Intro Cassandra

Adelaide Big Data Meetup

instaclustr.com
@Instaclustr
Who am I and what do I do?

• Alex Lourie

• Worked at Red Hat, Datastax and now Instaclustr

• We currently manage x10s nodes for various customers, who do various things with it.
Objectives

• A quick history of Databases
• Introducing Cassandra
• Why use Cassandra?
• Introducing Instaclustr
• Demo?
Start Demo

• It takes a while, so let’s do it.
1980 - Stand Alone and Mainframes

1990 - 2005 Networked Computing

2005+ Real Time Web and Big Data
End 90’s - Early 2000’s

• What happens when you have more data than could fit on a single server?
Throw money away at the problem
Lets try a little computer science instead

- **BigTable (2006)** - 1 Key: Lots of values, Fast sequential access
- **Dynamo (2007)** - Reliable, Performant, Always On,
- **Cassandra (2008)** - Dynamo Architecture, BigTable data model and storage (NoSQL)
What is NoSQL?

• Key/Value (data) store
• Non-relational, distributed and horizontally scalable
• Schema-free, easy replication support, simple API, eventually consistent
• Supports huge amounts of data
Cassandra story

- Started at Facebook
- Released as an Open Source project
- Datastax formed
Cassandra

• Massively scalable, partitioned row store
• Masterless architecture
• Linear scale performance
• No single points of failure
• Read/Write support across multiple data centers & cloud availability zones.
Why use C*?

- You need to support tens of thousands to tens of millions operations per second
- You need to store and access terabytes to petabytes of data;
- You need fast (less than 5-10 millisecond) response time to database operations; and
- You need a service with no downtime.
What are the benefits to this approach

• Linear scalability
• High Availability
• Use commodity hardware
Linear scalability

Scale-Up Linearity

Client Writes/s by node count – Replication Factor = 3
High Availability?

“During Hurricane Sandy, we lost an entire data center. Completely. Lost. It. Our application fail-over resulted in us losing just a few moments of serving requests for a particular region of the country, but our data in Cassandra never went offline.”

Nathan Milford, Outbrain’s head of U.S. IT operations management
What are the benefits to this approach
How does it work?
One database, many servers

• All servers (nodes) participate in the cluster
• Need more capacity add more servers
• Multiple servers == built in redundancy
<table>
<thead>
<tr>
<th>Name</th>
<th>Age</th>
<th>Postcode</th>
<th>Gender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>34</td>
<td>2000</td>
<td>F</td>
</tr>
<tr>
<td>Bob</td>
<td>26</td>
<td>2000</td>
<td>M</td>
</tr>
<tr>
<td>Eve</td>
<td>25</td>
<td>2004</td>
<td>F</td>
</tr>
<tr>
<td>Frank</td>
<td>41</td>
<td>2902</td>
<td>M</td>
</tr>
</tbody>
</table>
How does it work?

consistentHash("Alice")

Replication Factor = 3
How do we keep data consistent?

consistentHash("Alice")

client

Write

Ack

0

2

4

CL.ONE
How do we keep data consistent?

consistentHash("Alice")

client

Write

Ack

Ack

Ack

0

CL.ALL

2

4
How do we keep data consistent?
Also supports multi-dc replication
Add capacity

consistentHash("Alice")
C* internals

• Logging data in the commit log
• Writing data to the memtable
• Flushing data from the memtable
• Storing data on disk in SSTables
• Compaction + Repairs
Cassandra scales well beyond relational databases and is more manageable for high-availability at scale. It is highly cost-effective compared to commercial relational databases.

But…Cassandra does not have the same analytical query capabilities (ie aggregations, joins) as a relational database.
C* vs Hadoop

Cassandra was designed as a **fast, reliable and scalable operational data store**. Hadoop was designed as a data store for vast amounts of data for **batch analytic processing**. Cassandra can provide faster operations and higher reliability than Hadoop.
C* vs MongoDB

MongoDB is **not masterless** making it **harder to manage** and imposing hard limits on its scalability. On the other hand, MongoDB does provide some additional flexibility in querying data.
Pitfalls

- CQL is not SQL!
- NoSQL is not RDBMS!
- Design your schema to match your data and usage patterns.
Instaclustr

Instaclustr is a company with extensive experience in designing, deploying and managing critical infrastructure for solutions that require immense scale.
Hosted and Managed Apache Cassandra, DSE, Apache Spark and other complimentary technologies.

We also deliver a wide range of related consulting and support services for these technologies.
Our tech

• AWS, Azure, IBM Softlayer, Heroku and others in development

• CoreOS as a common OS on all platforms.

• Docker containers for running applications.
Customers feedback

“Instaclustr provided us with a method for getting underway quickly with Cassandra and also delivered the support and expertise necessary to help us make our database as efficient as possible.”

Andre Barbosa, Head of Platform, Fling
“Instaclustr has enabled us to get underway quickly, the support team have been there from the beginning helping us to get it right the first time with our schema and architecture.”

Richard Wilson, Co-Founder, Maths Pathway
Customers feedback

“Instaclustr is awesome”, - Instaclustr
Check demo?
Questions?