# A Non-Relational Storage Analysis


## Cassandra & Couchbase

Alexandre Fonseca, Anh Thu Vu, Peter Grman

Cloud Computing - 2nd semester 2012/2013
Universitat Politècnica de Catalunya

# Microblogging - big data?

- Twitter:
  - Over 500 million registered users.
  - 340 million tweets per day.
  - More than 12TB of data per day in 2010.

- Hard to provide this service relying solely on a small number of centralized servers.
  - Scaling up has its limits.

# Cassandra

- Tested version: 1.2.4, April 2013
- Data model:
  - Hybrid between key-value and tabular storage.
  - Data split in column families (like RDS tables).
  - Column families split into rows (indexed by row ke
  - Dynamic columns (schema).
- Interface: CQL via Thrift.
- Our setup:
  - 256 virtual nodes per physical node.
  - Cluster given topology awareness using EC2 snitch.
  - Consistency level of ONE for R/W.
  - Replication factor of 3 per datacenter.

# Couchbase

- Tested version: 2.0.1, Apr 2013
- Data model
  - Key (String) - Value (JSON)
  - View/Index: incremental MapReduce
- Automatic Replication
  - Configurable replication factor for each bucket with a max of 3.
  - Set to 3 in single region experiments
  - Set to 2 in multi-region experiments
- Consistency:
  - Strong consistency for access with "key"
  - Eventual consistency for queries on views

# PyDLoader

- Custom Python script
- 2 components:
  - Manager:
    - Interactive console.
    - Deploy new nodes.
    - Install and control slaves on those nodes.
  - Slaves:
    - Automatic setup, populating and takedown of database clusters (database slaves).
    - Generation of application workload towards database clusters (workload slaves).
- Used libraries: Boto (AWS), Paramiko (SSH), RpyC (RPC).

# Evaluation

- Done using Amazon Web Services (AWS).
- 6 database nodes (m1.small)
  - 1.7 GB of RAM
  - 1 compute unit
  - 150GB of storage
- 12 workload nodes (t1.micro)
  - 615 MB or RAM
  - 1 compute unit
  - No local storage (NAS)
- Distributed equally over 2 datacenters in Ireland.

# Evaluation

- Focus on 3 main operations:
  - Tweet
  - Userline (all tweets by user)
  - Timeline (all tweets by people user is following).
- Limit to 50 items per userline/timeline.
- Basic data structure:
  - Aggregate data according to operations, not entities:
    - Timeline table/bucket:
      - UserID, TweetID, PostedBy, Body
    - Userline table/bucket:
      - UserID, TweetID, Body

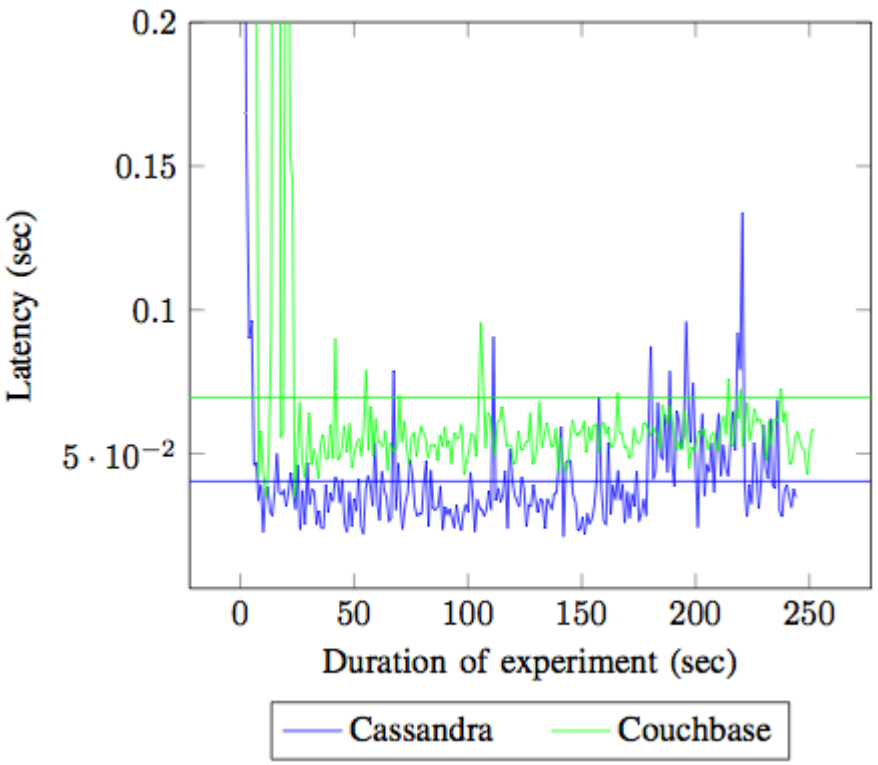
# Ease of Setup

- Cassandra - Awesome!
  - Install, configure addresses, partitioner, snitch, replication factors and seed nodes, launch!
  - Automatic partitioning and replication.
  - Automatic adjustment to churn.
- Couchbase - Equivalently Awesome!
  - Install, configure RAM, directory, launch!
  - Easily add/failover/remove servers
  - Rebalance: background process, asynchronous, incremental
  - Automatic replication, partitioning and node failure detection.

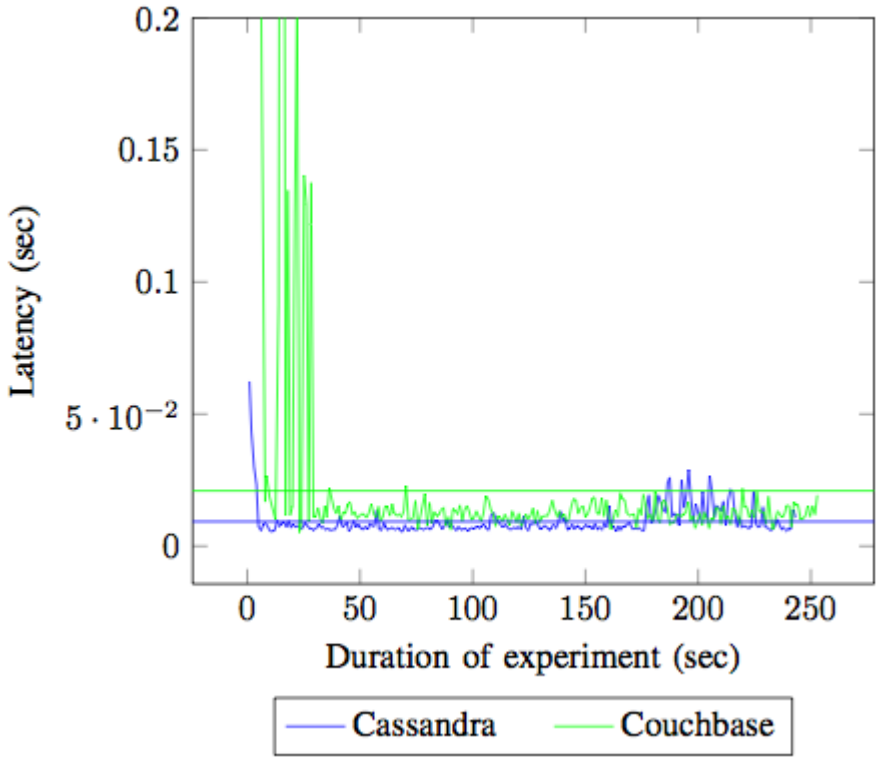
# Setup time

- Cassandra:
  - 6 nodes form and stabilize ring after ~2 minutes.
  - Populating with sample data:
    - ~1 minute in 1 node configuration.
    - 6.5 minutes in 6 node configuration.
- Couchbase:
  - 6 nodes form and stabilize ring after ~2 minutes.
  - Populating with sample data:
    - ~1 minute in 1 node configuration.
    - ~2 minutes in 6 node configuration.

# Latency

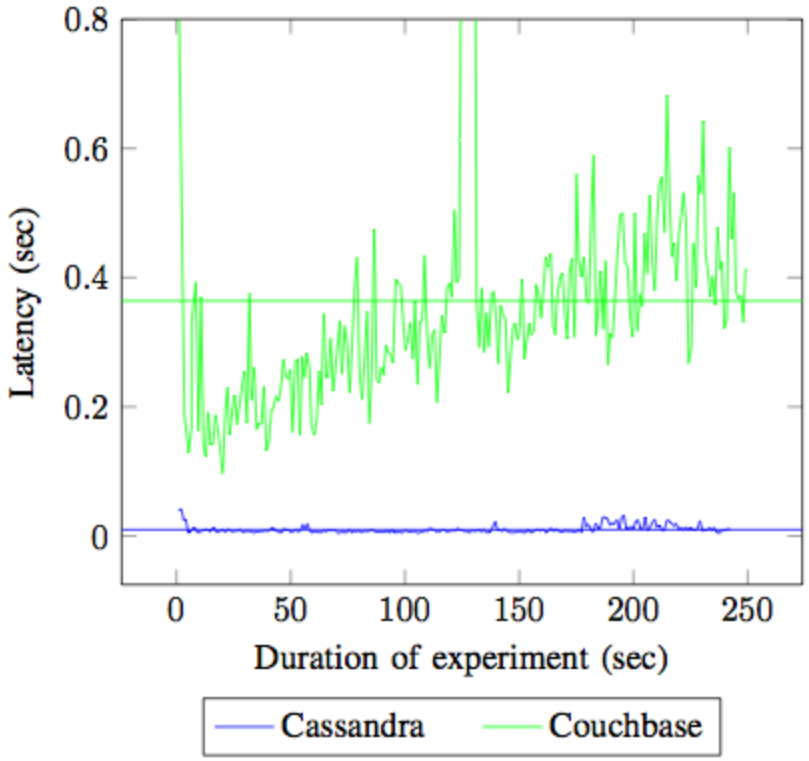



Tweet Latency



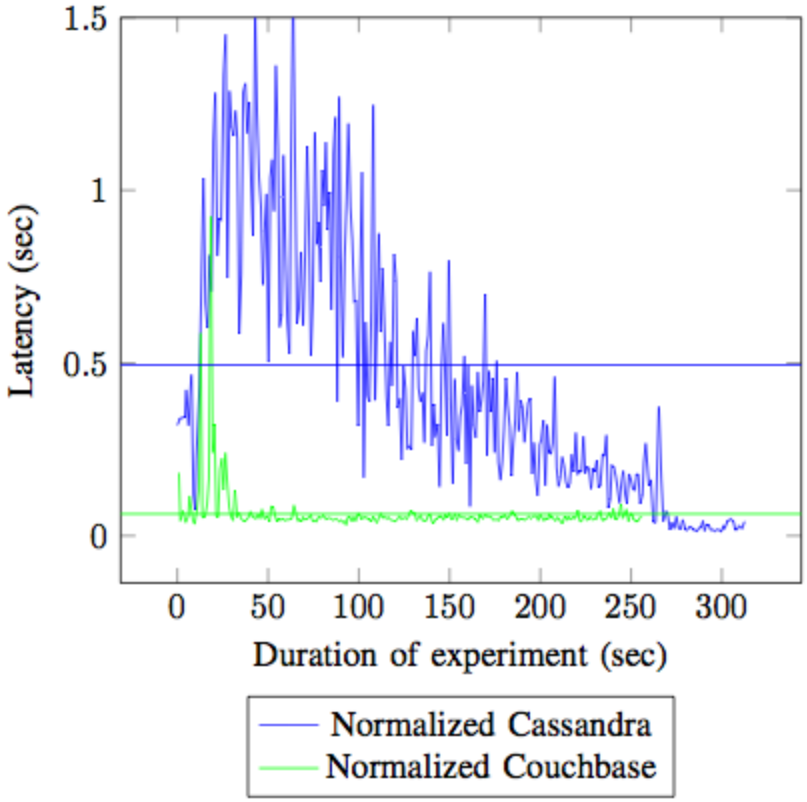

Userline Latency

# Latency



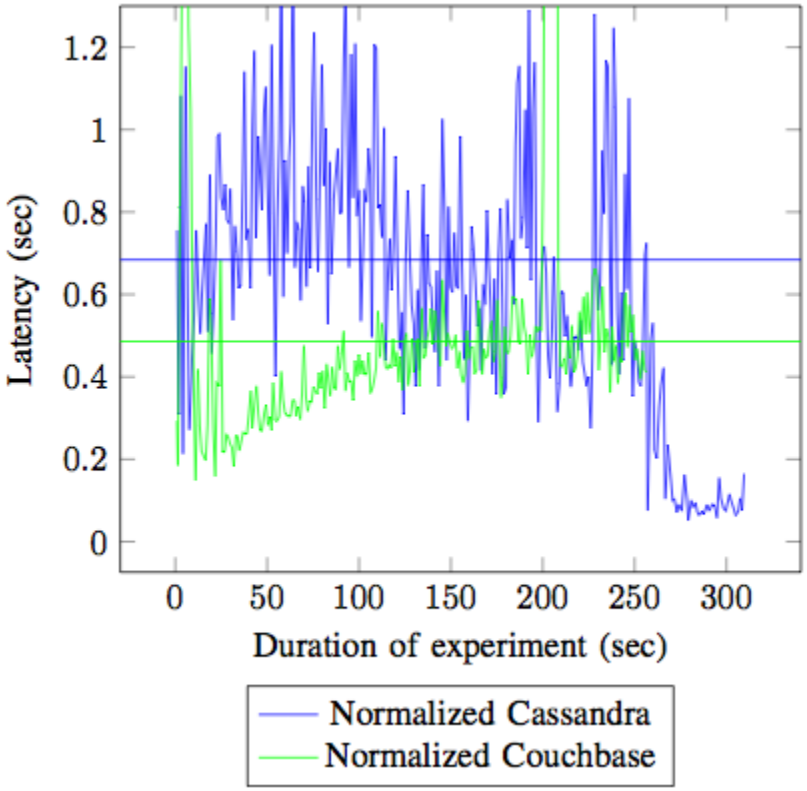

Timeline Latency

# Normalized latency

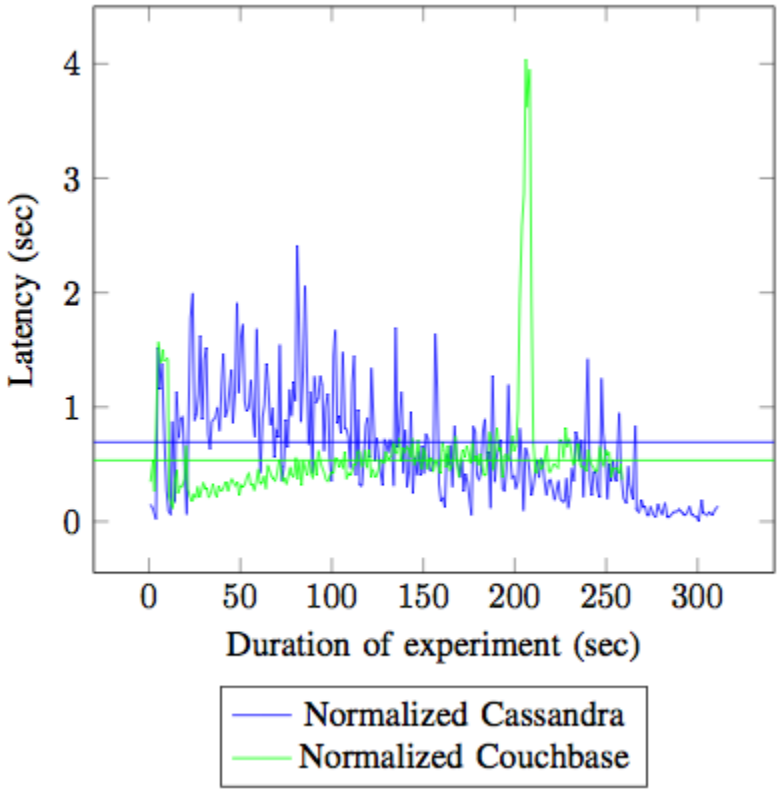



Tweet Latency

# Normalized latency

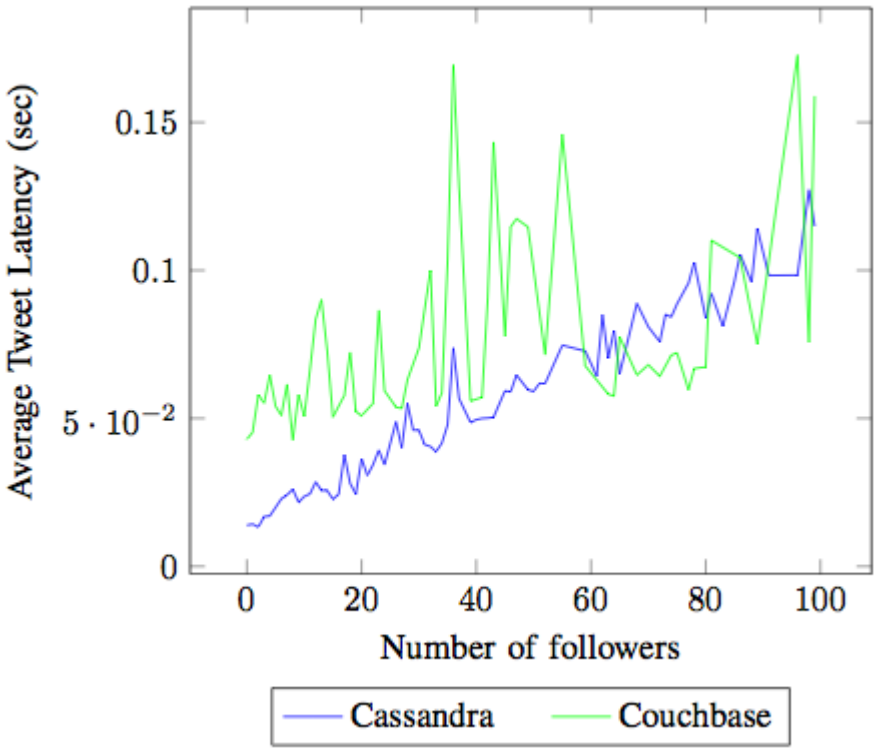



Userline Latency




Timeline Latency

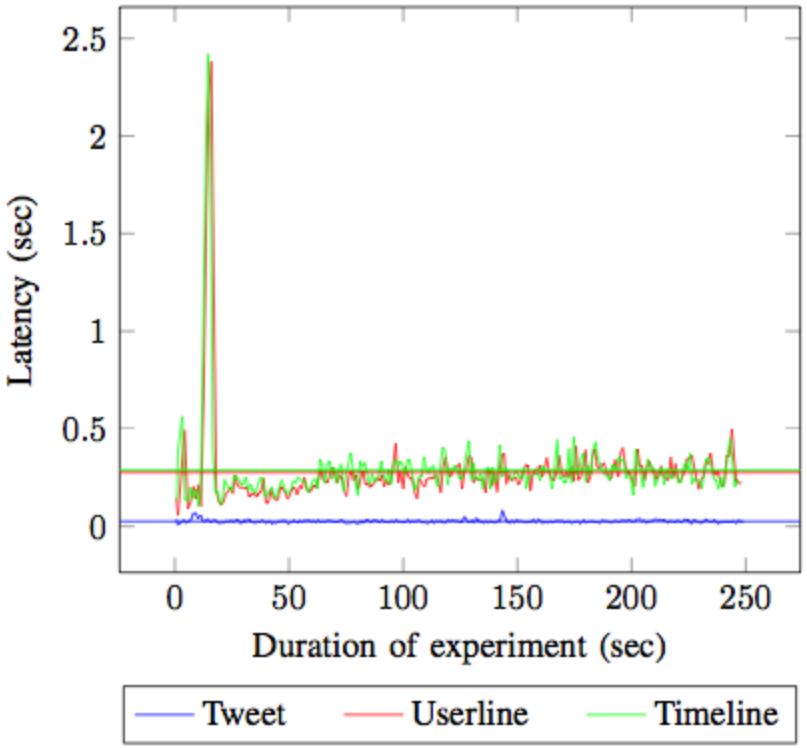
# Tweets & Denormalization




Tweet Latency v.s. Number of Followers

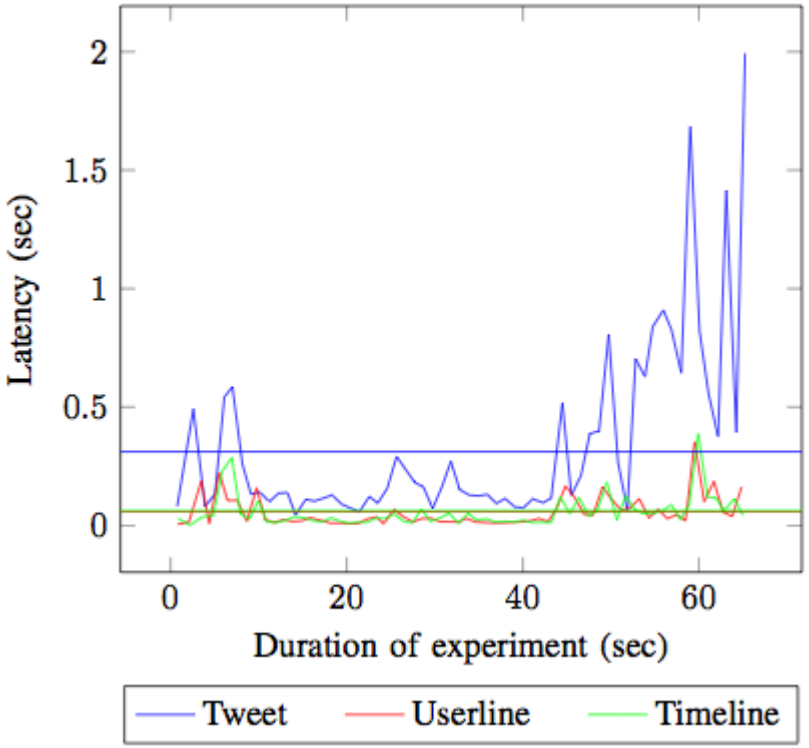
# Tweets & Denormalization

- Immediate denormalization not scalable.
- How to make asynchronous?
  - Cassandra:
    No native support.
    External processing.
  - Couchbase: Views!



# Reconfiguration Latency

- Cassandra
  Adding a new node
  to a cluster: ~5 mins.



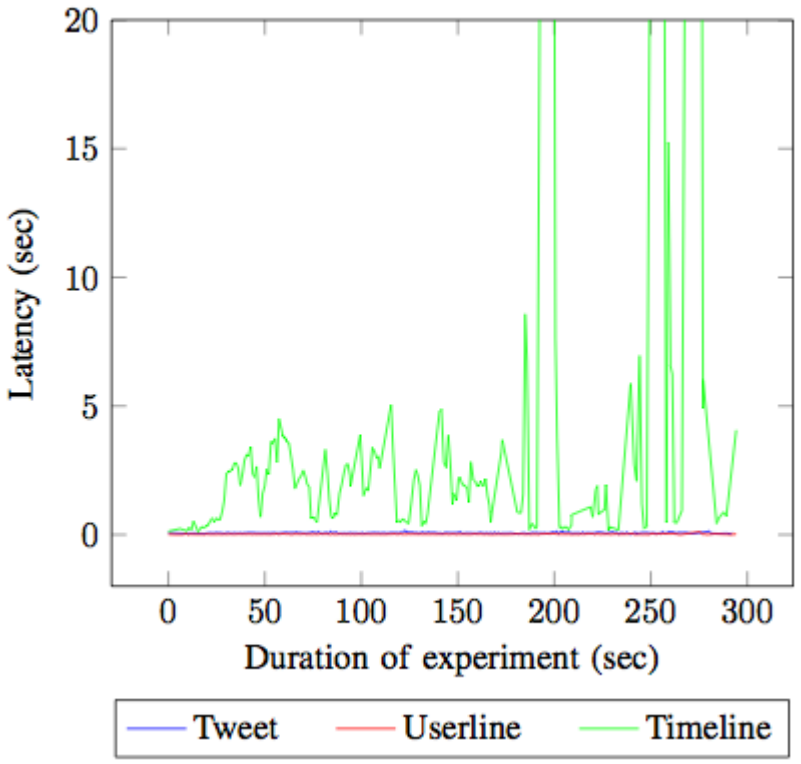

Latency while node joining cluster

# Reconfiguration Latency

- Couchbase

  Adding a new node to a cluster: immediate

  BUT rebalance ~ 30 mins



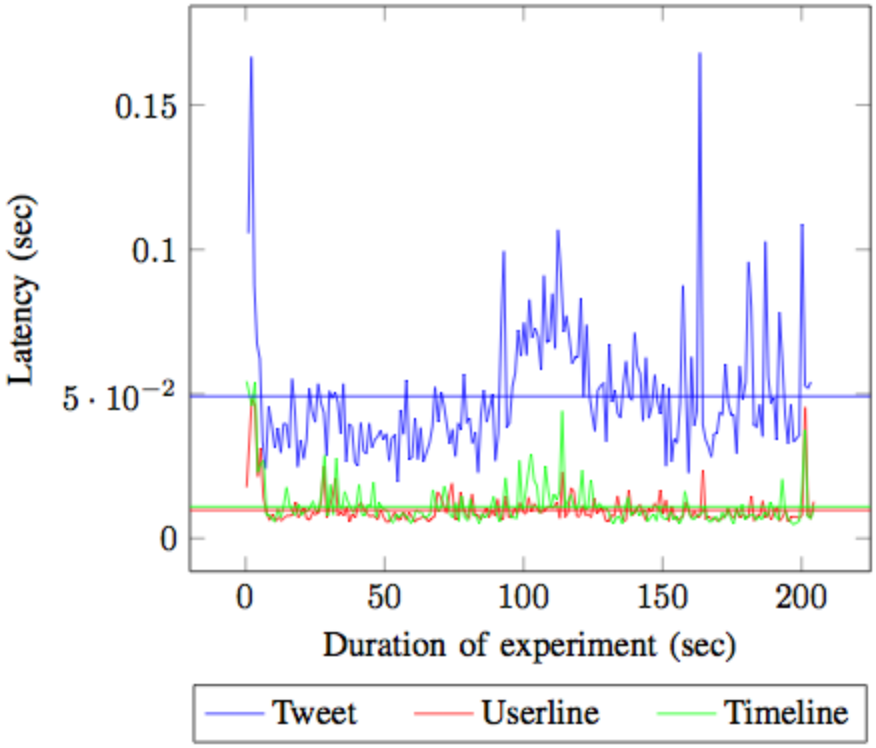

Latency while node joining cluster

# Consistency/Convergence

- Cassandra:
  - Average of 0.096498 seconds to detect new tweet.
  - Standard deviation of 0.096319:
    - Same datacenter => very fast detection.
    - Different datacenter => slower detection.
- Couchbase:
  - Average of 0.007501 seconds to detect new tweet with standard deviation of 0.012476
  - The delay for new tweet to appear on timeline proportional to the schedule period

# Replication

- Cassandra:
  - Very flexible in terms of replication configuration.
  - Per datacenter replication factors with no hard-coded limitations.

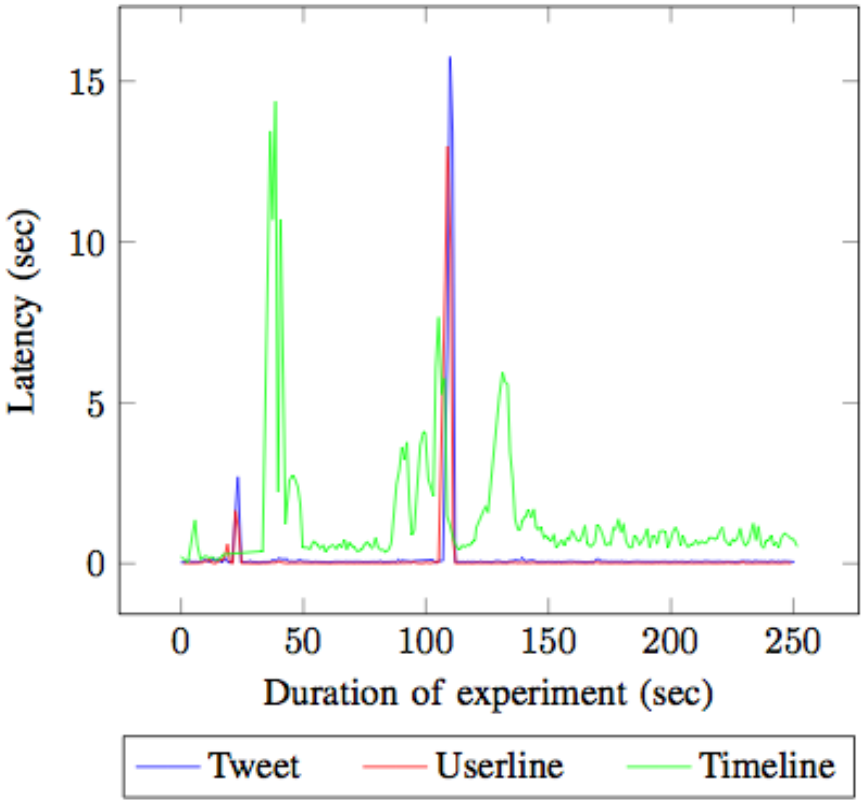



Behaviour under crash of 2 nodes @ second 100th

# Replication

- Couchbase

  Automatic and configurable per bucket with a limit of (1+3) replicas.




Behaviour under crash of 2 nodes @ second 30th, 100th

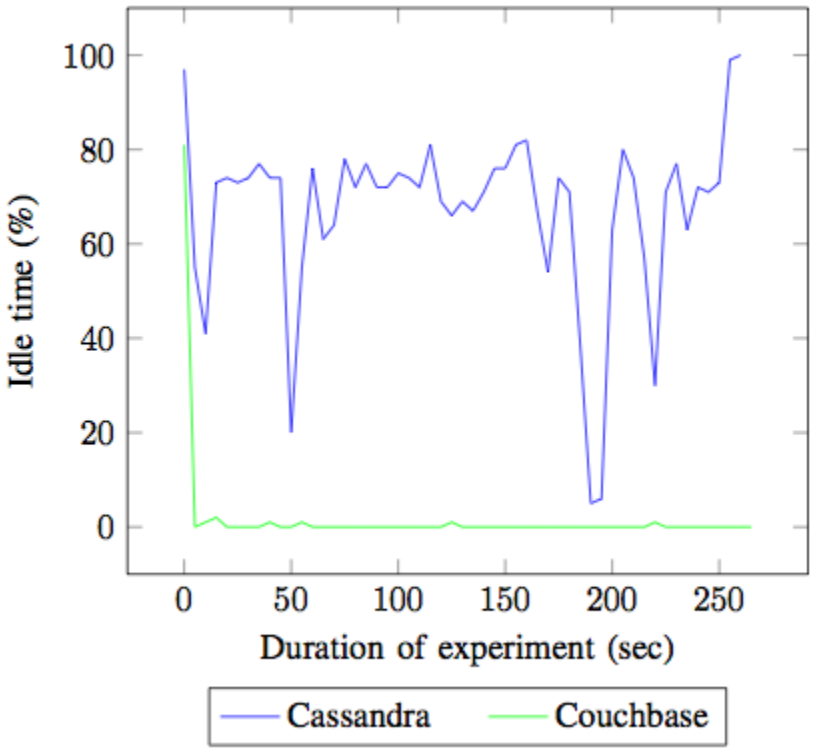
# Load Balancing

- Cassandra:
  - Average data ownership per node: 16.68%
  - Standard deviation: 1.23%
- Couchbase:
  - Evenly distributed with standard deviation:

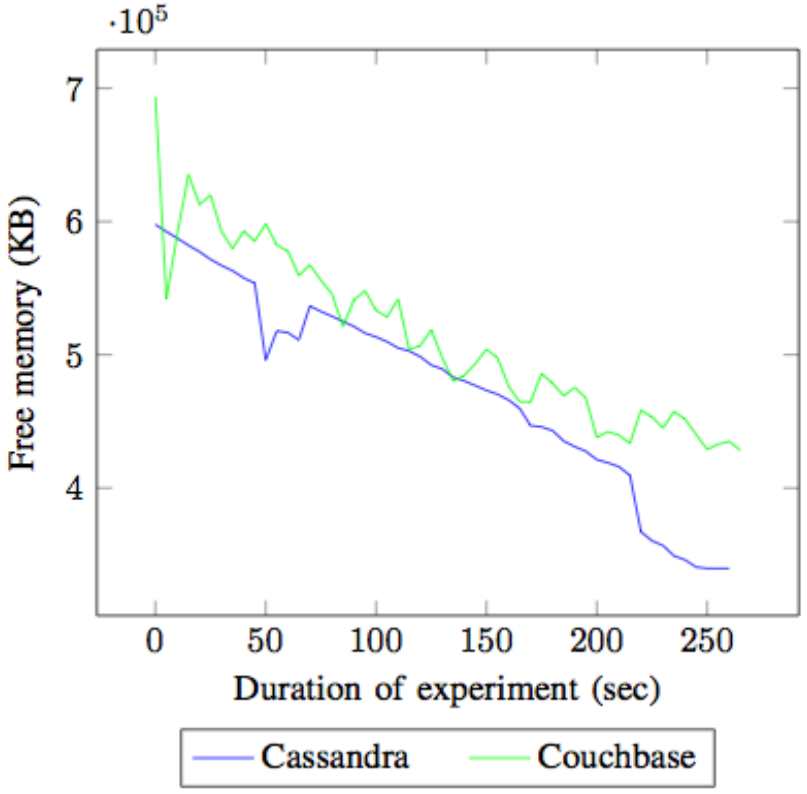
    0.65% for data on disk

    0.04% for RAM usage

# System Load

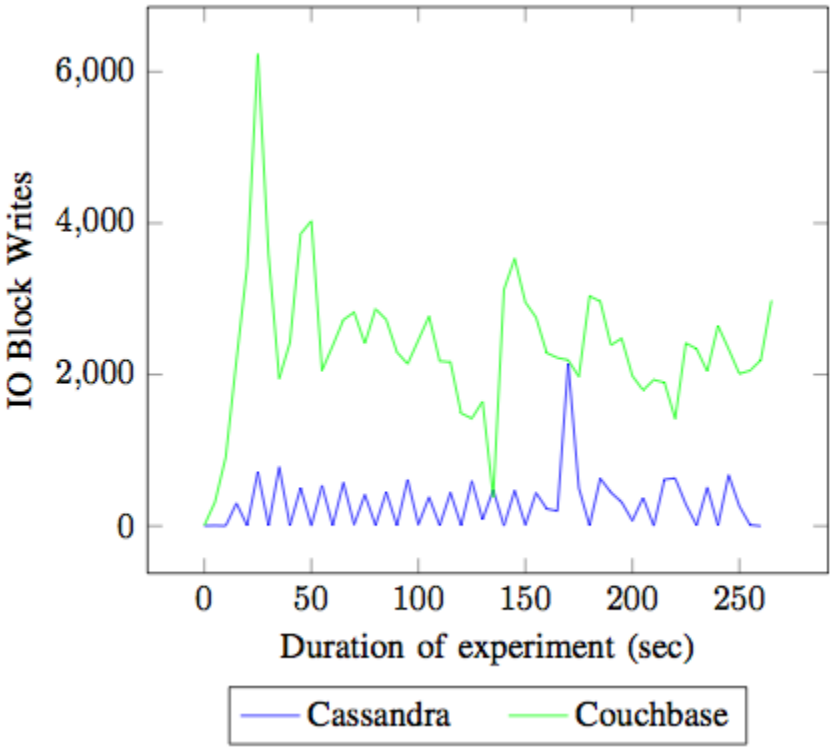


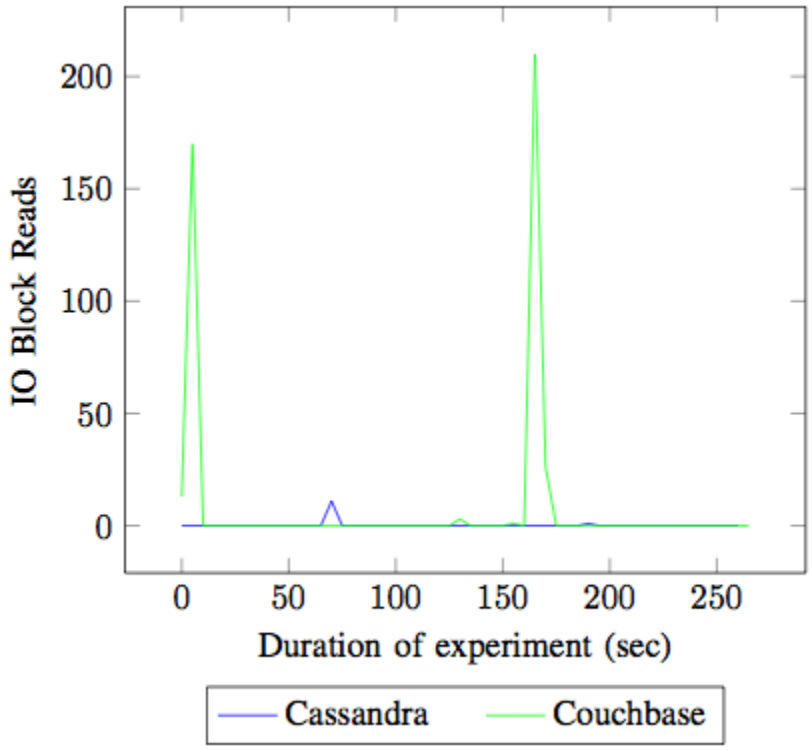

CPU Idle Time




Free Memory

# System Load




Disk IO

# Disk Space Usage

- SQLite database: 1.6MB
- Cassandra:
  - Fully denormalized (single node): 16.31MB.
  - Fully denormalized (6-nodes): 9.5MB/node.
    - Includes partitioning and replication.
  - "Normalized" (no body in userline/timeline):
    - Single node: 2.8MB
    - 6-nodes: 1.6MB/node.
  - Commit log after populating: 54MB
  - Need 50% of free disk space at all times:
    - Column family compactions.
    - Data redistributions.

# Disk Space Usage

- Couchbase:
  - Total of 250MB distributed across 6 nodes.
  - No minimum free space requirement.

- Lack of disk usage limitations in both DBs:
  - Not ideal for voluntary computing systems.

# Multi Region Setup

- Nodes distributed across Ireland and N.Virginia.

| | Single Region | | Multiple Region | |
|---|---|---|---|---|
| | Cassandra | Couchbase | Cassandra | Couchbase |
| Setup+Populate time | 8.5 min | 4 min | 8.7 min | 12 min |
| Avg. Tweet | 0.0405 sec | 0.0696 sec | 0.3043 sec | 0.0681 sec |
| Avg. Userline | 0.0093 sec | 0.0208 sec | 0.0477 sec | 0.0169 sec |
| Avg. Timeline | 0.0101 sec | 0.3639 sec | 0.0836 sec | 0.9791 sec |
| Consistency | $\mu = 0.096498$<br>$\sigma = 0.096319$ | $\mu = 0.007501$<br>$\sigma = 0.012476$ | $\mu = 0.609560$<br>$\sigma = 1.080330$ | $\mu = 2.929887$<br>$\sigma = 3.384052$ |

# Conclusion

- Easy cluster setup
  - Allows horizontal scaling over multiple nodes
- Improved performance through denormalization
- Higher storage requirements
- The future is hybrid
  - A mix of RDS and NoSQL-Systems
  - Cassandra's CQL, CouchBase's Views, NoSQL in RDS