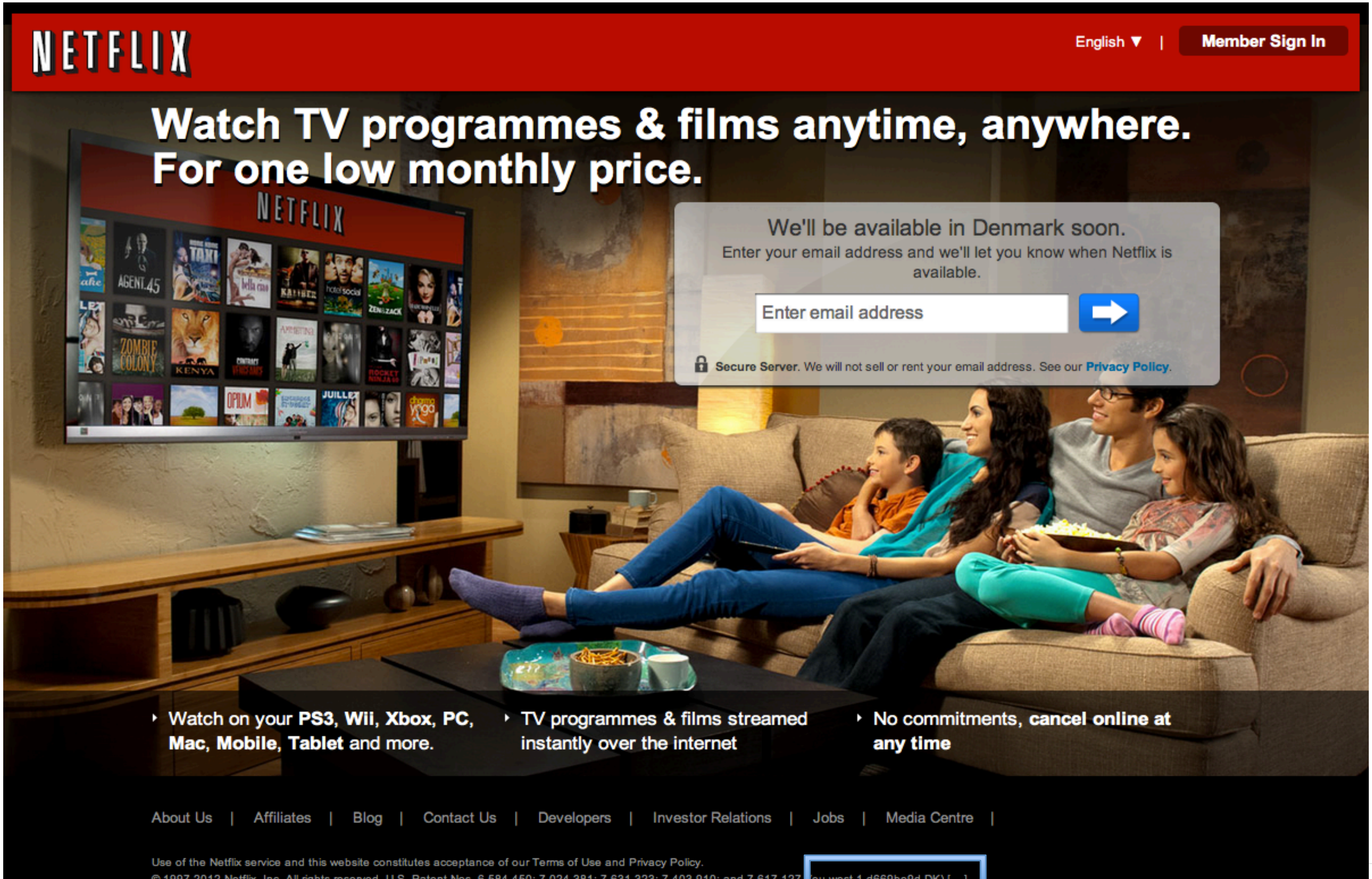
Running Netflix on Cassandra in the Cloud

www.netflix.com in DK



hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Blah Blah  Blah

(I’m skipping all the cloud intro etc. did that yesterday… Netflix runs in the cloud, if you hadn’t figured that out already you aren’t paying attention and should go read slideshare.net/netflix)

Netflix Deployed on AWS

amazon web services™

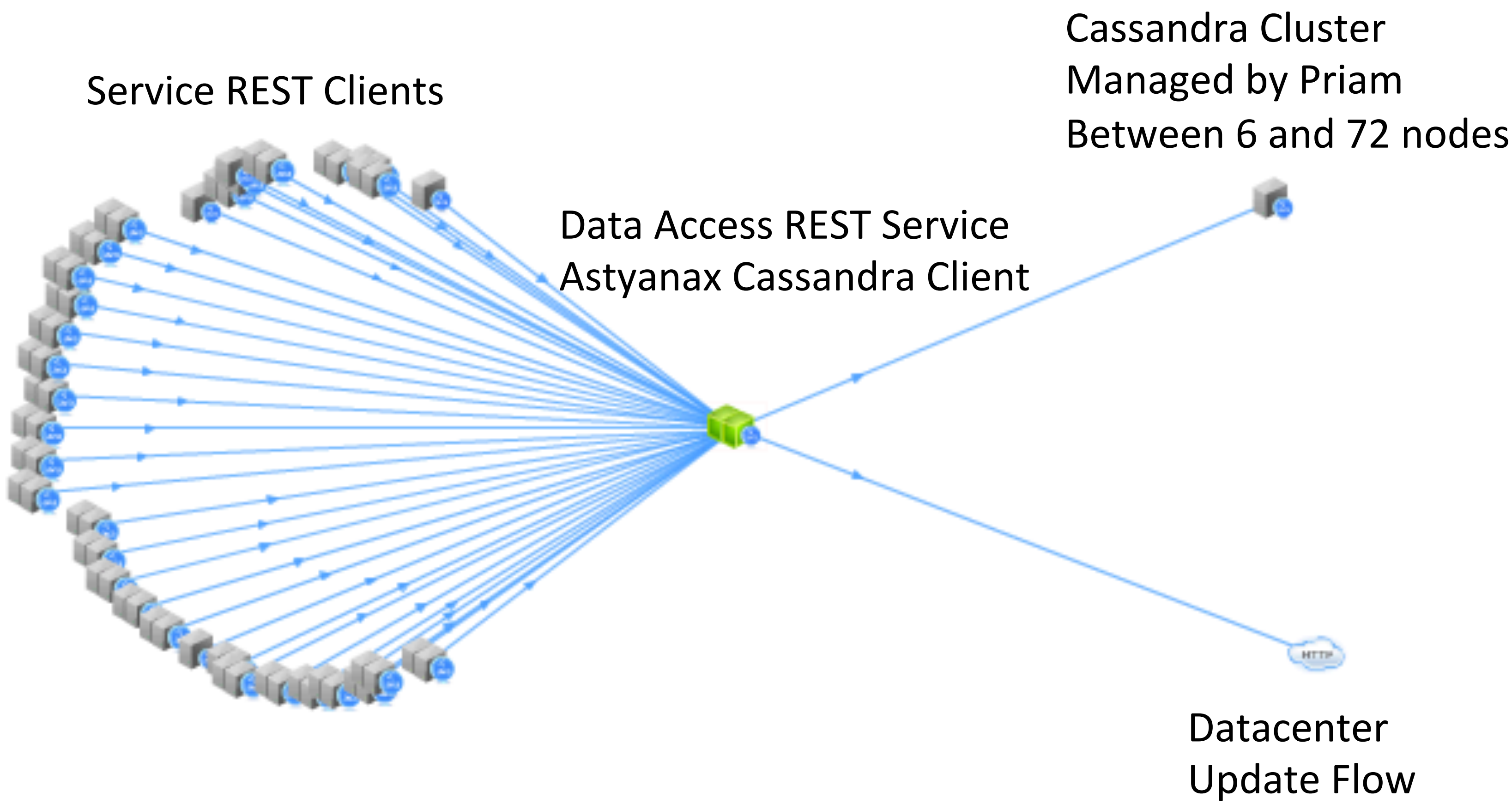
2009	2009	2010	2010	2010	2011
Content	**Logs**	**Play**	**WWW**	**API**	**CS**
Content Management	S3 Terabytes	DRM	Sign-Up	Metadata	International CS lookup
EC2 Encoding	EMR	CDN routing	Search	Device Config	Diagnostics & Actions
S3 Petabytes	Hive & Pig	Bookmarks	Movie Choosing	TV Movie Choosing	Customer Call Log
	Business Intelligence	Logging	Ratings	Social Facebook	CS Analytics
CDNs ISPs Terabits Customers	hadoop	Cassandra	MySQL	Cassandra	Cassandra

Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Cassandra on AWS

A highly available and durable deployment pattern

Cassandra Service Pattern



Appdynamics Service Flow Visualization

Production Deployment

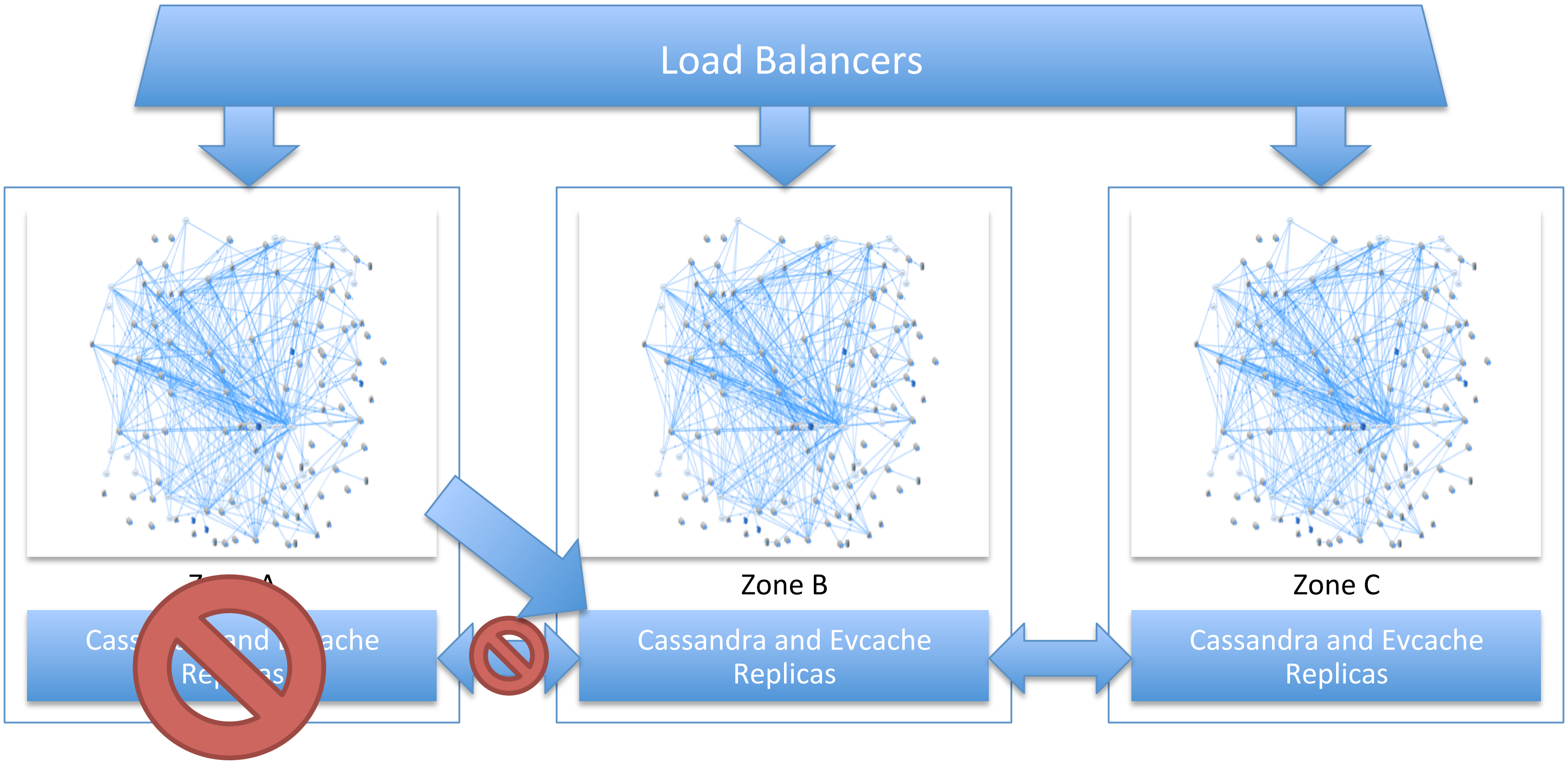
- Over 50 Cassandra Clusters
- Over 500 nodes
- Over 30TB of daily backups
- Biggest cluster 72 nodes
- 1 cluster over 250Kwrites/s

High Availability

- Cassandra stores 3 local copies, 1 per zone
  - Synchronous access, durable, highly available
  - Read/Write One fastest, use for fire and forget
  - Read/Write Quorum 2 of 3, use for read-after-write
- AWS Availability Zones
  - Separate buildings
  - Separate power etc.
  - Fairly close together

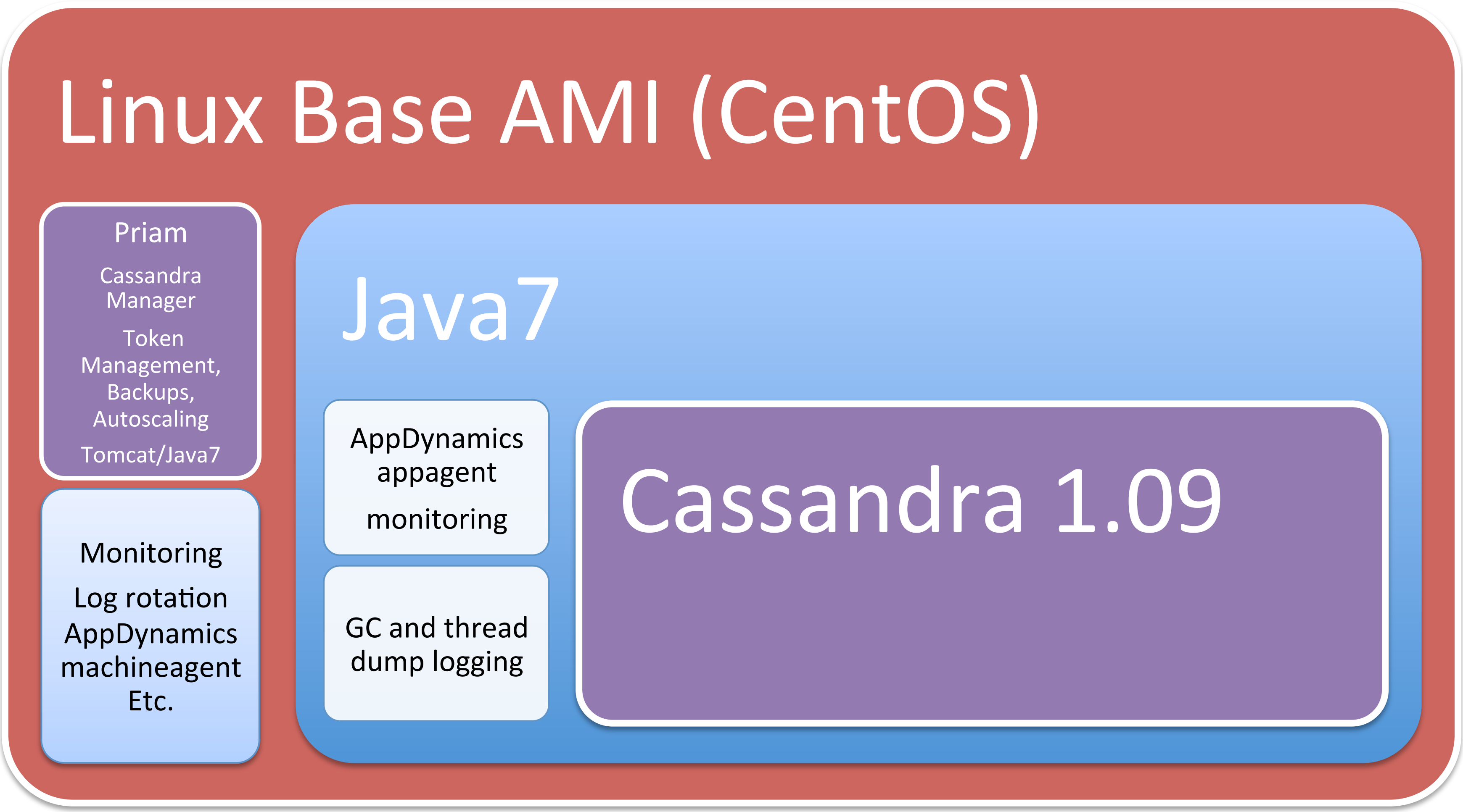
Triple Replicated Persistence

Cassandra maintenance drops individual replicas



Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Cassandra Instance Architecture



Priam – Cassandra Automation

Available at http://github.com/netflix

- Netflix Platform Tomcat Code
- Zero touch auto-configuration
- State management for Cassandra JVM
- Token allocation and assignment
- Broken node auto-replacement
- Full and incremental backup to S3
- Restore sequencing from S3
- Grow/Shrink Cassandra “ring”

Astyanax

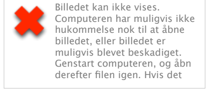
Available at http://github.com/netflix

- Features
  - Complete abstraction of connection pool from RPC protocol
  - Fluent Style API
  - Operation retry with backoff
  - Token aware
- Recipes
  - Distributed row lock (without zookeeper)
  - Multi-DC row lock
  - Uniqueness constraint
  - Multi-row uniqueness constraint
  - Chunked and multi-threaded large file storage

Astyanax Query Example

Paginate through all columns in a row

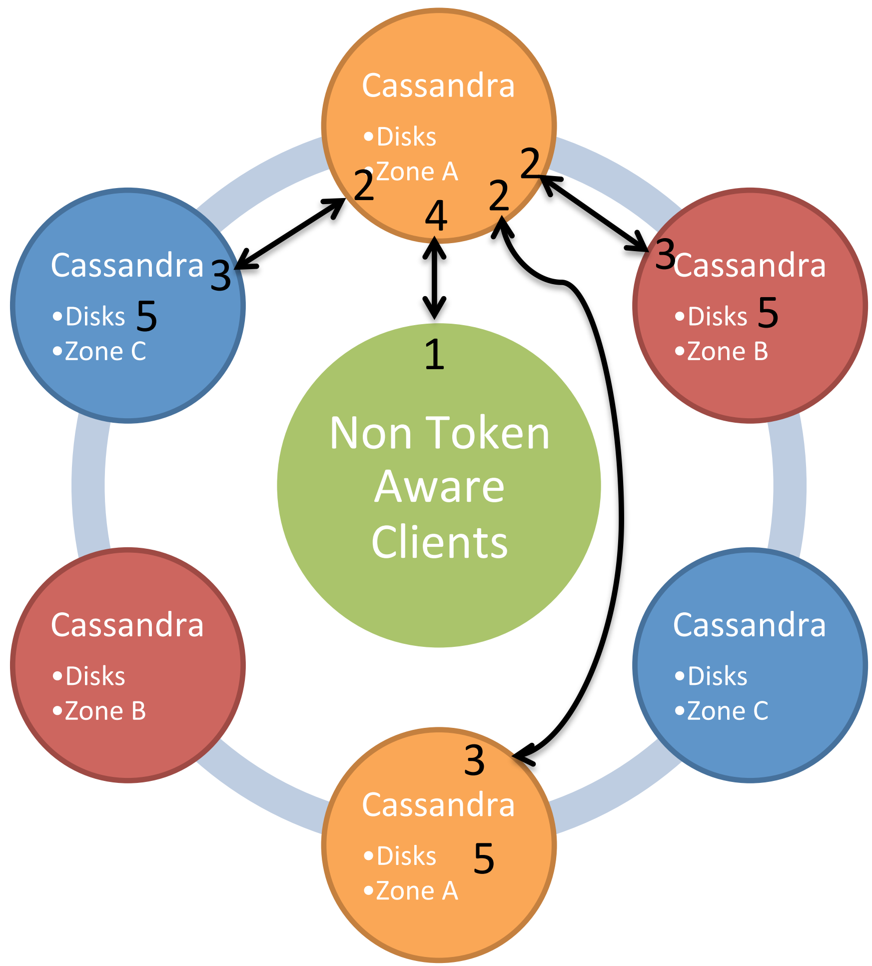
```
ColumnList<String> columns;
int pageize = 10;
try {
   RowQuery<String, String> query = keyspace
      .prepareQuery(CF_STANDARD1)
      .getKey("A")
      .setIsPaginating()
      .withColumnRange(new RangeBuilder().setMaxSize(pageize).build());

   while (!(columns = query.execute().getResult()).isEmpty()) {
      for (Column<String> c : columns) {
      }
   }
} catch (ConnectionException e) {
}
```



"Traditional" Cassandra Write Data Flows

Single Region, Multiple Availability Zone, Not Token Aware

1. Client Writes to any Cassandra Node
2. Coordinator Node replicates to nodes and Zones
3. Nodes return ack to coordinator
4. Coordinator returns ack to client
5. Data written to internal commit log disk (no more than 10 seconds later)



If a node goes offline, hinted handoff completes the write when the node comes back up.

Requests can choose to wait for one node, a quorum, or all nodes to ack the write

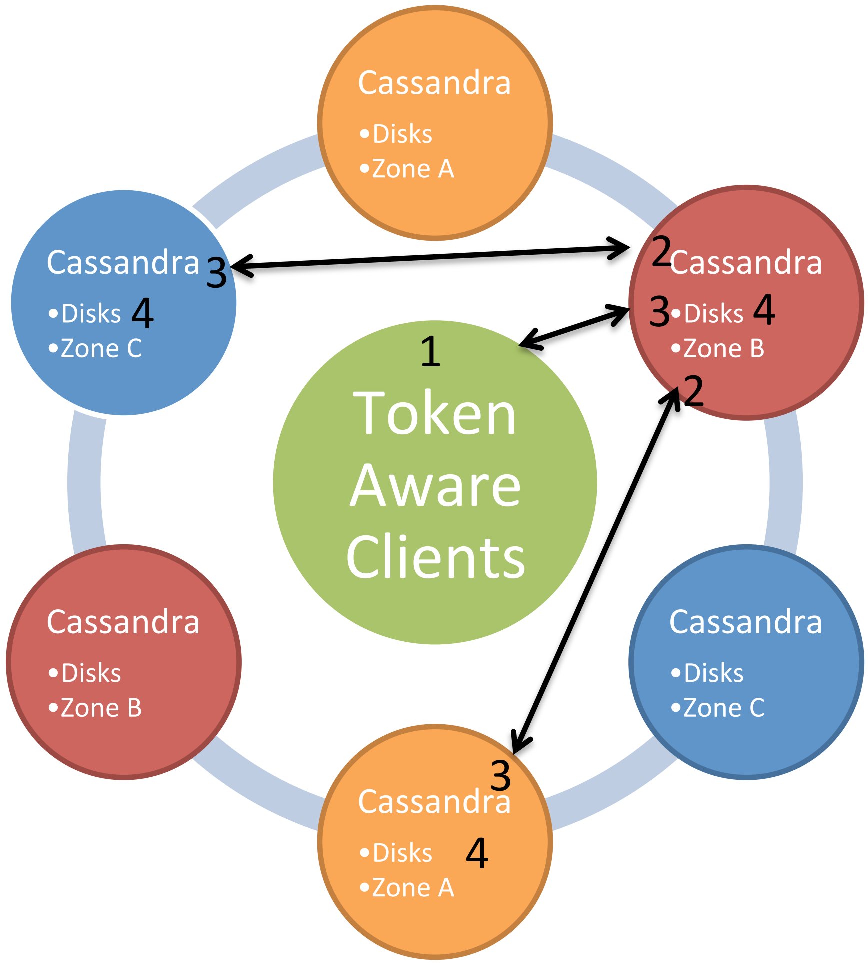
SSTable disk writes and compactions occur asynchronously

Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Astyanax - Cassandra Write Data Flows

Single Region, Multiple Availability Zone, Token Aware

1. Client Writes to local coordinator
2. Coodinator writes to other zones
3. Nodes return ack
4. Data written to internal commit log disks (no more than 10 seconds later)



If a node goes offline, hinted handoff completes the write when the node comes back up.

Requests can choose to wait for one node, a quorum, or all nodes to ack the write

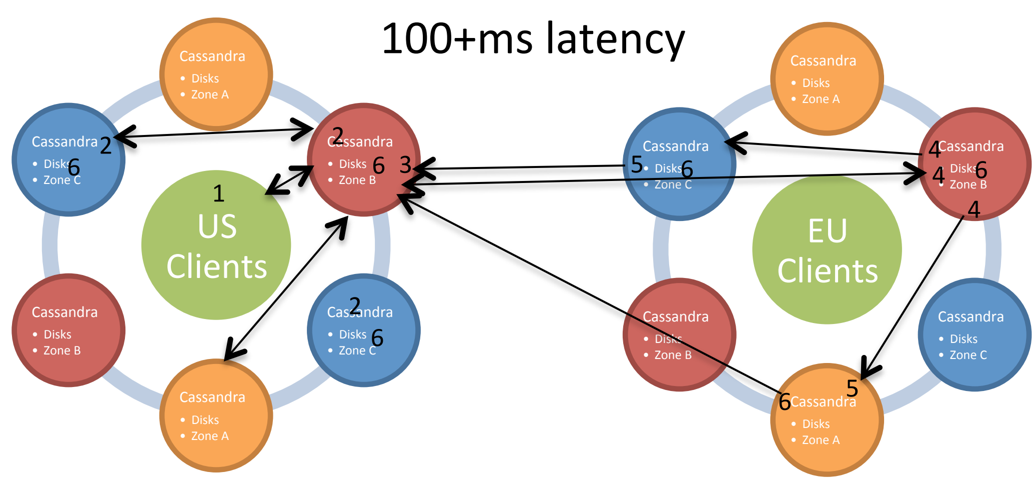
SSTable disk writes and compactions occur asynchronously

Data Flows for Multi-Region Writes

Token Aware, Consistency Level = Local Quorum

1. Client writes to local replicas
2. Local write acks returned to Client which continues when 2 of 3 local nodes are committed
3. Local coordinator writes to remote coordinator.
4. When data arrives, remote coordinator node acks and copies to other remote zones
5. Remote nodes ack to local coordinator
6. Data flushed to internal commit log disks (no more than 10 seconds later)

If a node or region goes offline, hinted handoff completes the write when the node comes back up. Nightly global compare and repair jobs ensure everything stays consistent.

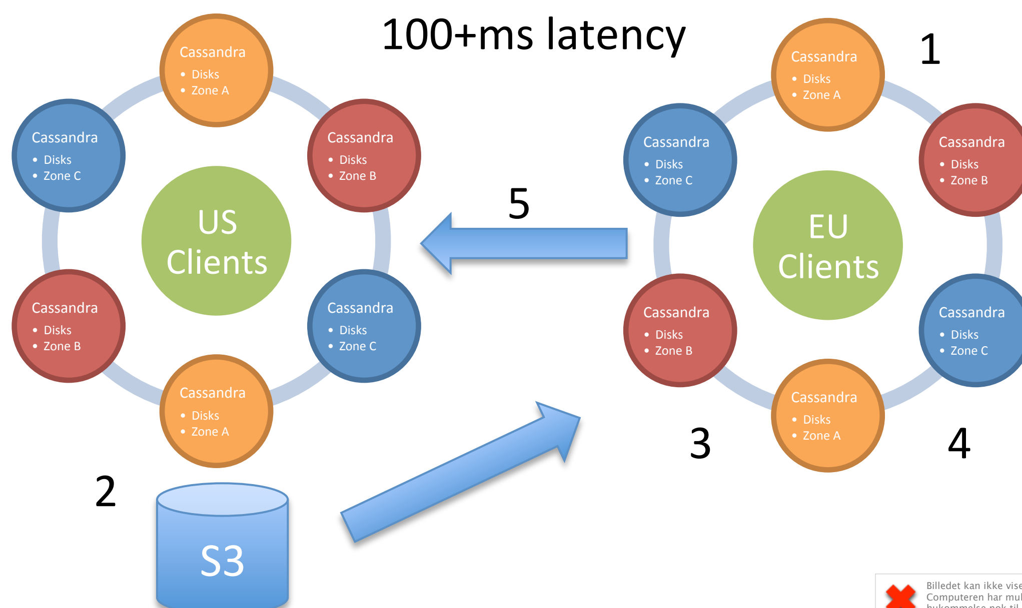


Extending to Multi-Region

Added production UK/Ireland support with no downtime
Minimize impact on original cluster using bulk backup move

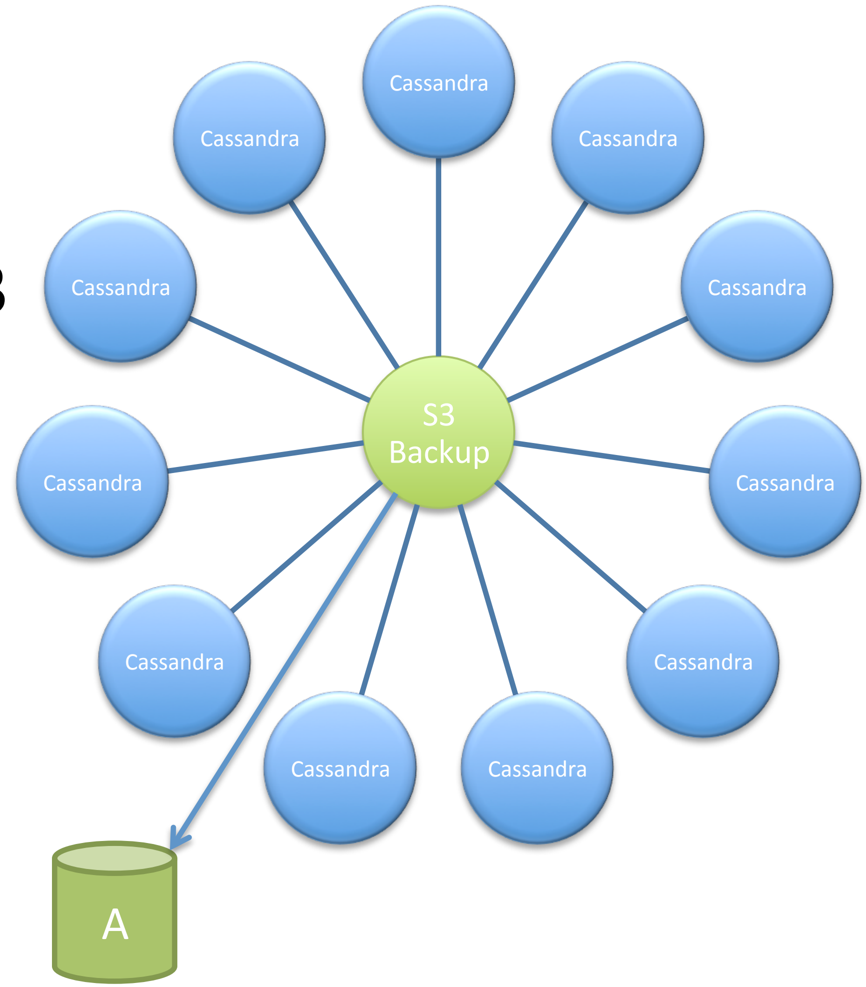
1. Create cluster in EU
2. Backup US cluster to S3
3. Restore backup in EU
4. Local repair EU cluster
5. Global repair/join

Take a Boeing 737 on a domestic flight, upgrade it to a 747 by adding more engines, fuel and bigger wings and fly it to Europe without landing it on the way…



Cassandra Backup

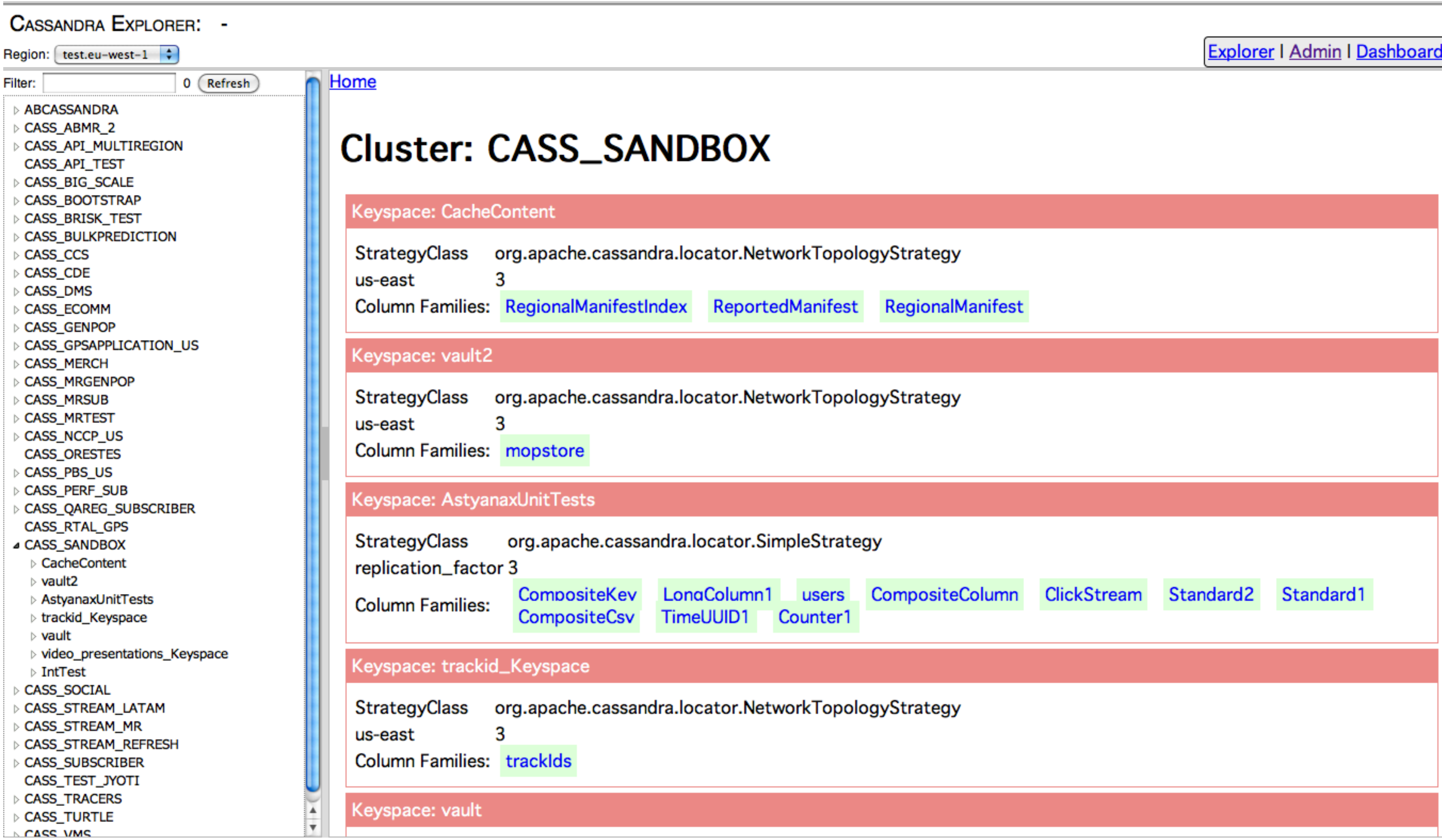
- Full Backup
  - Time based snapshot
  - SSTable compress -> S3
- Incremental
  - SSTable write triggers compressed copy to S3
- Archive
  - Copy cross region



Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Cassandra Explorer

Open source on github soon



ETL for Cassandra

- Data is de-normalized over many clusters!
- Too many to restore from backups for ETL
- Solution – read backup files using Hadoop
- Aegisthus
  - http://techblog.netflix.com/2012/02/aegisthus-bulk-data-pipeline-out-of.html
  - High throughput raw SSTable processing
  - Re-normalizes many clusters to a consistent view
  - Extract, Transform, then Load into Teradata

Asgard

http://techblog.netflix.com/2012/06/asgard-web-based-cloud-management-and.html

- Replacement for AWS Console at Scale
  - Groovy/Grails/JVM based
  - Supports all AWS regions on a global basis
  - Specific to AWS feature set
- Hides the AWS credentials
  - Use AWS IAM to issue restricted keys for Asgard
  - Each Asgard instance manages one account
  - One install each for test, prod, audit accounts

Open Source Projects

Legend

- Github / Techblog
- Apache Contributions
- Techblog Post
- Coming Soon

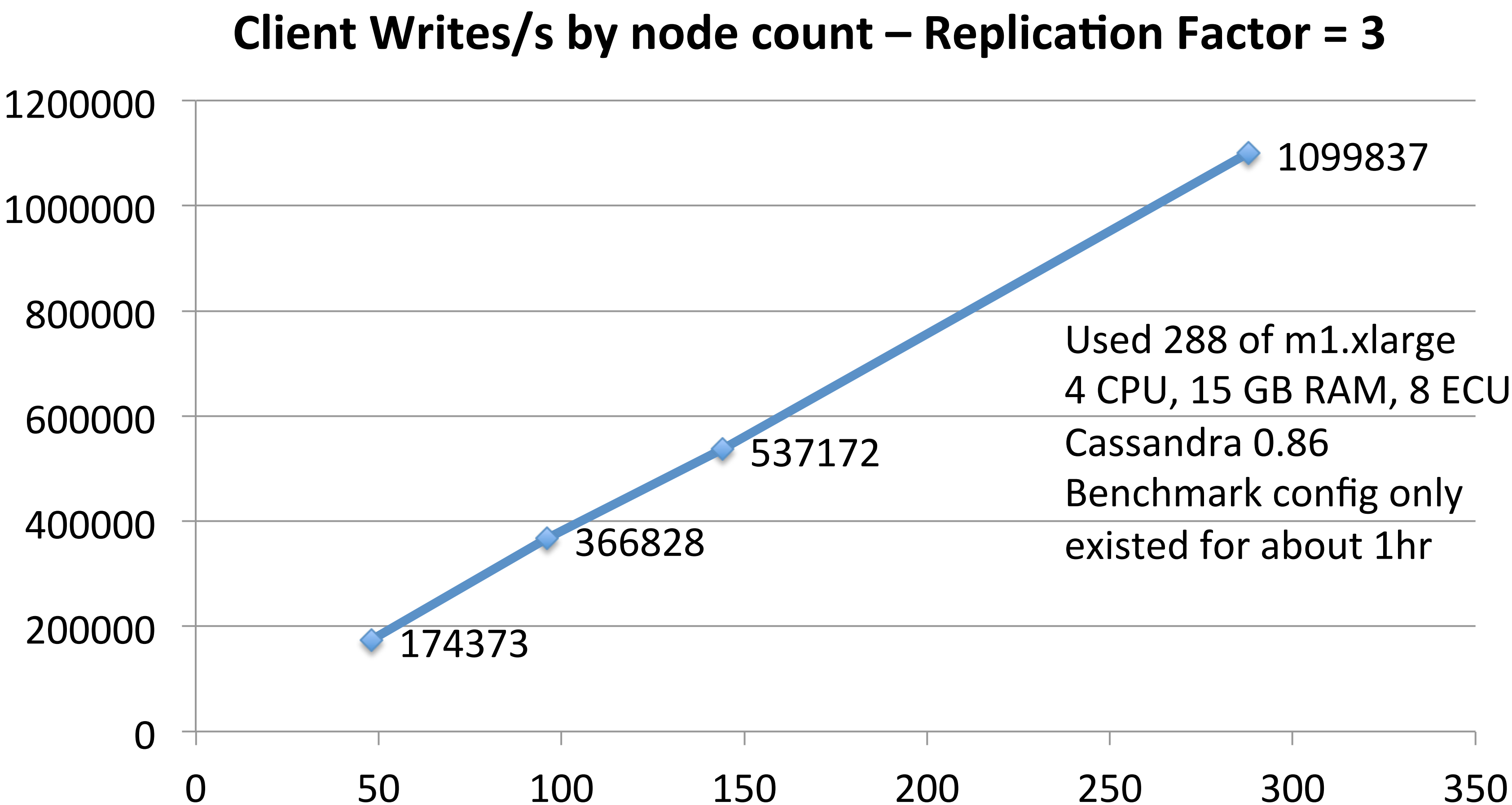
Priam Cassandra as a Service	Exhibitor Zookeeper as a Service	Servo and Autoscaling Scripts
Astyanax Cassandra client for Java	Curator Zookeeper Patterns	Honu Log4j streaming to Hadoop
CassJMeter Cassandra test suite	EVCache Memcached as a Service	Circuit Breaker Robust service pattern
Cassandra Multi-region EC2 datastore support	Eureka / Discovery Service Directory	Asgard - AutoScaleGroup based AWS console
Aegisthus Hadoop ETL for Cassandra	Archaius Dynamics Properties Service	Chaos Monkey Robustness verification
Explorers	EntryPoints	Latency Monkey
Governator - Library lifecycle and dependency injection	Server-side latency/error injection	Janitor Monkey
Odin Workflow orchestration	REST Client + mid-tier LB	Bakeries and AMI
Async logging	Configuration REST endpoints	Build dynaslaves

Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Benchmarks and Scalability

Scalability from 48 to 288 nodes on AWS

http://techblog.netflix.com/2011/11/benchmarking-cassandra-scalability-on.html



Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

“Some people skate to the puck,
I skate to where the puck is going to be”
Wayne Gretzky

Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Cassandra on AWS

The Past

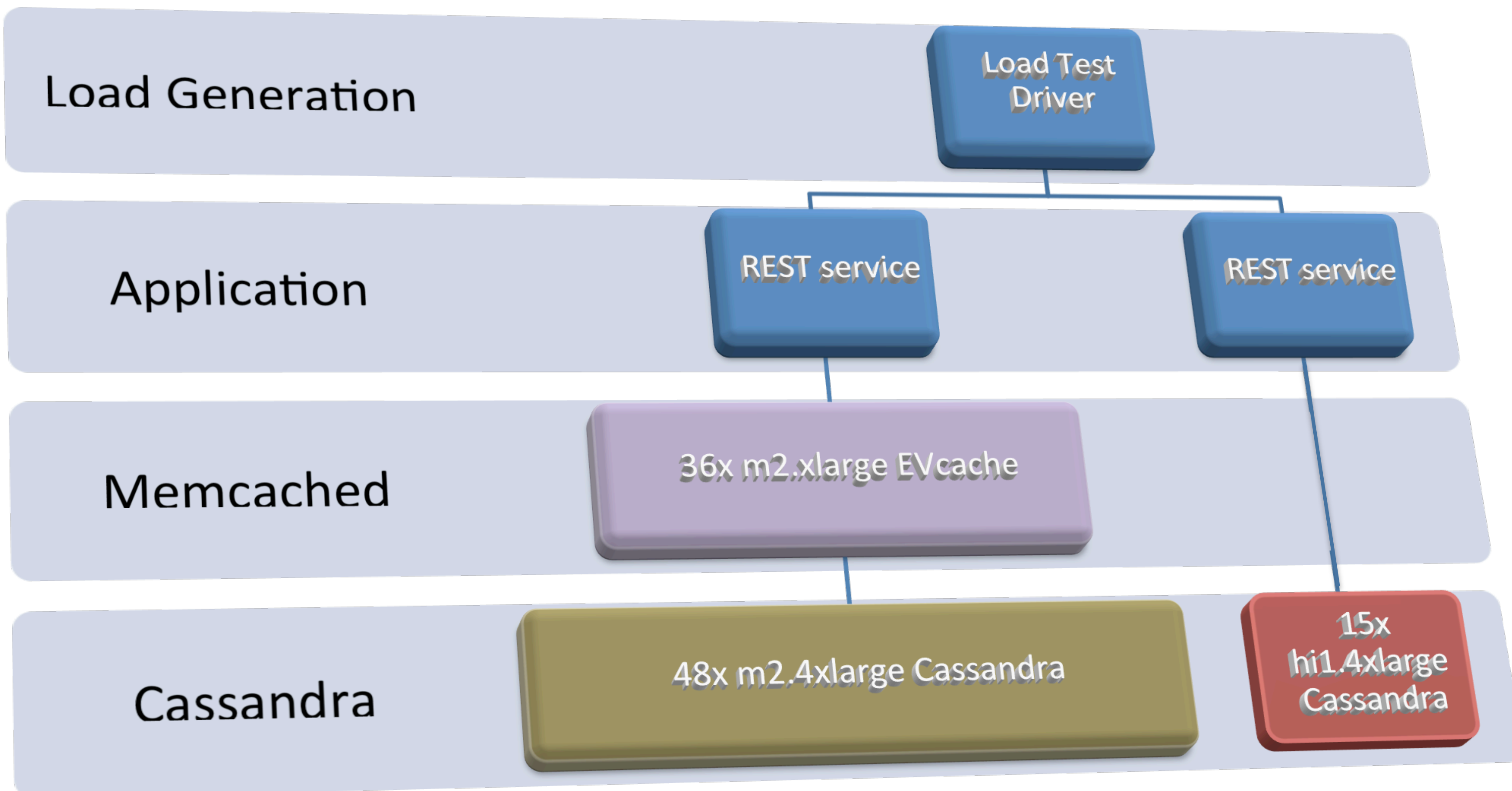
- Instance: m2.4xlarge
- Storage: 2 drives, 1.7TB
- CPU: 8 Cores, 26 ECU
- RAM: 68GB
- Network: 1Gbit
- IOPS: ~500
- Throughput: ~100Mbyte/s
- Cost: $1.80/hr

The Future

- Instance: hi1.4xlarge
- Storage: 2 SSD volumes, 2TB
- CPU: 8 HT cores, 35 ECU
- RAM: 64GB
- Network: 10Gbit
- IOPS: ~100,000
- Throughput: ~1Gbyte/s
- Cost: $3.10/hr

Cassandra Disk vs. SSD Benchmark

Same Throughput, Lower Latency, Half Cost



Get stuck with wrong config

Wait

Wait

File tickets

Ask permission

Wait

Wait

Things we don't do

Wait

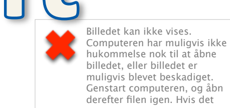
Wait

Run out of space/power

Plan capacity in advance

Wait

Have meetings with IT



Things we do do. Run benchmarks.
Live on stage.
(but this time it's a recording).


YOLO
www.itsokaytobesmart.com

Live Demo Workload

- Jenkins automation
  - Jmeter load driver
  - Asgard provisioning
  - Priam instance management
- Traffic
  - Reading/writing whole 100 column rows
  - Randomly selected from 25M row keys
  - Run for 10minutes, then double ring size

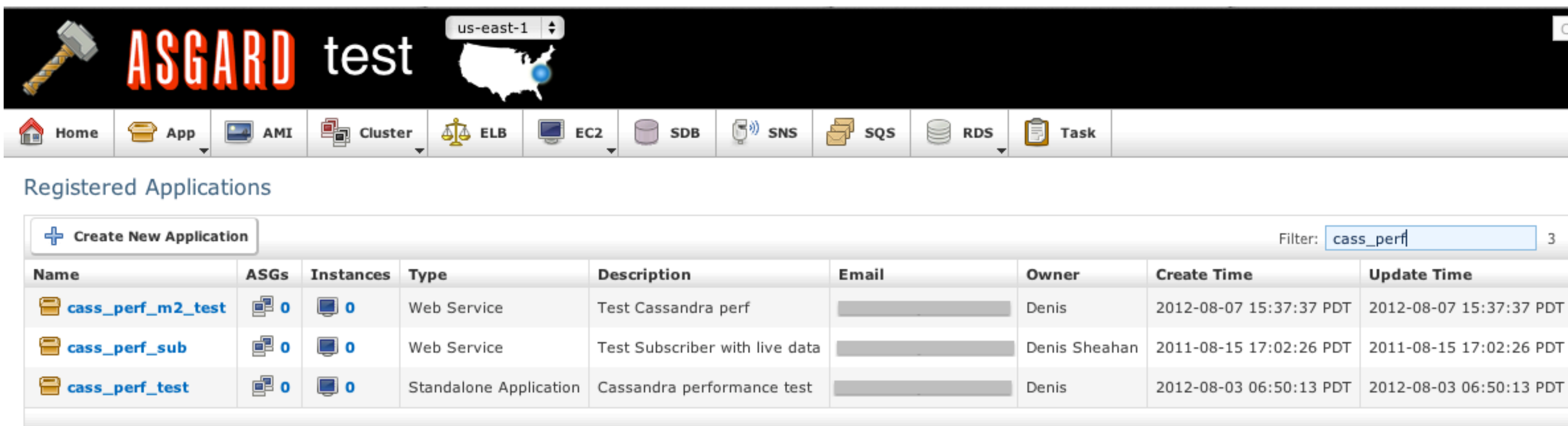
Screenshots from Live Demo

Cassandra Summit 2012 last July

Live on stage demo

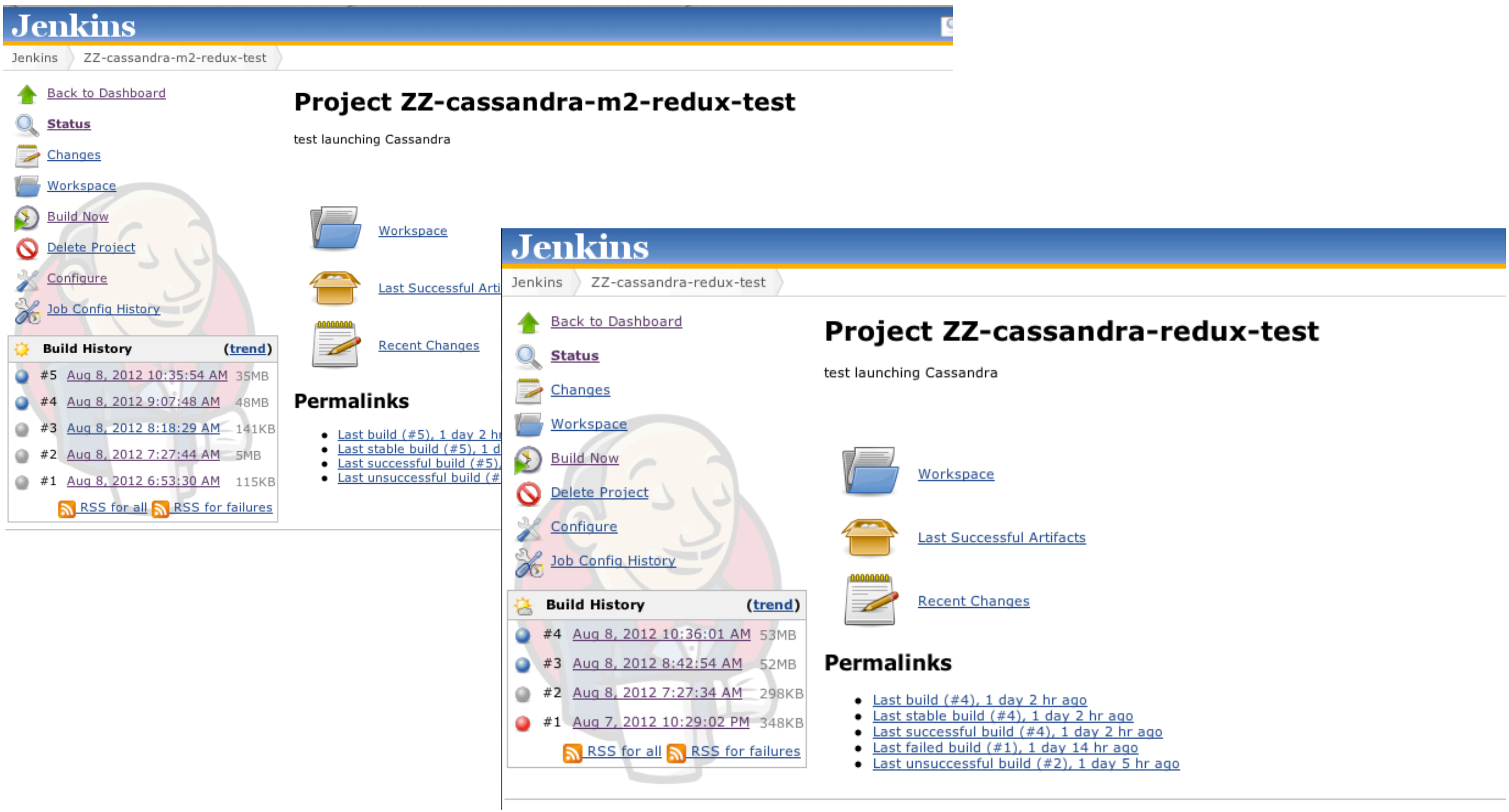
Asgard

cass_perf apps, with no instances running



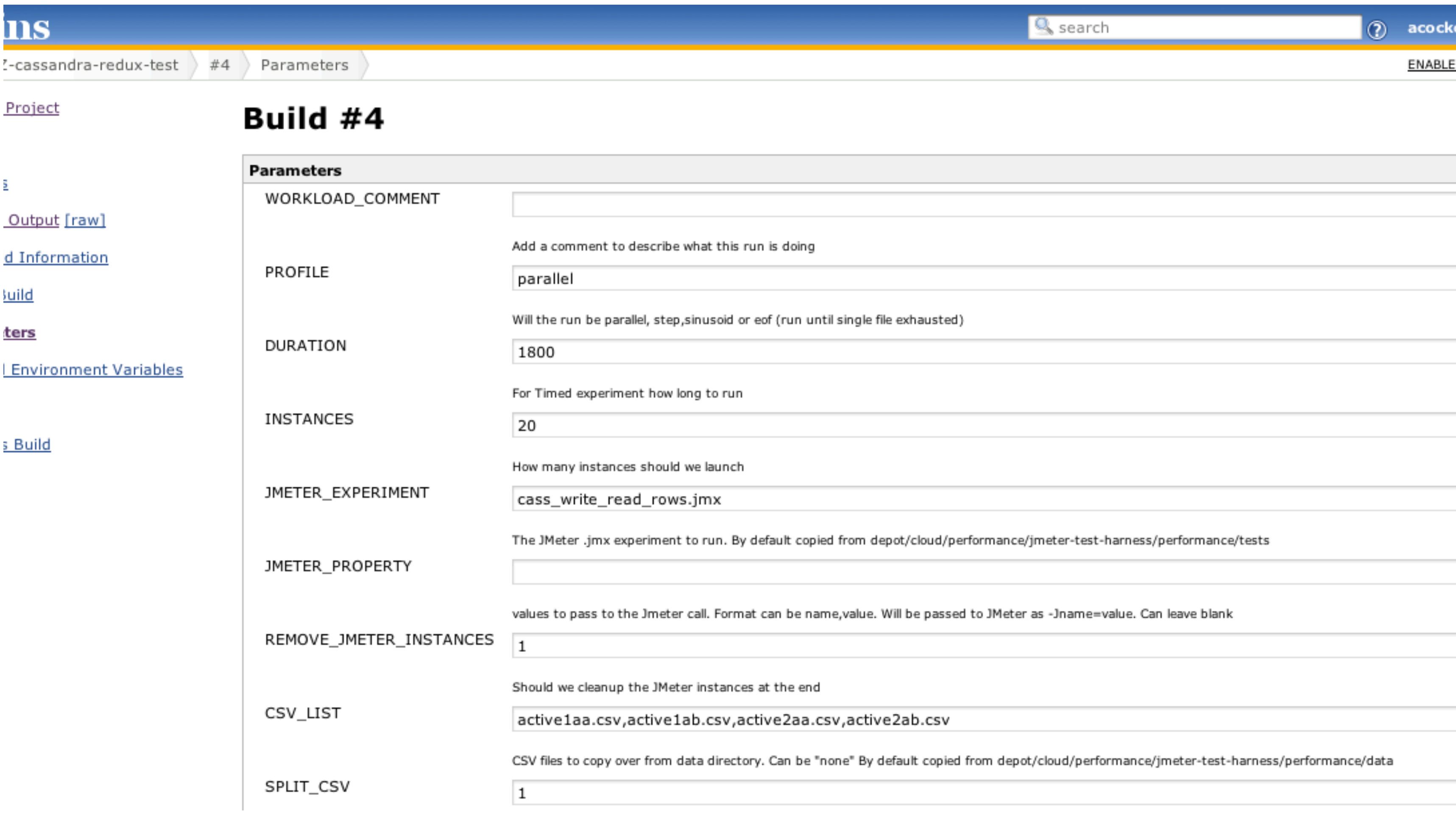
Jenkins

Jenkins perf_test jobs



Jmeter Setup

Build parameters



Jmeter Setup

Build parameters

:-cassandra-redux-test › #4 › Parameters ENABLE A

CSV files to copy over from data directory. Can be "none" By default copied from depot/cloud/performance/jmeter-test-harness/performance/data

SPLIT_CSV: 1

Split the CSV files evenly across the clients

THREAD_LIST: 20

List of thread counts for each experiment eg 10 20 30. Experiments will be run in order

LOOP_COUNT: timed

Type of experiment - "timed", "forever" or number_of_iterations

REGION: us-east-1

Region to run the test in

TARGET_ASG: cass_perf_test,0,11

The service ASG that we are targeting performance. Will create a servers.csv file with a list of instances in this service. Parameter is ,,

JMETER_SAMPLE_NAMES: Cassandra_Batch_Put,Cassandra_Get_Range_Slice

List of Samplers in the JMeter experiment, this list will be post processed and plotted

MERGE_DATA: 0

Should we merge the JMeter results from all the client machines

EPIC_CONFIG: cass_scale_list,cluster:cass_perf_test@--useast1e.awstest+cass_scale_list,cluster:cass_perf_test@--useast1c.awstest+cass

List of epics to dump at the end of the run parameter format list1,cluster:list2:cluster.... list format: output_file,title,counter1...counterN

NODESTATS: cass_perf_test,0,3

Collect stats on Target ASG. Parameter is , or none

COLLECT_PROFILE: none

Collect Studio profile on node, Parameter is , or none

Jmeter Setup

Build parameters

Z-cassandra-redux-test › #4 › Parameters   ENABLE A

Collect Studio profile on node, Parameter is , or none

Parameter	Value
COLLECT_HEAP_ALLOCATIONS	none

Collect Heap allocations on node with hprof. Parameter is , or none

Parameter	Value
CLEANUP_SCRIPT	none

script,parameter to reset the environment after the run. Example is an AB delete script

Parameter	Value
TARGET_DEPLOY	ami-69f85000 hi1.4xlarge 1 1 create_keyspace

How to deploy the target AMI instance-type

Parameter	Value
CASSANDRA	1
JMETER_INSTANCE_TYPE	m2.2xlarge

The size to use for JMeter

Parameter	Value
TARGET_TEARDOWN	1

Delete the target after test

Parameter	Value
PRESERVE_RUN	0

Should this run be preserved for a replay

Parameter	Value
PARALLEL_SCRIPT	800:cass_perf_test.create_double_ring

Kick this script of at the start of the load to run in parallel

Parameter	Value
DELTA	240

Offset subsequent thread groups

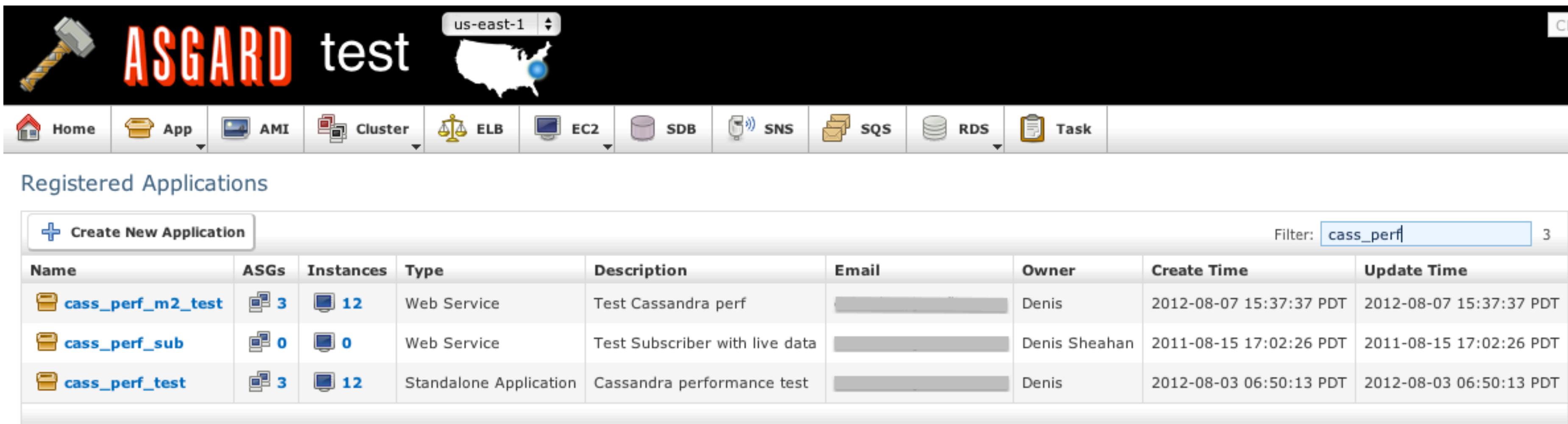
Parameter	Value
REPLAY_BUILD	jmeter_client-ZZ_cassandra_test-54

ocalize this page   Page generated: Aug 9, 2012 1:32:52 PM   Jenki

Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Asgard

Iinitial set of cass instances up and running



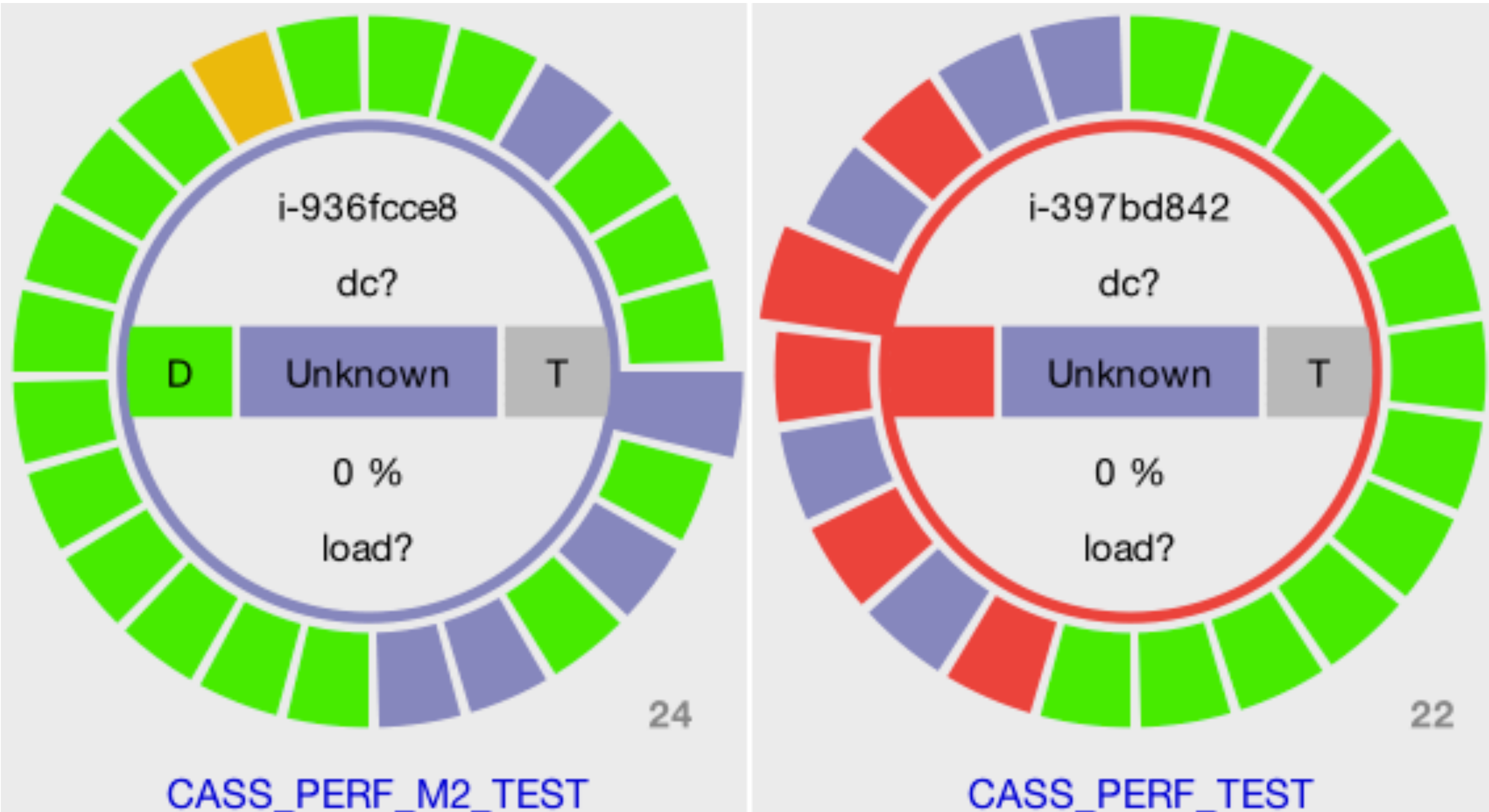
Name	ASGs	Instances	Type	Description	Email	Owner	Create Time	Update Time
cass_perf_m2_test	3	12	Web Service	Test Cassandra perf		Denis	2012-08-07 15:37:37 PDT	2012-08-07 15:37:37 PDT
cass_perf_sub	0	0	Web Service	Test Subscriber with live data		Denis Sheahan	2011-08-15 17:02:26 PDT	2011-08-15 17:02:26 PDT
cass_perf_test	3	12	Standalone Application	Cassandra performance test		Denis	2012-08-03 06:50:13 PDT	2012-08-03 06:50:13 PDT

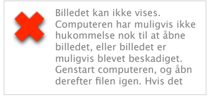
Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Kiklos (open source soon)

Clusters growing from 12 to 24

in-service, bootstrapping, garbage-collecting, cass-down

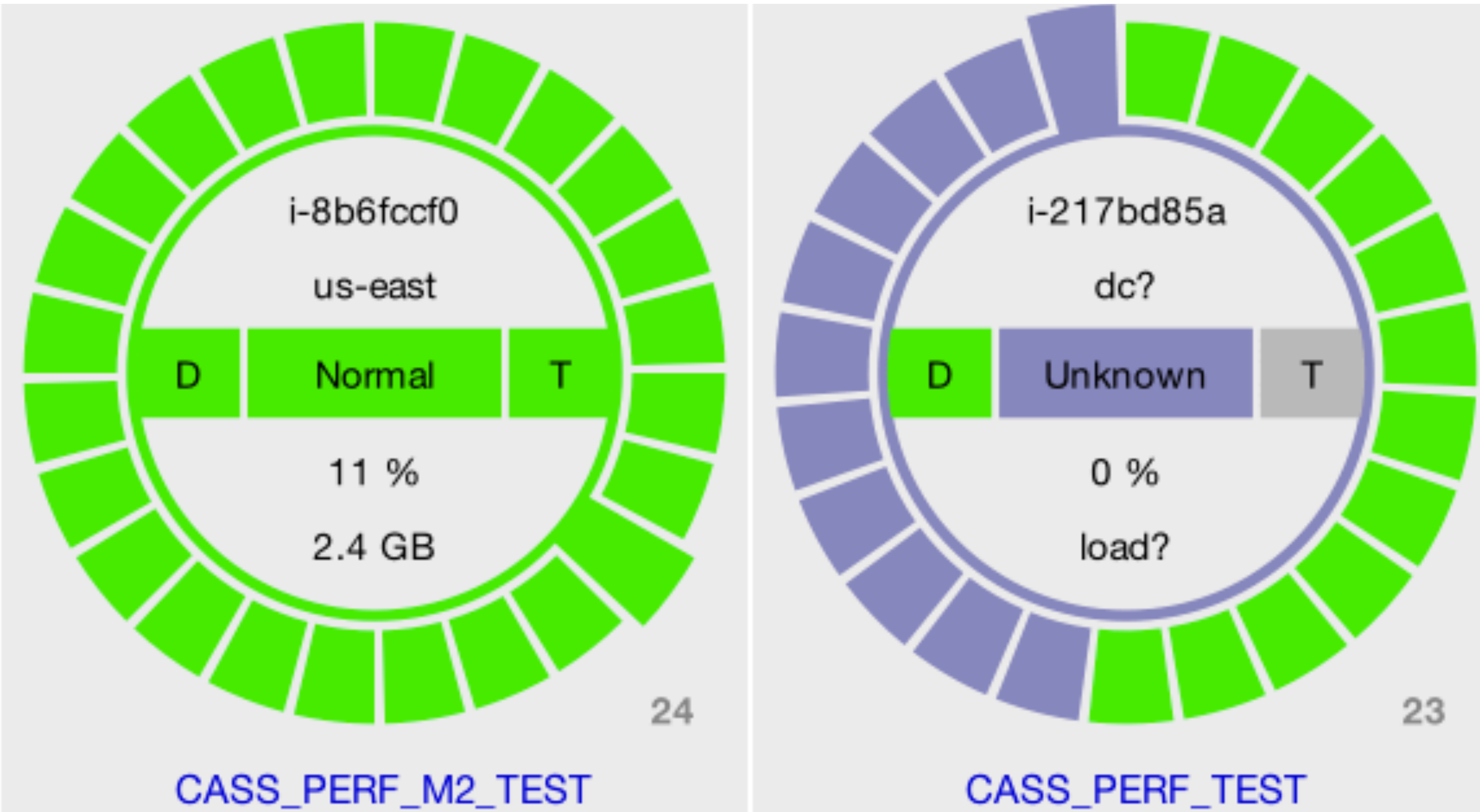




Kiklos

Clusters growing from 12 to 24

in-service, bootstrapping, garbage-collecting, cass-down

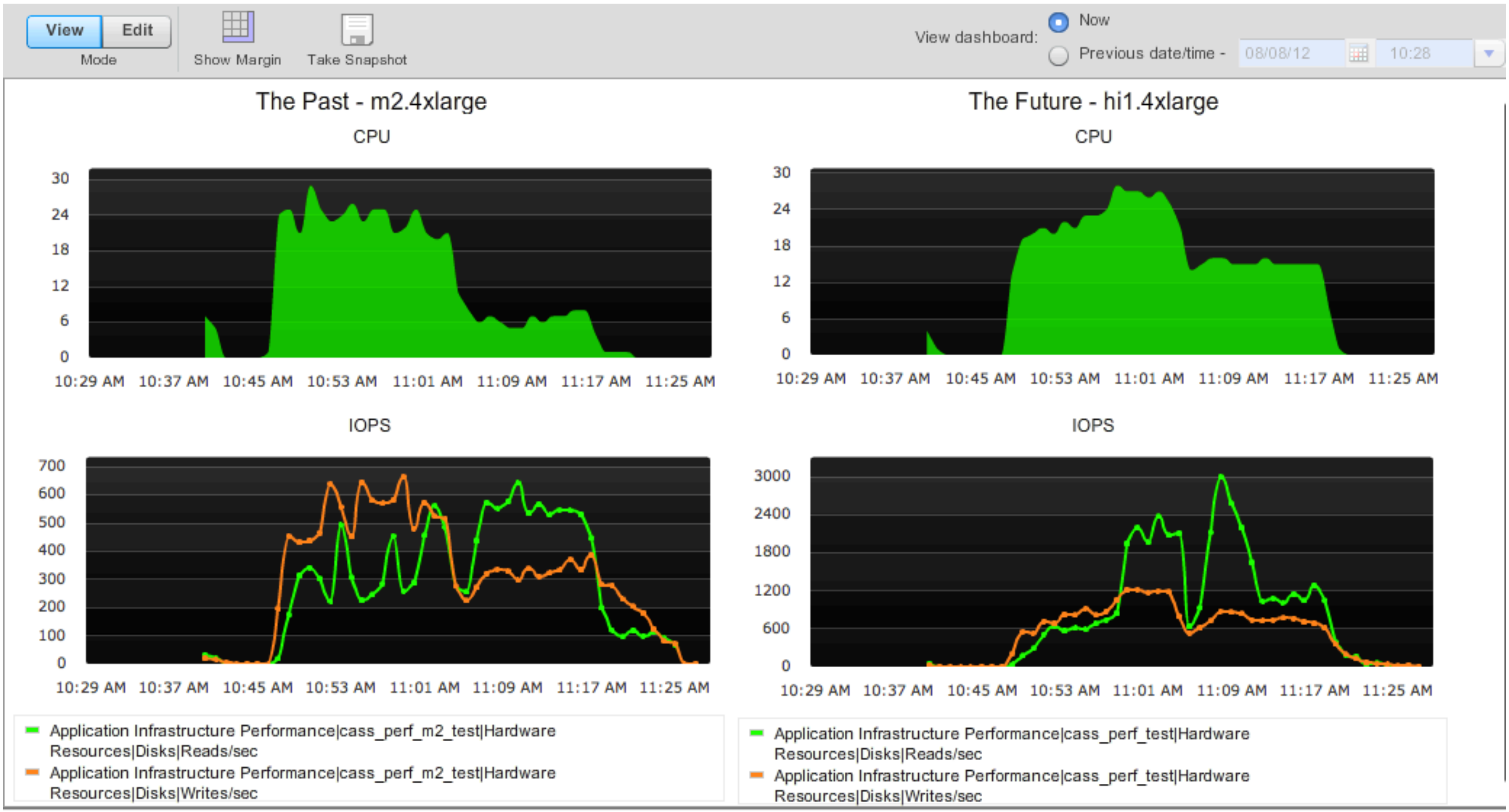


Disclaimers

- We didn't have time to tune the demo
- These are the plots from the live demo run
- Run's need to be longer to get to steady state
- Data size only reached around 5GB per node
- Plenty of "I wonder why it did that" remains
- It's a fair comparison, but not the best absolute performance possible for this workload and configuration
- When you remove the IO bottleneck, the next few bottlenecks appear…

Activity during the talk 10:30-11:30

Custom AppDynamics dashboard showing CPU and IOPS per node

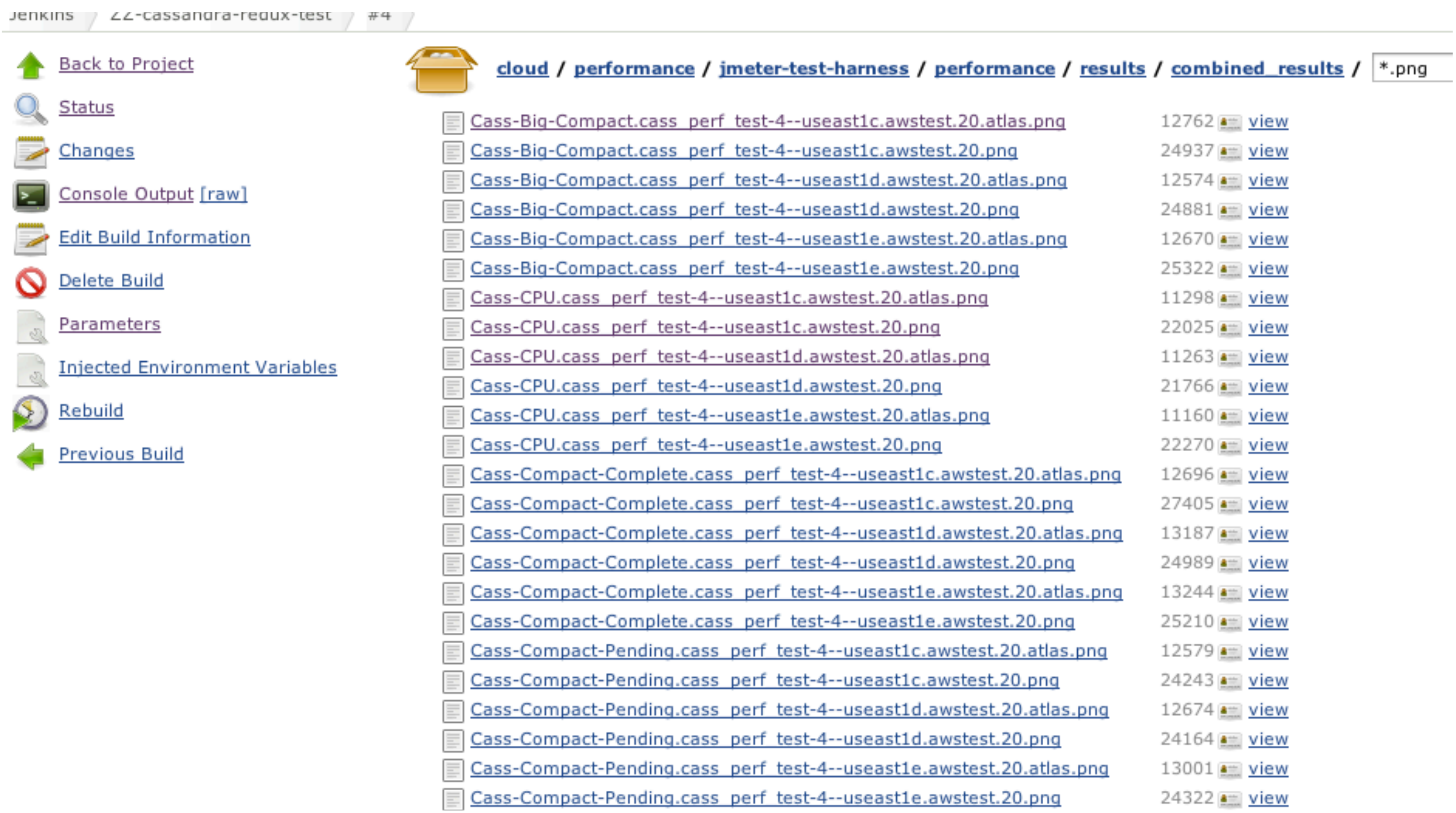


Jmeter Plots

- Plots are the output of the Jenkins build
- Each instance has its own set of plots
- Each availability zone has its own summary plots
- One of the three zone summary plots is compared for each metric
- Plot collection is currently duplicated as we are transitioning from “Epic” to “Atlas”

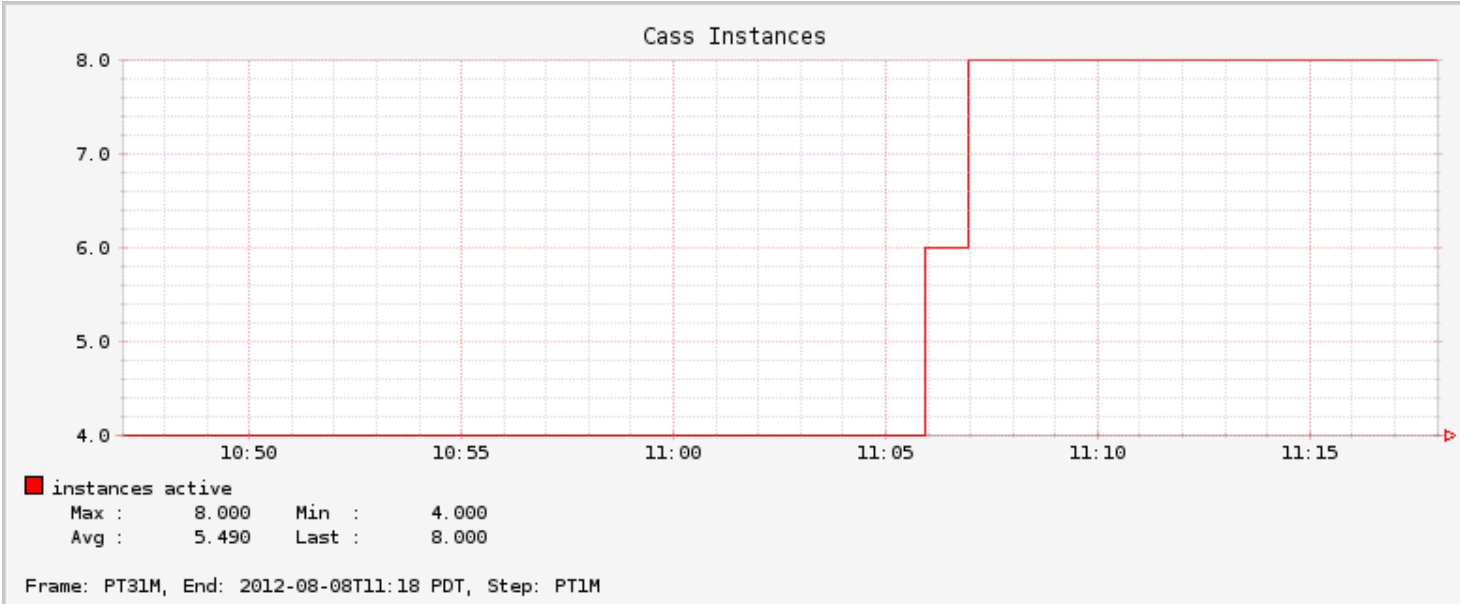
Jenkins

Collected results and graphs after job has completed

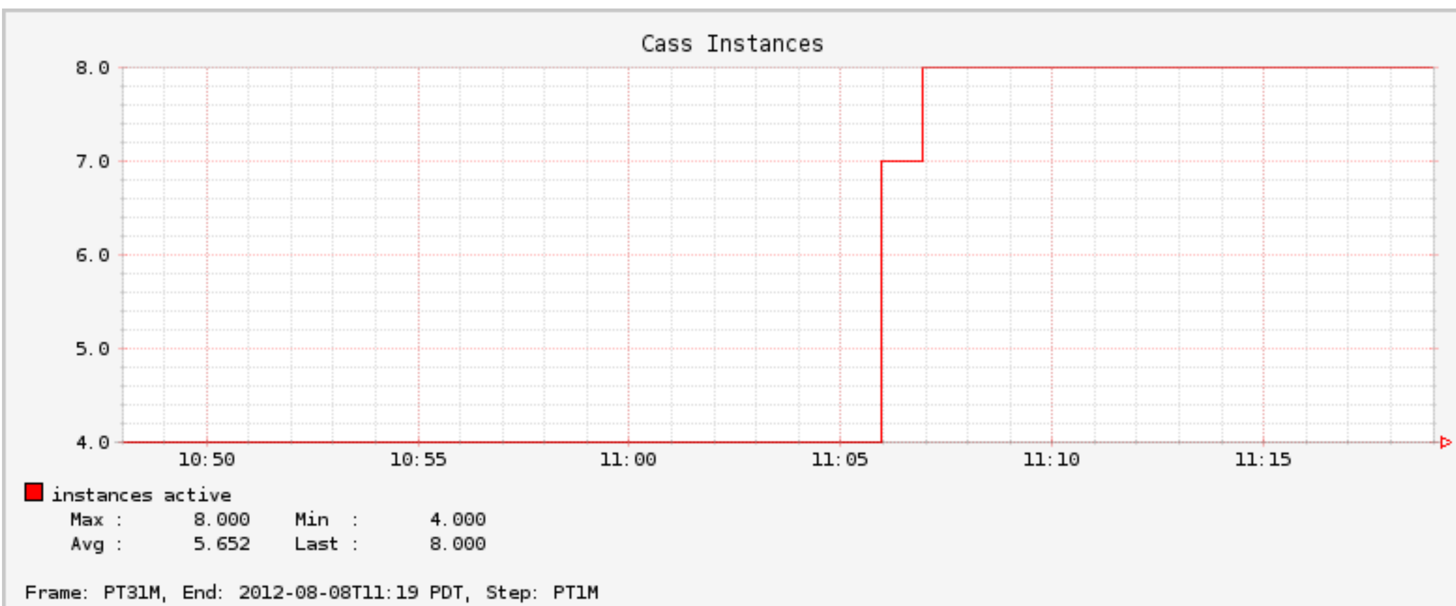


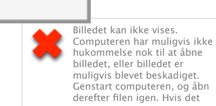
The past
m2.4xlarge

Instances
per zone



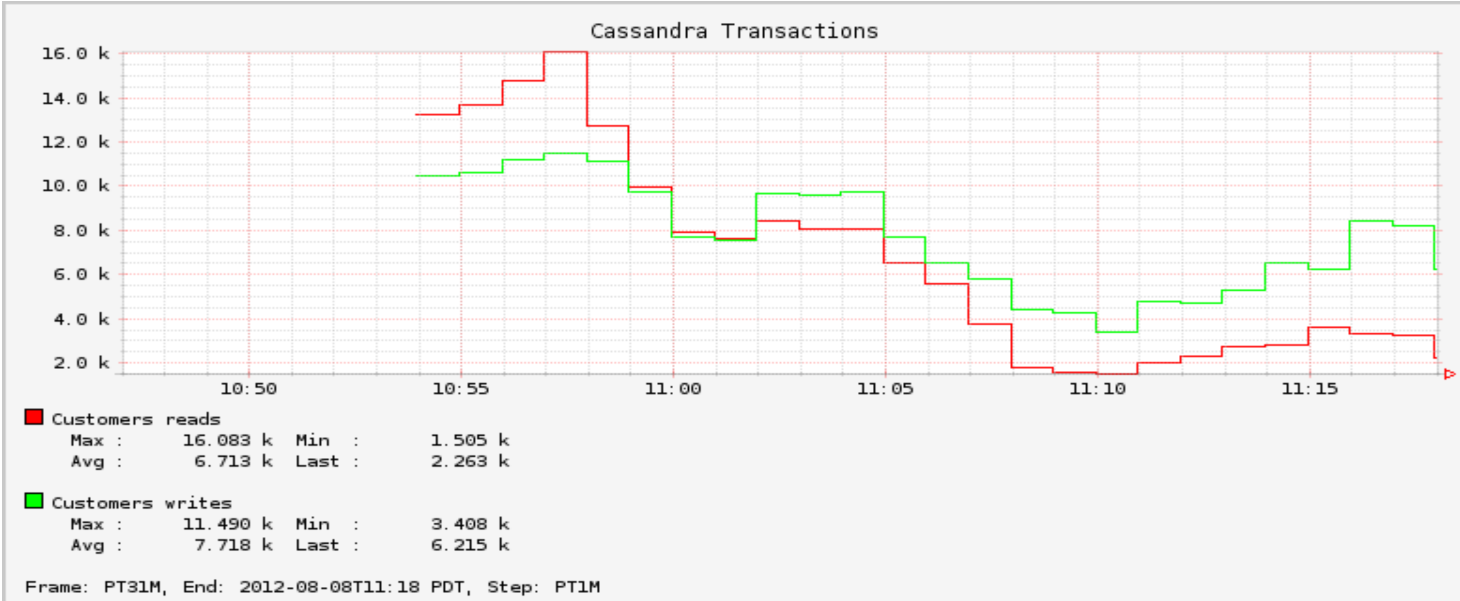
The future
hi1.4xlarge



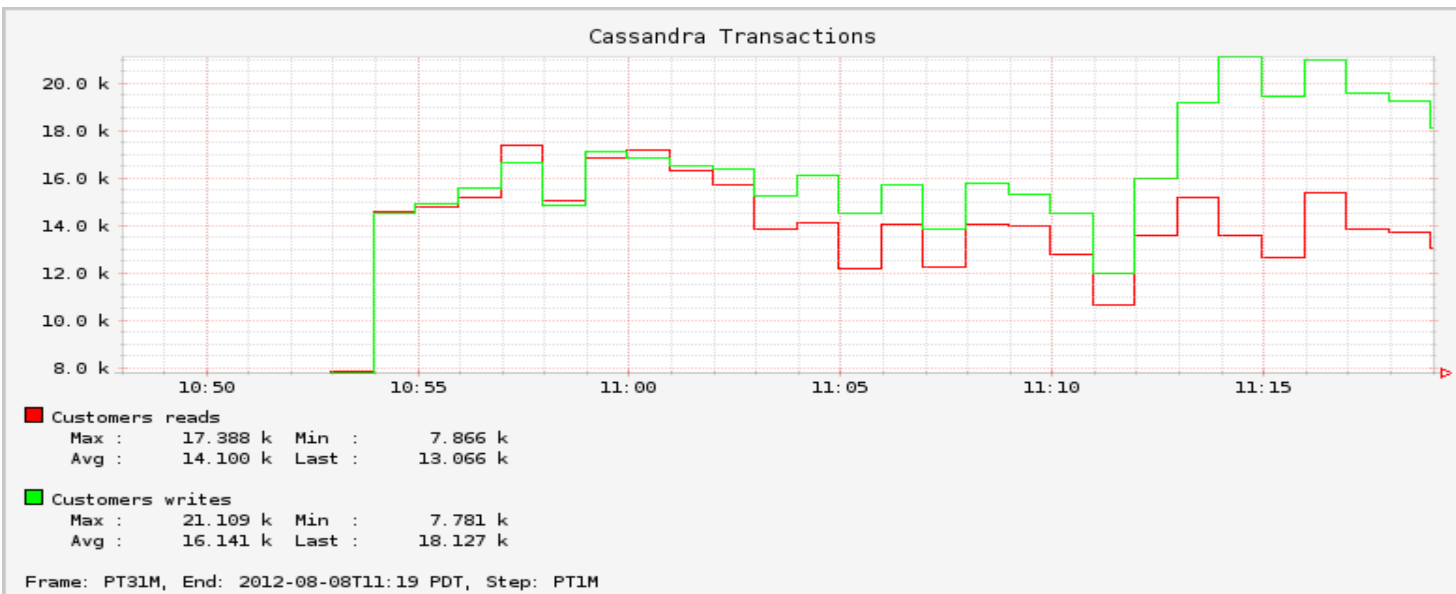


The past
m2.4xlarge

Transactions per zone, same as total client transactions

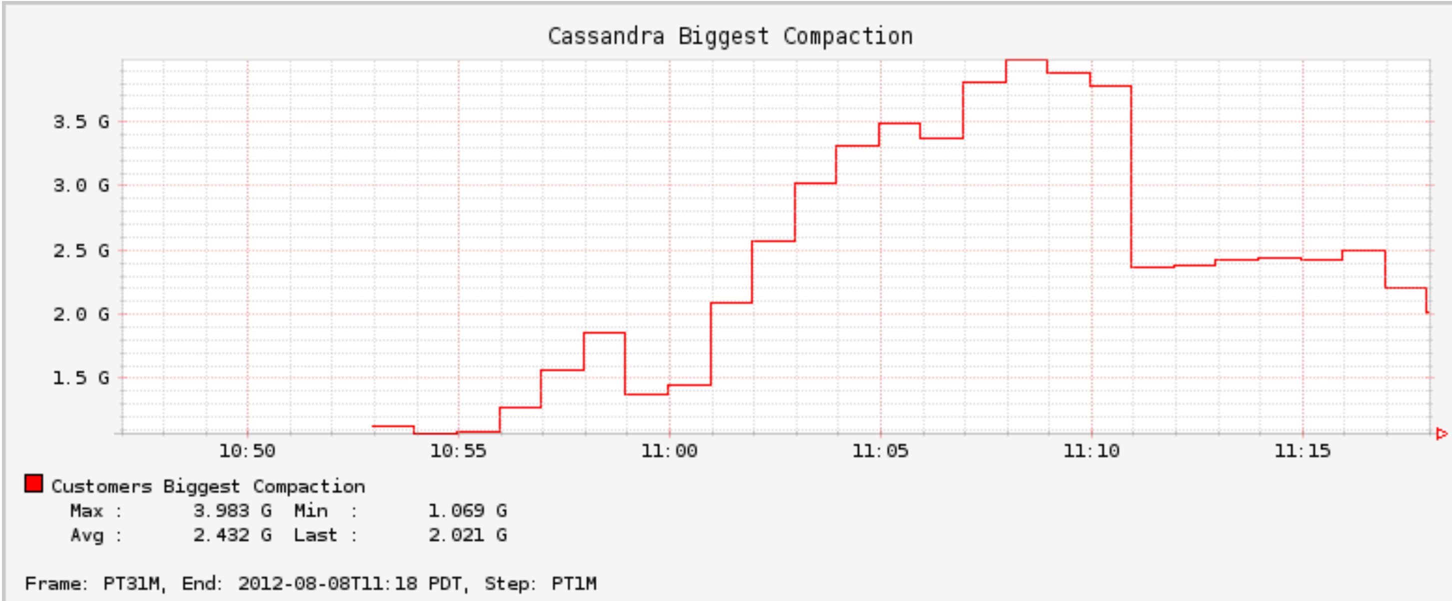


The future
hi1.4xlarge

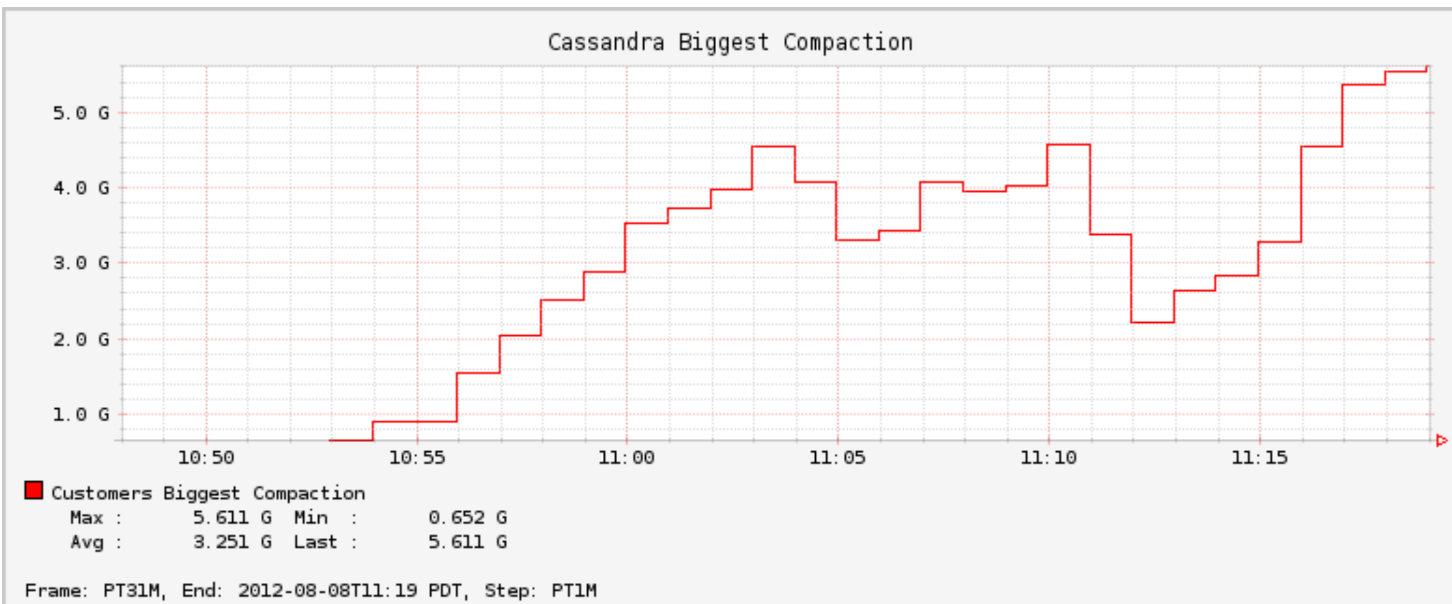


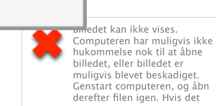
Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

The past
m2.4xlarge



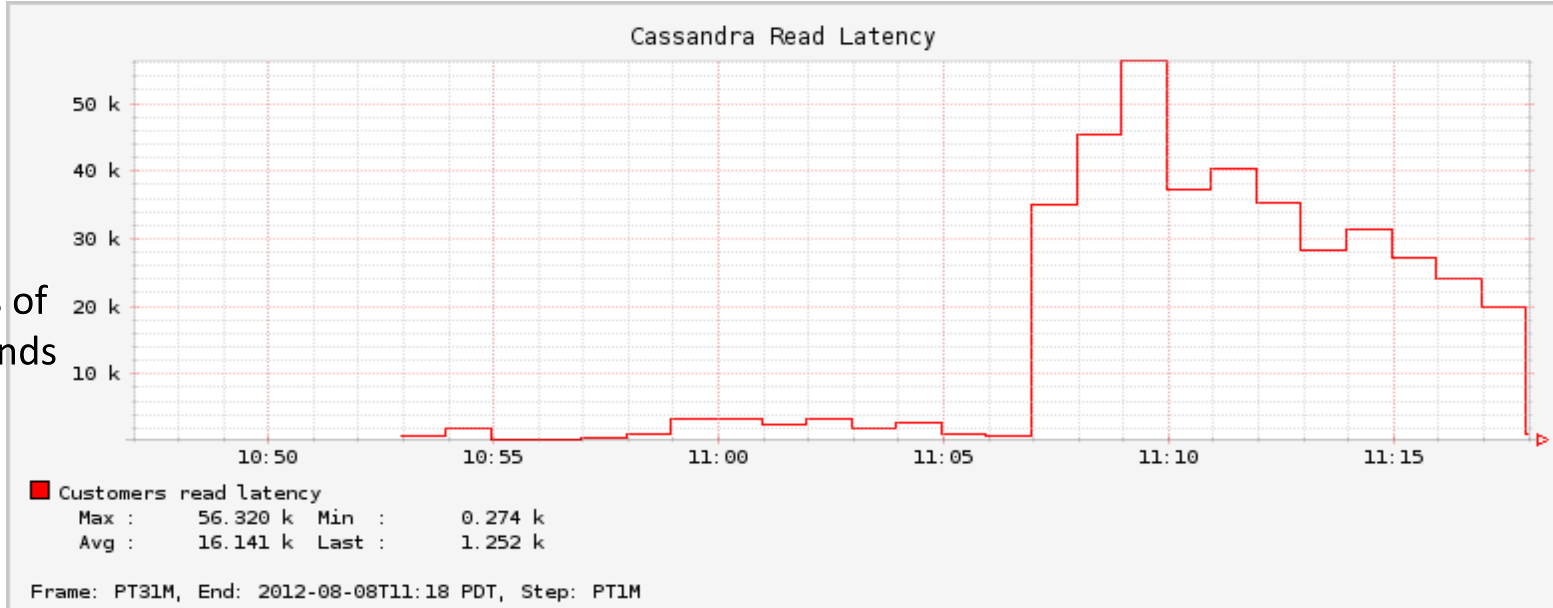
The future
hi1.4xlarge



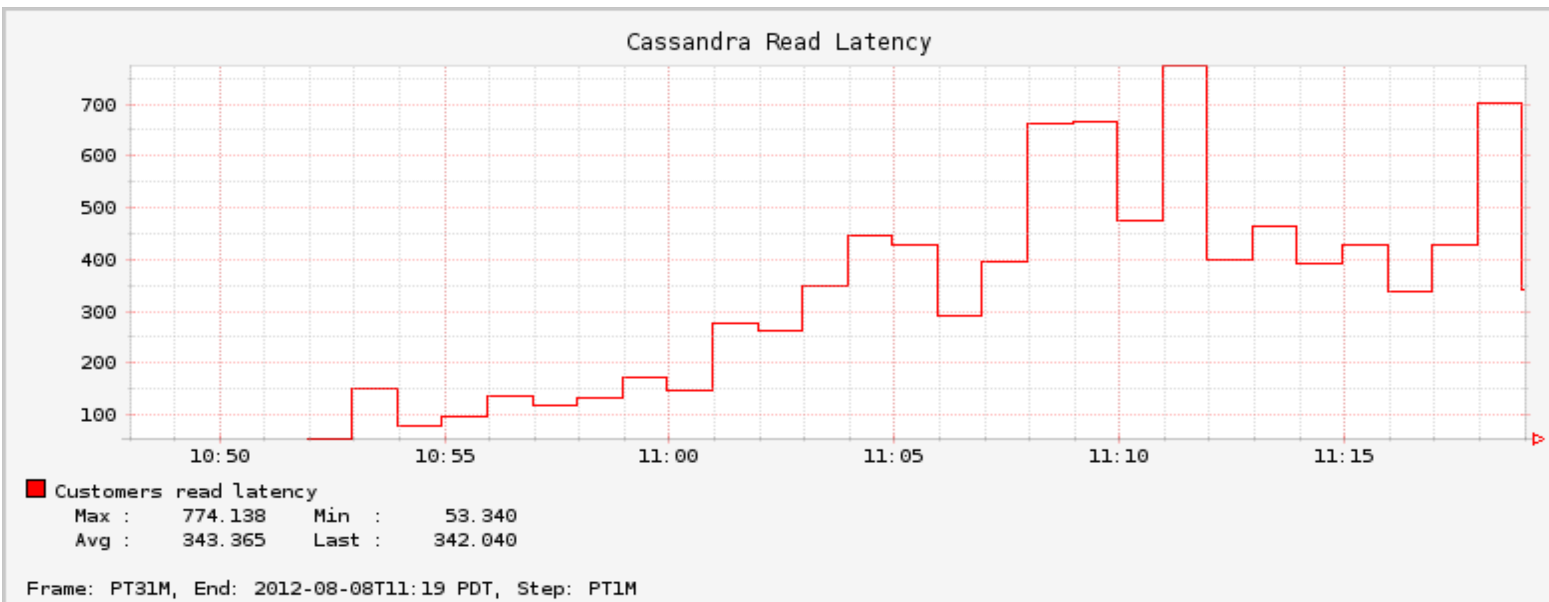


The past
m2.4xlarge

Thousands of
Microseconds

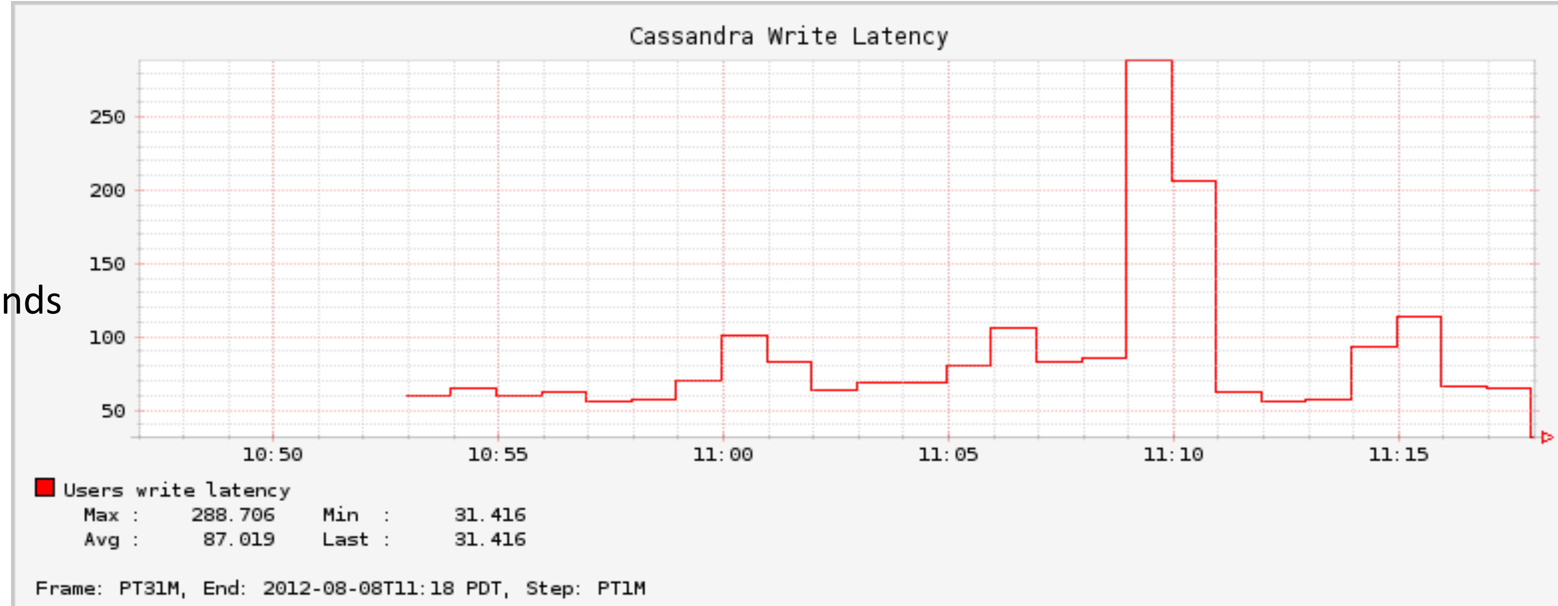


The future
hi1.4xlarge

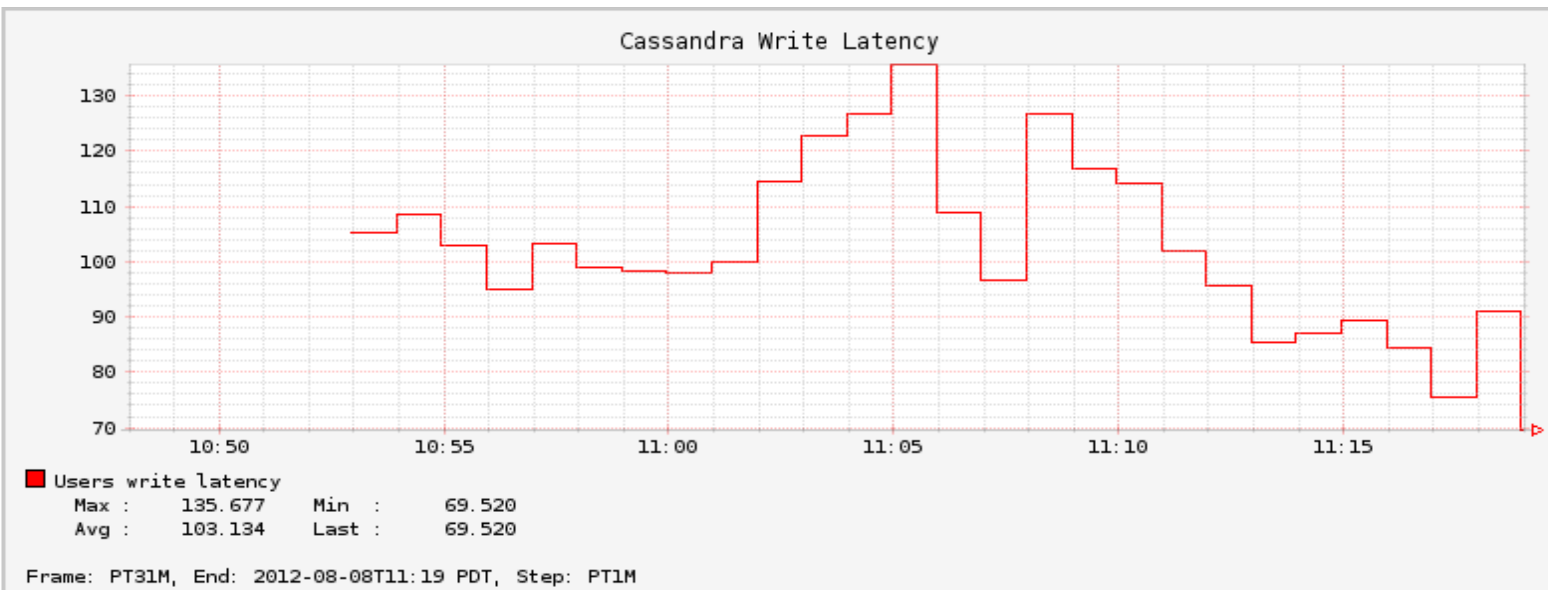


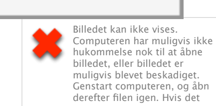
The past
m2.4xlarge

Microseconds

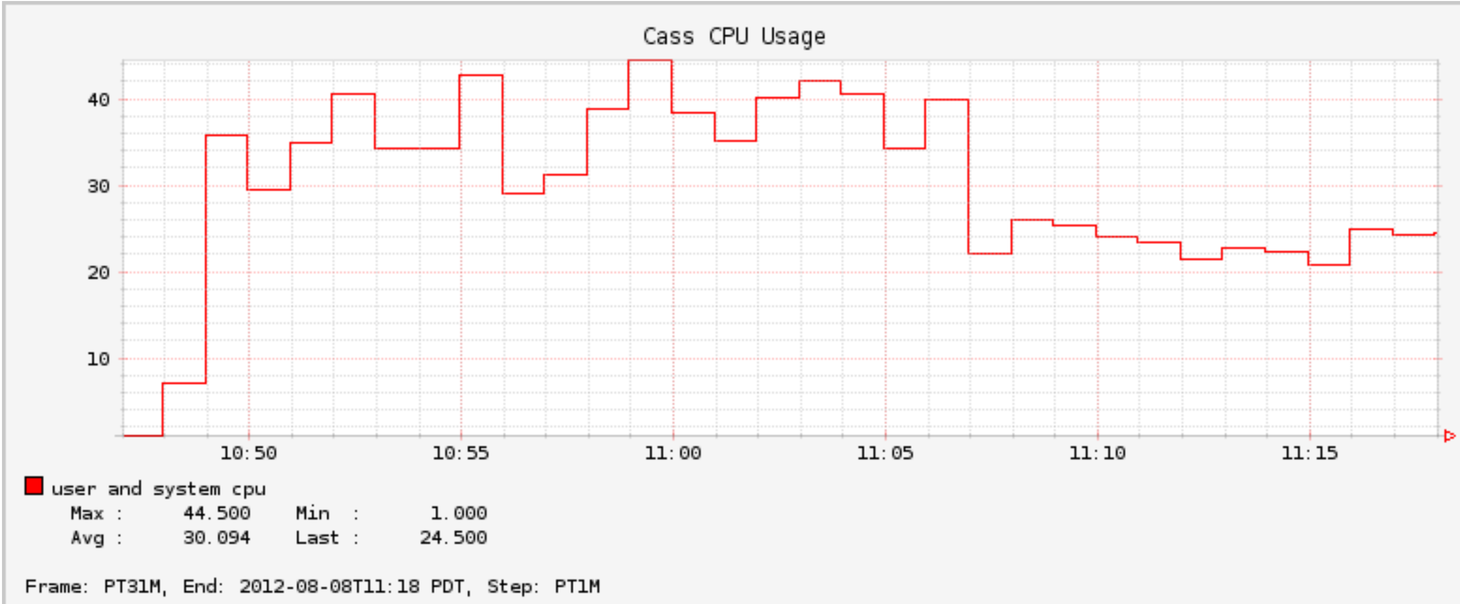


The future
hi1.4xlarge

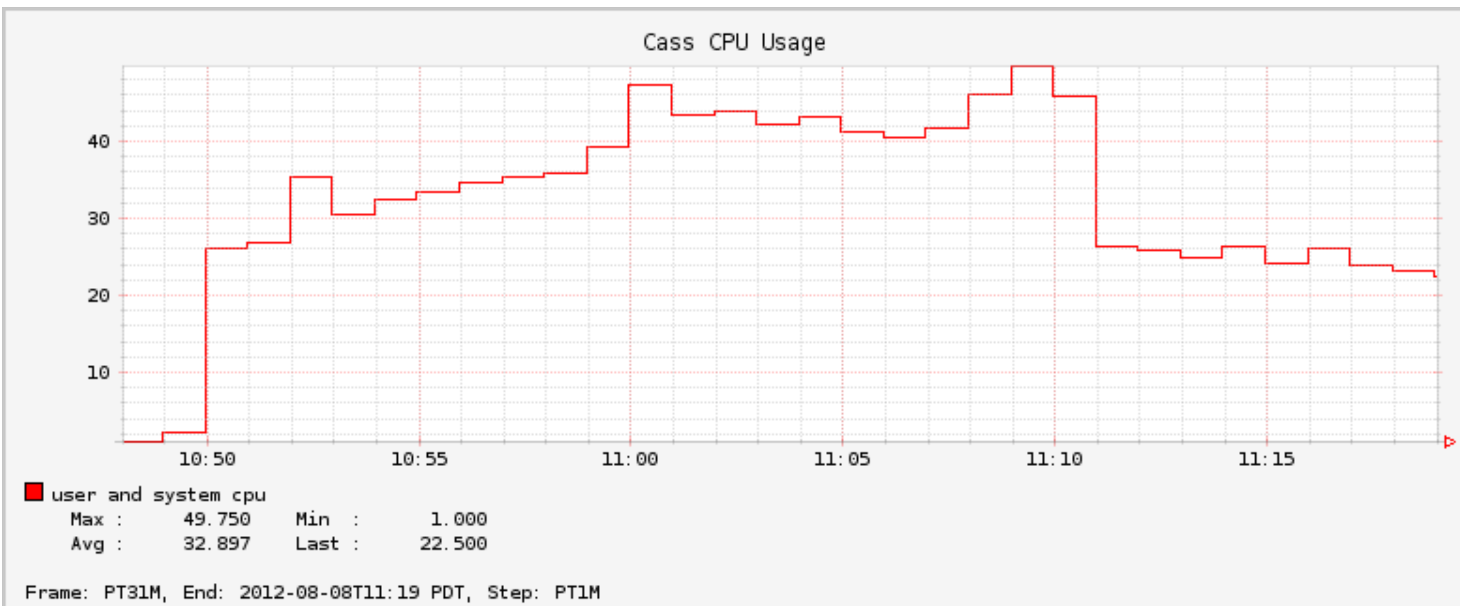


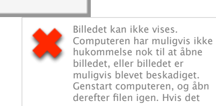


The past
m2.4xlarge

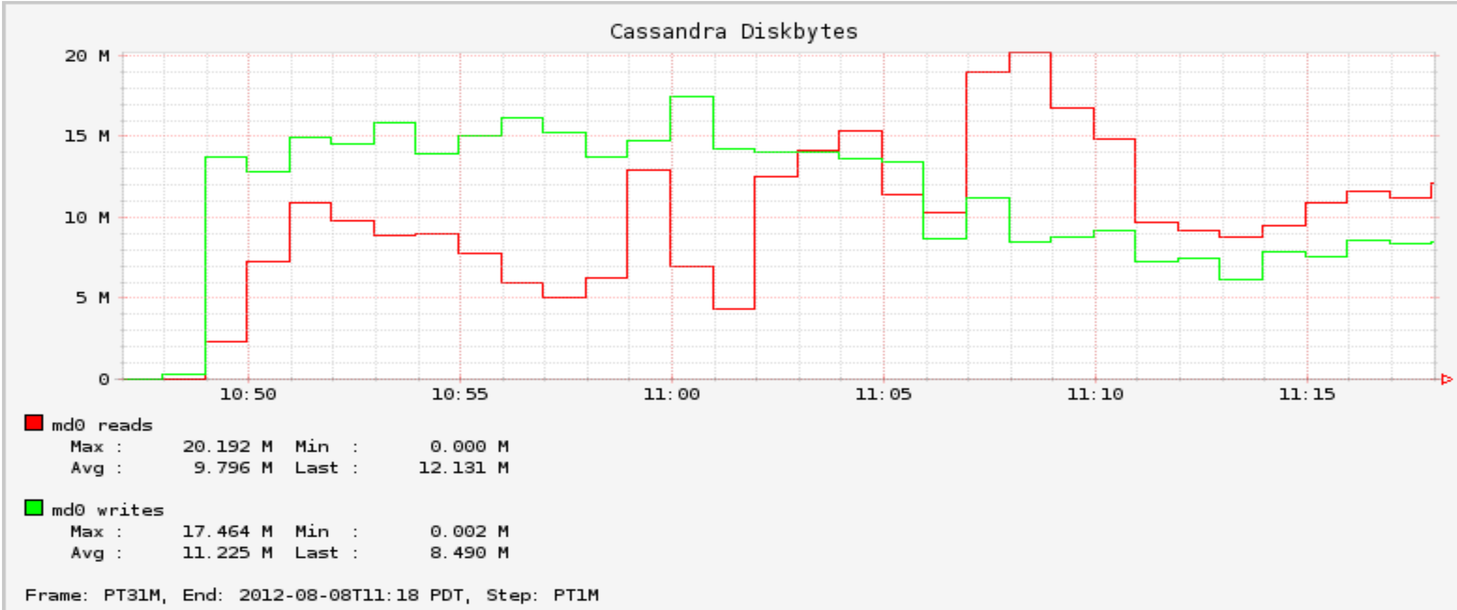


The future
hi1.4xlarge

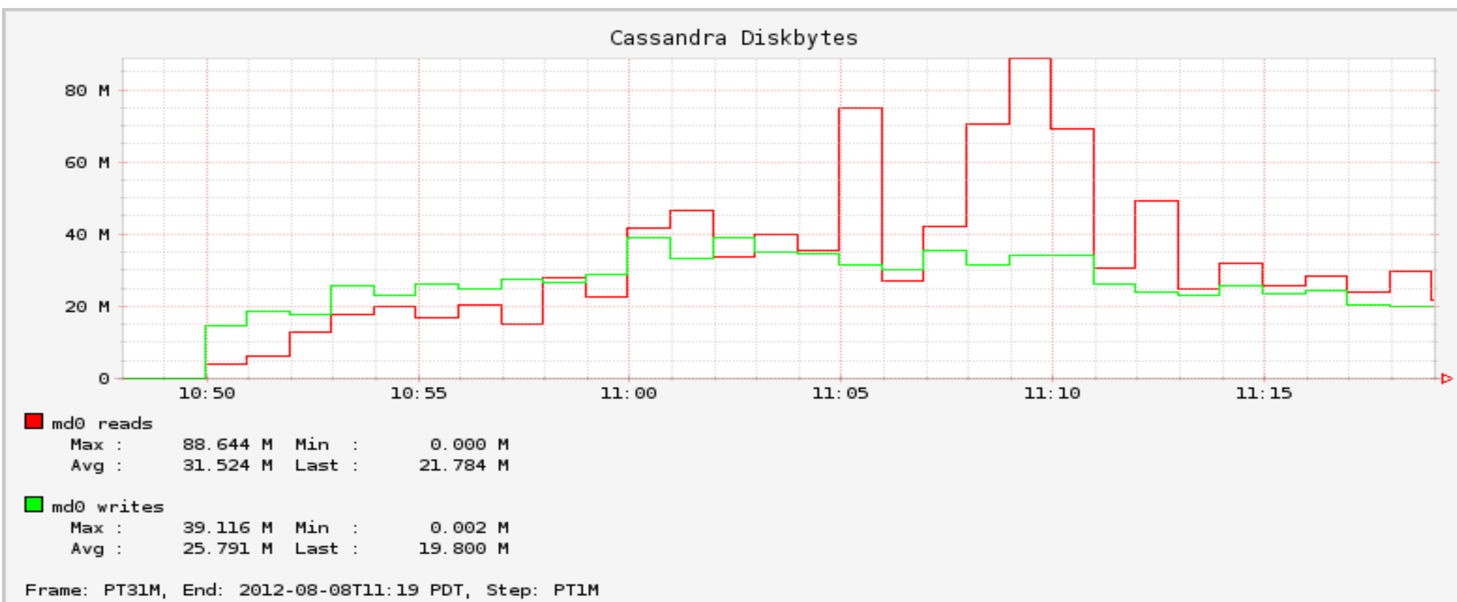




The past
m2.4xlarge



The future
hi1.4xlarge



Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

Next Steps

- Migrate Production Cassandra to SSD
  - Several clusters done
  - ~100 SSD nodes running
- Autoscale Cassandra using Priam
  - Cassandra 1.2 Vnodes make this easier
  - Shrink Cassandra cluster every night

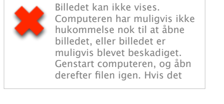
Skynet

A Netflix Hackday project that might just terminate the world…

(hack currently only implemented in Powerpoint – luckly)

The Plot (kinda)

- Skynet is a sentient computer that defends itself when people try to turn it off
- Connor is the guy who eventually turns it off
- Terminator is the robot sent to kill Connor



The Hacktors

- Cass_skynet is a Cassandra cluster that detects that it is being attacked and responds
- Connor_monkey kills cass_skynet nodes
- Terminator_monkey kills connor_monkey nodes

Billedet kan ikke vises. Computeren har muligvis ikke hukommelse nok til at åbne billedet, eller billedet er muligvis blevet beskadiget. Genstart computeren, og åbn derefter filen igen. Hvis det

The Hacktion

- Cass_skynet stores a history of its world and action scripts that trigger from what it sees
- Action response to losing a node
  - Auto-replace node and grow cluster size by one
- Action response to losing more nodes
  - Replicate cluster into a new zone or region
- Action response to seeing a Connor_monkey
  - Startup a Terminator_monkey in that zone

Implementation (plan)

- Priam
  - Autoreplace missing nodes
  - Grow cass_skynet cluster
  - Increase replication to new zone (new code)
  - Increase replication to new region (new code)
- Cassandra Keyspaces
  - Actions – scripts to be run
  - Memory – record event log of everything seen
- Cron job once a minute
  - Extract actions from Cassandra and execute

- Chaos Monkey configuration
  - Terminator_monkey: pick a zone, kill any connor_monkey
  - Connor_monkey: kill any cass_skynet or terminator_monkey

Simulation


TERMINATOR
SALVATION


SKYNET
NEURAL NET-BASED ARTIFICIAL INTELLIGENCE


SKYNET
SKYNET
SKYNET
SKYNET

Takeaway

Netflix has built and deployed a scalable global platform based on Cassandra and AWS.

Key components of the Netflix PaaS are being released as Open Source projects so you can build your own custom PaaS.

SSD's in the cloud are awesome….

http://github.com/Netflix
http://techblog.netflix.com
http://slideshare.net/Netflix

http://www.linkedin.com/in/adriancockcroft
@adrianco #netflixcloud #cassandra12