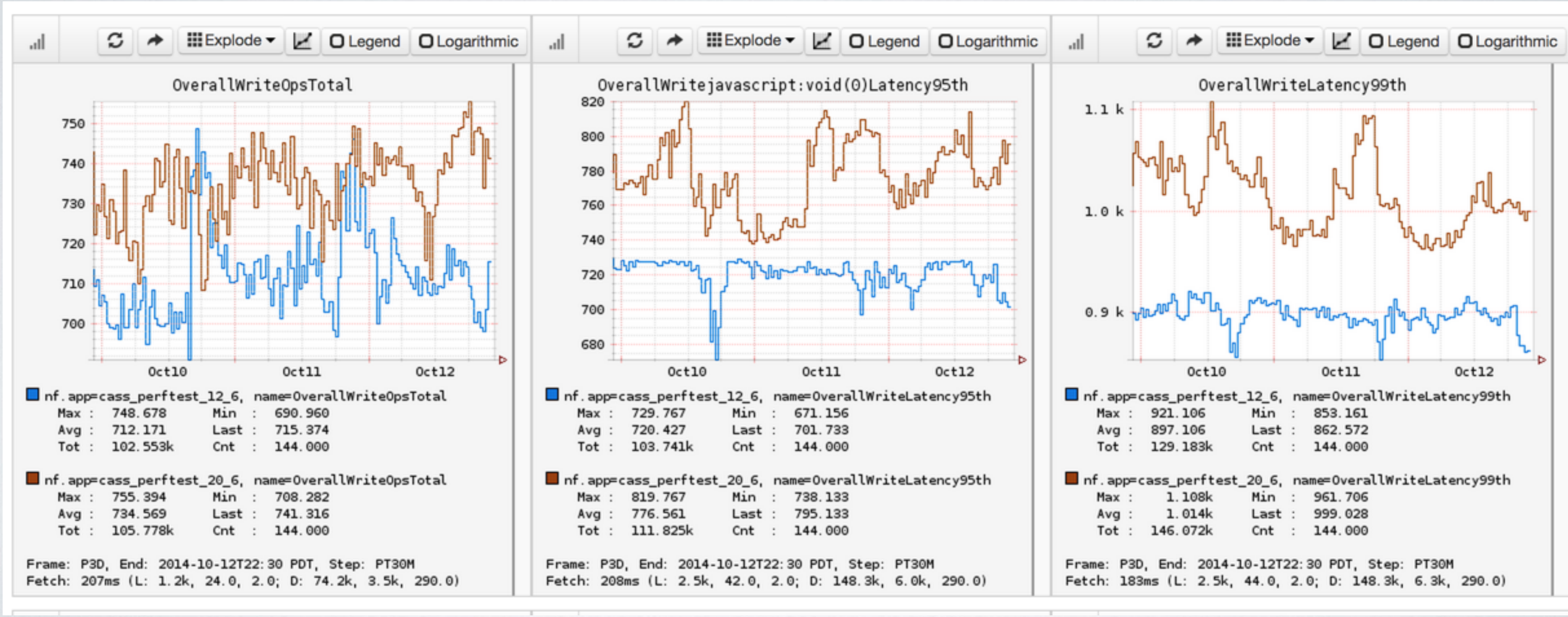
Test Setup #1 - Results

Test Setup #1	**C* 1.2**	**C* 2.0**
ROps	3.1 K - 4.0 K / sec	2.6 - 3.5 K/ sec
WOps	710 /sec	730 /sec
Read 95th	2.25k micro	2.89k micro
Read 99Th	3.6k micro	4.7k micro
Write 95th	720 micro	770 micro
Write 99Th	890 micro	1.01 k micro
Data / Node	30 - 36 GB	30 - 36 GB

Read Metrics



Write Metrics



Summary…

Metrics	Test Setup #1(12G Heap/ 30G)		Test Setup #2(8GB Heap/~30G)		Test Setup #3 (8GB Heap/ 60GB data)	
	C* 1.2	C* 2.0	C* 1.2	C* 2.0	C* 1.2	C* 2.0
ROps	3.1K - 4.0K/ sec	2.6 - 3.5 K/ sec	3.5K - 4.0K/ sec	3.0 - 3.5 K/ sec	2.5 K - 3.5 K / sec	2.0 K - 3.0 K/ sec
WOps	710 /sec	730 /sec	720 /sec	710 /sec	700 /sec	620 /sec
Read 95th	2.25k micro	2.89k micro	1.7k micro	2.1k micro	2.4k micro	3.5k micro
Read 99Th	3.6k micro	4.7k micro	2.6k micro	3.7k micro	9.0k micro (spikes 20k)	9.5k micro (spikes 19K)
Write 95th	720 micro	770 micro	730 micro	910 micro	730 micro	820 micro
Write 99Th	890 micro	1.01 k micro	910 micro	1.2 k micro	940 micro	1.1 k micro
Data / Node	30 - 36 GB	30 - 36 GB	30 - 36 GB	30 - 36 GB	60 GB	60 GB

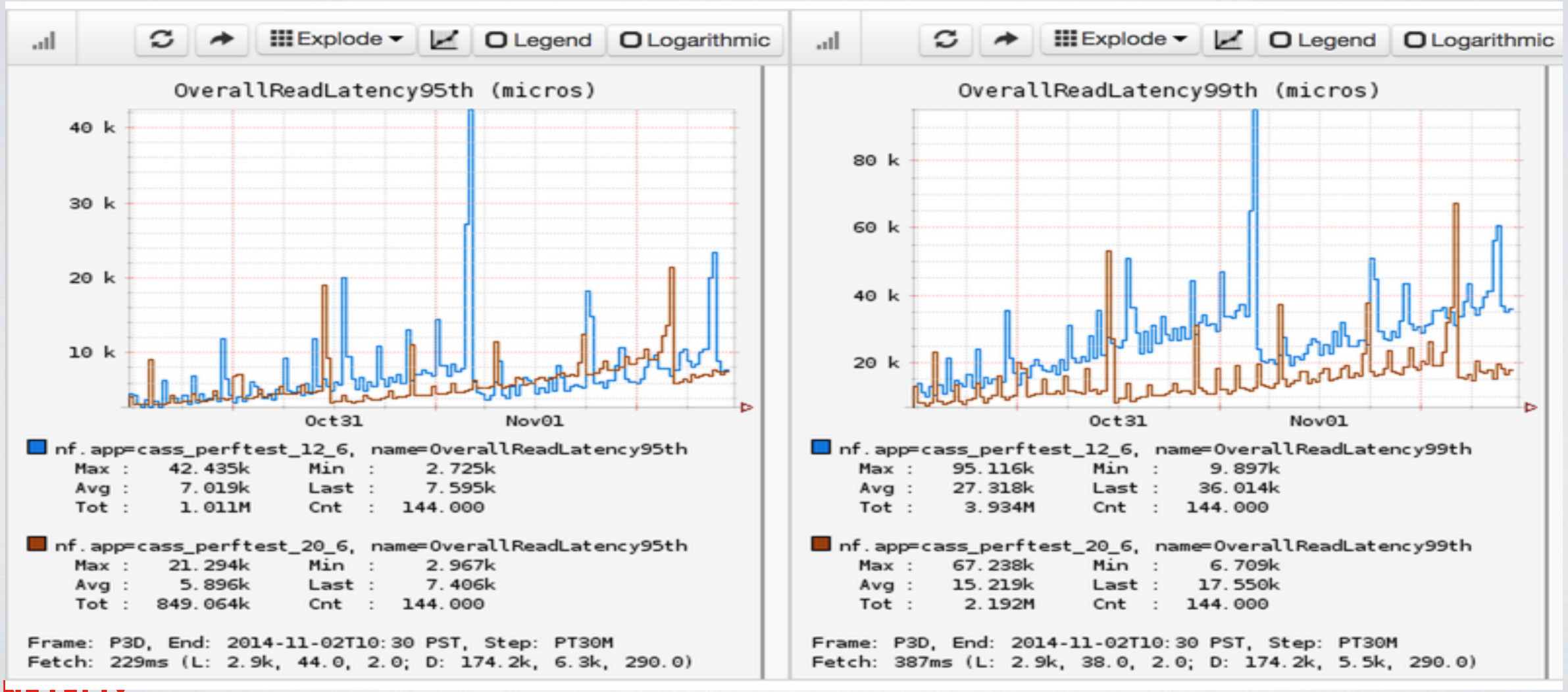
Wave - 2 Testing

1. speculative_retry (New feature in 2.0)
2. For low connections, use rpc_server_type = sync
3. index_interval = 256 --> default value has changed.
4. internode_compression = all
5. memtable_flush_writers: 4
6. newgen -> 800mb
7. Heap: 12 GB - 2 GB(new gen)
8. read_repair_chance=0 and dclocal_read_repair_chance = 0

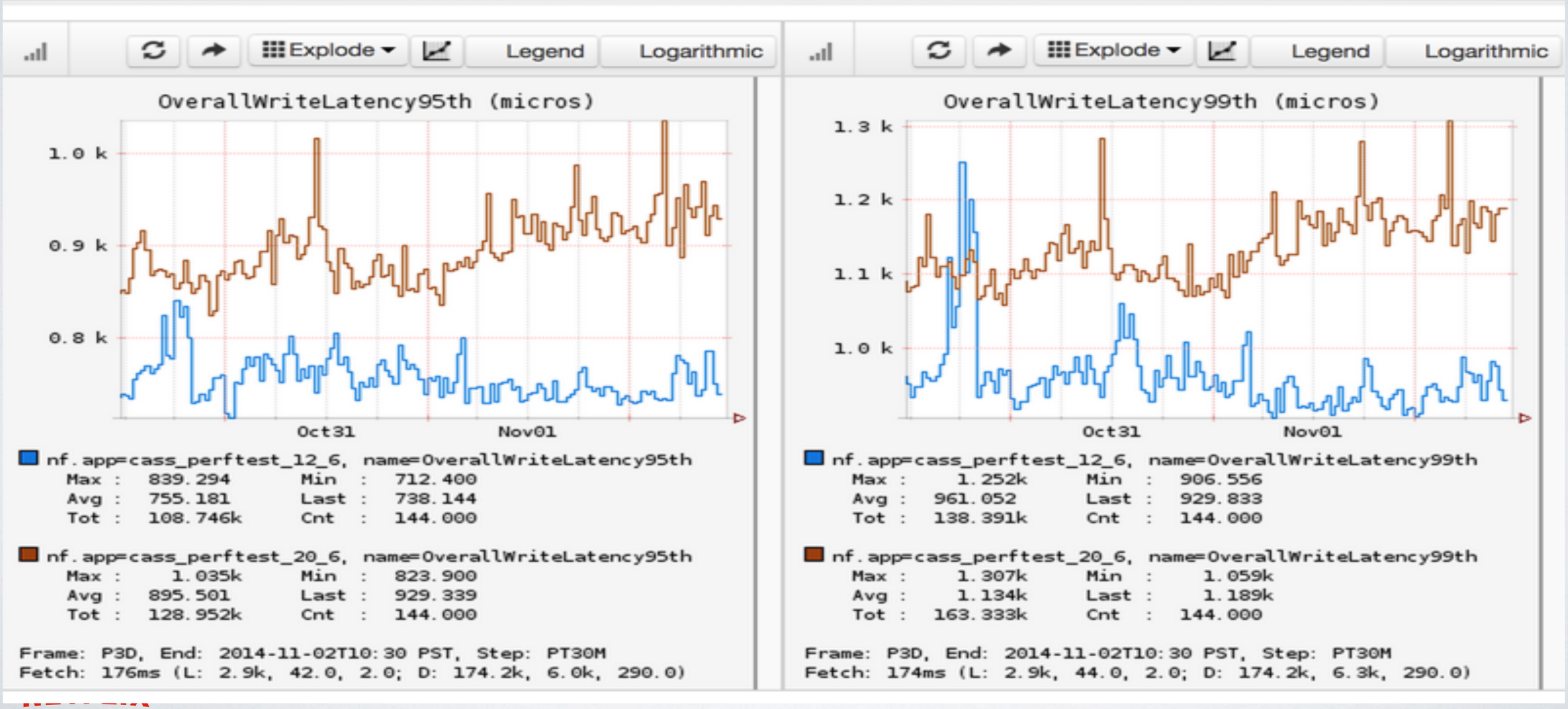
Test Setup #5 - Results (Duration: 2 Days)

Test Setup #5	**C* 1.2**	**C* 2.0**
ROps	3.8 K / sec	3.8 K/ sec
WOps	198 /sec	198 /sec
Read 95th	**7.01**k micro	**5.89**k micro
Read 99Th	**27.3**k micro (spikes upto 80k)	**15.2**k micro (spikes upto 60 k)
Read Avg	**2.15**	**2.04**
Write 95th	*755* micro	*895* micro
Write 99Th	*961* micro	*1.1* k micro
Write Avg	**396**	**444**
Data / Node	60 GB	60 GB

Test Case #5: Read 95th and 99th



Test Case #5: Write 95th and 99th



Teams Engaged

- Performance Team
- Several Application Teams

Few Migration Hiccups/ Roadblocks

- InterNodeEncryption = ‘ALL’ => why do we need this?
  - https://issues.apache.org/jira/browse/CASSANDRA-8751
- Hints causing Heap disaster
  - https://issues.apache.org/jira/browse/CASSANDRA-8485.


NETFLIX
Thank You
QUESTIONS???


NETFLIX
is
Hiring
jobs.netflix.com/#/jobs/1953/
or
jobs.netflix.com