

# Cassandra at Soundcloud

Alexander Simmerl, Jake Maizel  
9-June, 2011

# Dashboard

- Most popular endpoint
- One user - many updates
- Capped Collection in MySQL
- 200 Million Events in one Table

# Requirements

- Writes should be
  - fast(enough)
  - sequential disk appends
  - avoid disk seeks

# Schema

- Personalized Index
- UserID row keys
- TimeUUID SuperColumns keys
- Origin and Owner
- Business-logic is encoded in Tags

# Writes

- Own Ruby repository
- Shipped as rubygem with executables
- AMQP pipeline
- Three steps: fanout, tagging, serializing
- Pure SQL and Cassandra gem

# Reads

- Own module in main rails app
- Reads sequentially
- Reversed chronological order
- Builds Dashboard Stories

# Operations

# Metrics

- 6-nodes (Soon to be 12)
- 3+ TB Data
- 4 Column Families in one key space
- Billions of Value/Key Pairs
- 1400 write per sec and 30 reads per sec

# Operations Overview

- Running 0.6.6, planning an upgrade soon
- Very stable and mostly maintenance free
- Storage use have been the main pain

# Nice Tricks

- Used data center awareness to move data centers
- Use rsync to move data and logs to new hardware
- Split write stream at application to write to more than one cluster (Ex. zero downtime upgrades)

# Things we're psyched about

- Counters!
- Improved space usage!
- Other things!

# jmx2Munin

- <https://github.com/tcurdt/jmx2munin>

# Questions?

- [alx@soundcloud.com](mailto:alx@soundcloud.com) (Lead Developer)
- [jake@soundcloud.com](mailto:jake@soundcloud.com) (Head of NetOps)
- [eb@soundcloud.com](mailto:eb@soundcloud.com) (Lead Ops)