Apache Cassandra and DataStax

DataStax EMEA

Agenda

DATASTAX

1. Apache Cassandra
2. Cassandra Query Language
3. Sensor/Time Data-Modeling
4. DataStax Enterprise
5. Realtime Analytics
6. What´s New

About me

DATASTAX

Christian Johannsen

Solutions Engineer @ DataStax

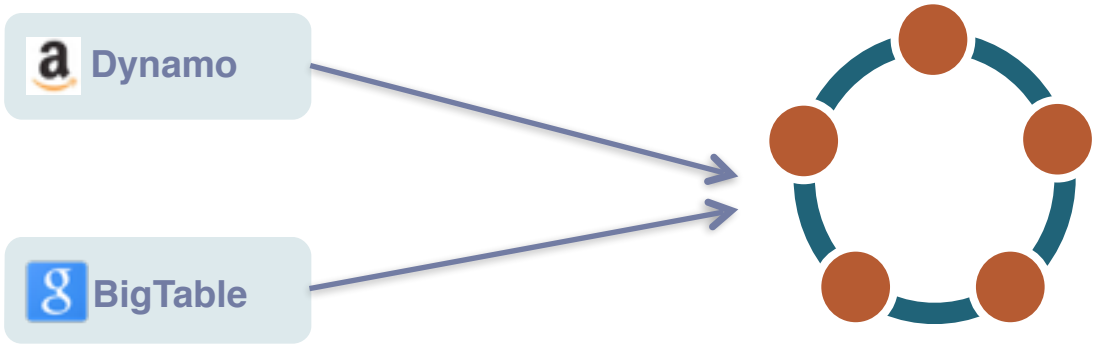
@cjohannsen81

Apache Cassandra

What is Apache Cassandra

DATASTAX

- Apache Cassandra is a massively scalable and available NoSQL database, providing extreme performance
- Cassandra is designed to handle big data workloads across multiple data center, with no single point of failure

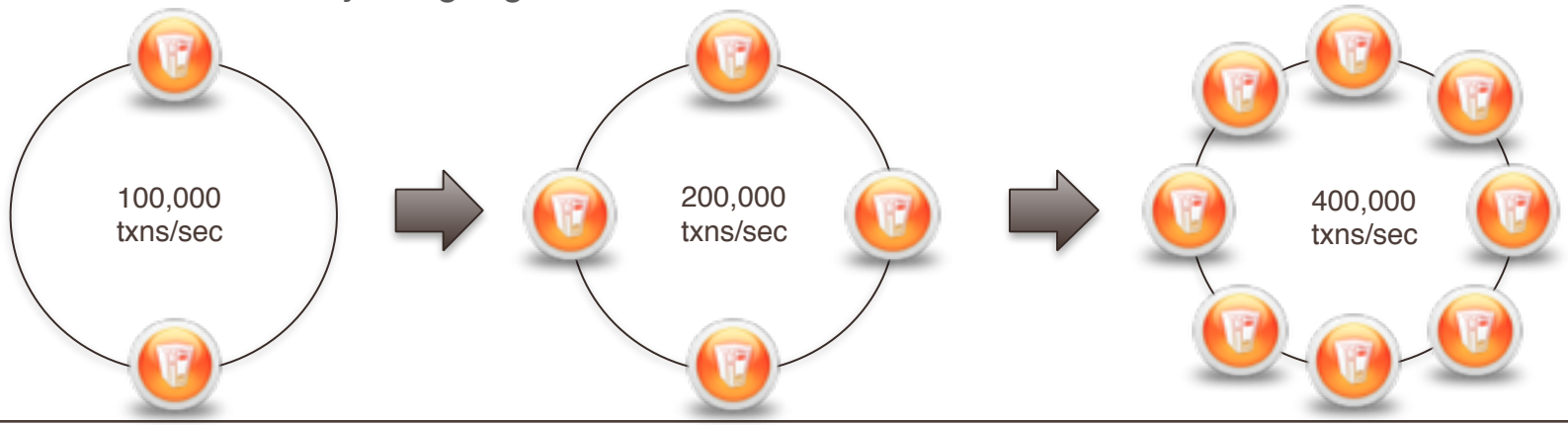


BigTable: http://research.google.com/archive/bigtable-osdi06.pdf
Dynamo: http://www.allthingsdistributed.com/files/amazon-dynamo-sosp2007.pdf

What is Apache Cassandra

DATASTAX

- Masterless Architecture with read/write anywhere design
- Continuous Availability with no single point of failure
- Multi-Data Center and Zone support
- Flexible data model for unstructured, semi-structured and structured data
- Linear scalable performance with online expansion (scale-out and scale-up)
- Security with integrated authentication
- Operationally simple
- CQL - Cassandra Query Language



Cassandra Adoption

DATASTAX

214 systems in ranking, February 2014

Rank	Last Month	DBMS	Database Model	Score	Changes
1.	1.	Oracle	Relational DBMS	1500.23	+32.43
2.	2.	MySQL	Relational DBMS	1288.39	-8.53
3.	3.	Microsoft SQL Server	Relational DBMS	1214.27	-11.75
4.	4.	PostgreSQL	Relational DBMS	230.45	+2.20
5.	↑ 6.	MongoDB	Document store	195.17	+16.94
6.	↓ 5.	DB2	Relational DBMS	188.46	+0.15
7.	7.	Microsoft Access	Relational DBMS	152.88	-22.11
8.	8.	SQLite	Relational DBMS	93.00	-4.29
9.	9.	Sybase ASE	Relational DBMS	87.88	-6.62
10.	10.	Cassandra	Wide column store	80.31	-0.87
11.	11.	Teradata	Relational DBMS	63.81	+2.36
12.	12.	Solr	Search engine	62.70	+2.37
13.	13.	Redis	Key-value store	55.81	+3.32
14.	14.	FileMaker	Relational DBMS	51.90	+2.27
15.	↑ 17.	Informix	Relational DBMS	35.67	+0.53
16.	16.	HBase	Wide column store	35.15	-0.11

Source: db-engines.com, Feb. 2014

Apache Cassandra - Important

DATASTAX

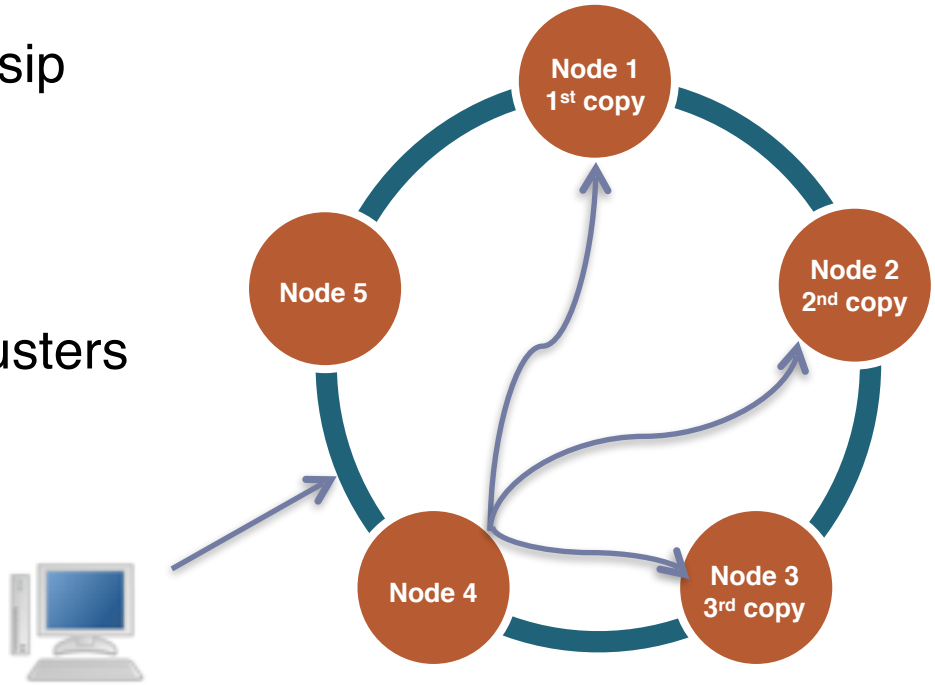
- **Cluster** - A ring of Cassandra nodes
- **Node** - A Cassandra instance
- **Replication-Factor** (RF) - How many copies of your data?
- **Replication-Strategy** - SimpleStrategy vs. NetworkTopologyStrategy
- **Consistency-Level** (CL) - What Consistency should be ensured for read/writes?
- **Partitioner** - Decides which node store which rows (Murmur3Partinioner as default)
- **Tokens** - Hash values assigned to nodes

Follow-Up: http://planetcassandra.org/blog/introduction-to-cassandra-clusters/

Cassandra - Locally Distributed

DATASTAX

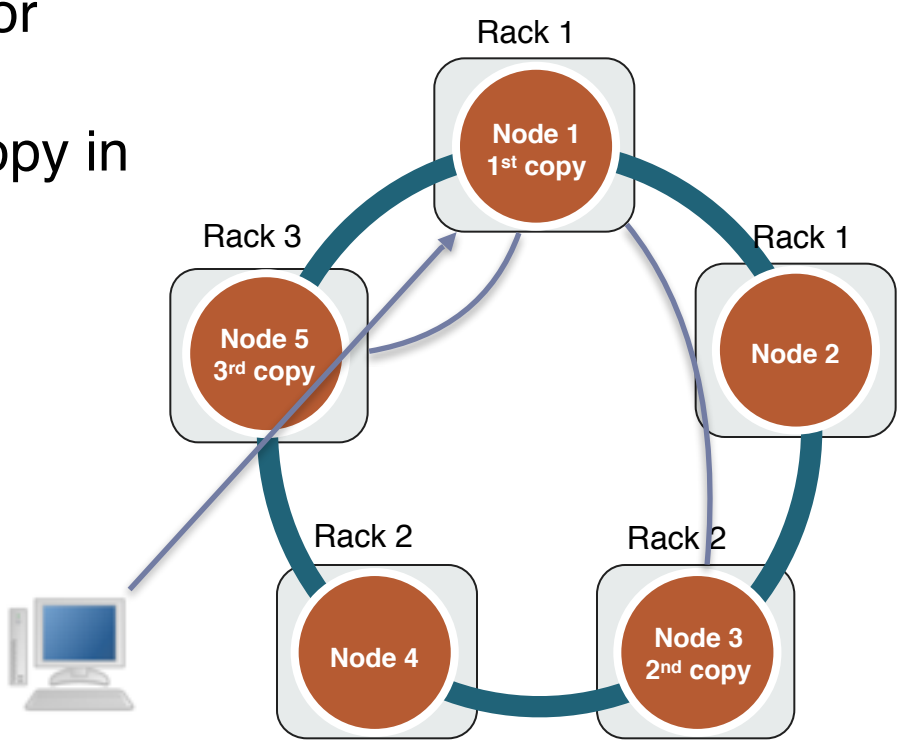
- Client reads or writes to any node
- Node coordinates with others (gossip protocol)
- Data read or replicated in parallel
- RF = 3 in this example
- Each node is strong 60% of the clusters Data i.e. 3/5



Cassandra - Rack/Zone aware

DATASTAX

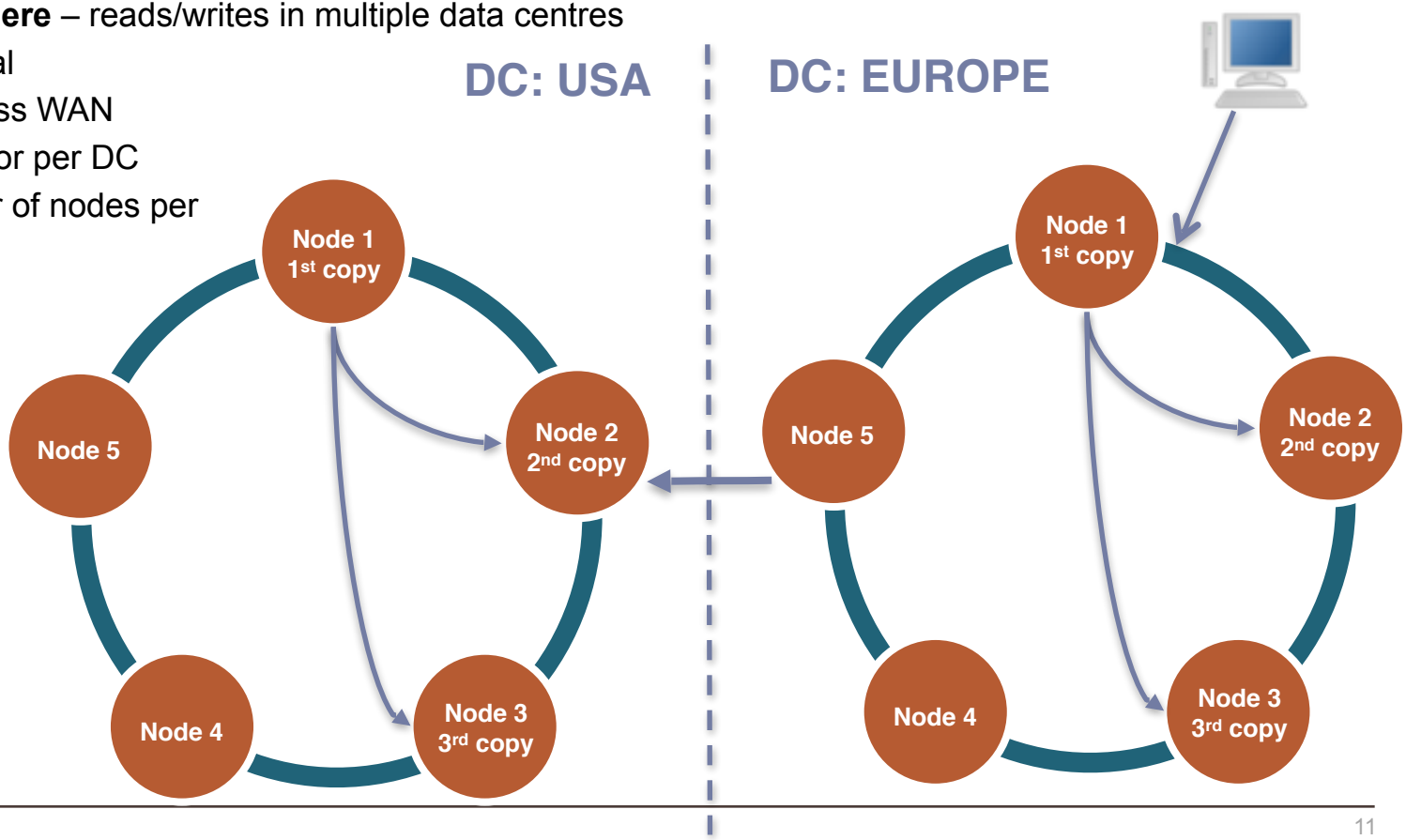
- Cassandra is aware of which rack or zone each node resides in
- It will attempt to place each data copy in a different rack
- RF=3 in this example



Cassandra - DC/Region aware

DATASTAX

- **Active Everywhere** – reads/writes in multiple data centres
- Client writes local
- Data syncs across WAN
- Replication Factor per DC
- Different number of nodes per data center



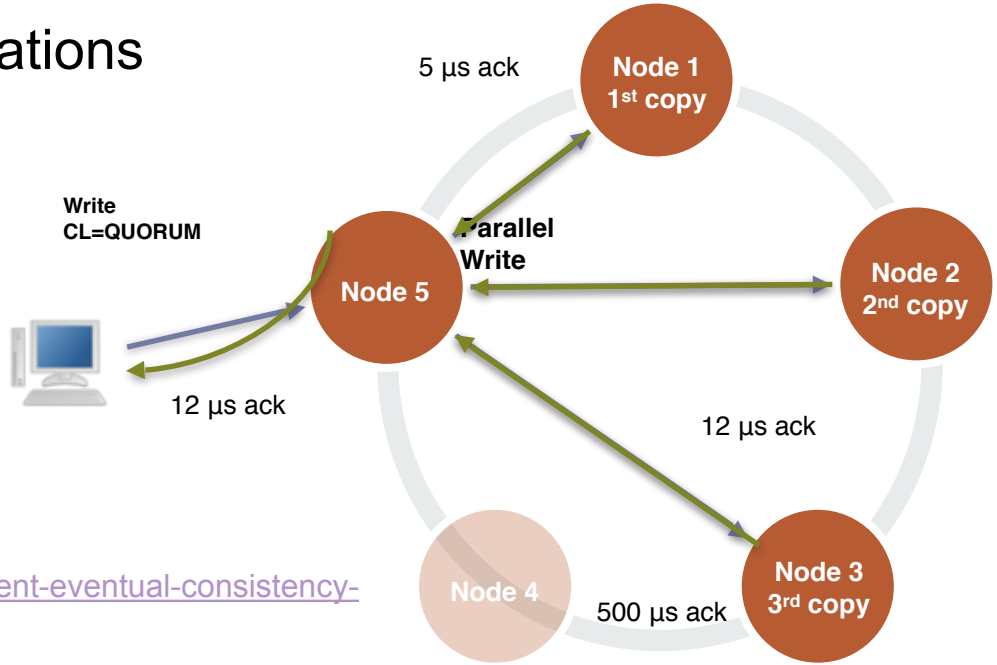
Cassandra - **Tuneable** Consistency

DATASTAX

- Consistency Level (CL)
- Client specifies per operation
- Handles multi-data center operations

- ALL = All replicas ack
- QUORUM = > 51% of replicas ack
- LOCAL_QUORUM = > 51% in **local DC** ack
- ONE = Only one replica acks
- Plus more…. (see docs)

- Blog: Eventual Consistency != Hopeful Consistency
  http://planetcassandra.org/blog/post/a-netflix-experiment-eventual-consistency-hopeful-consistency-by-christos-kalantzis/

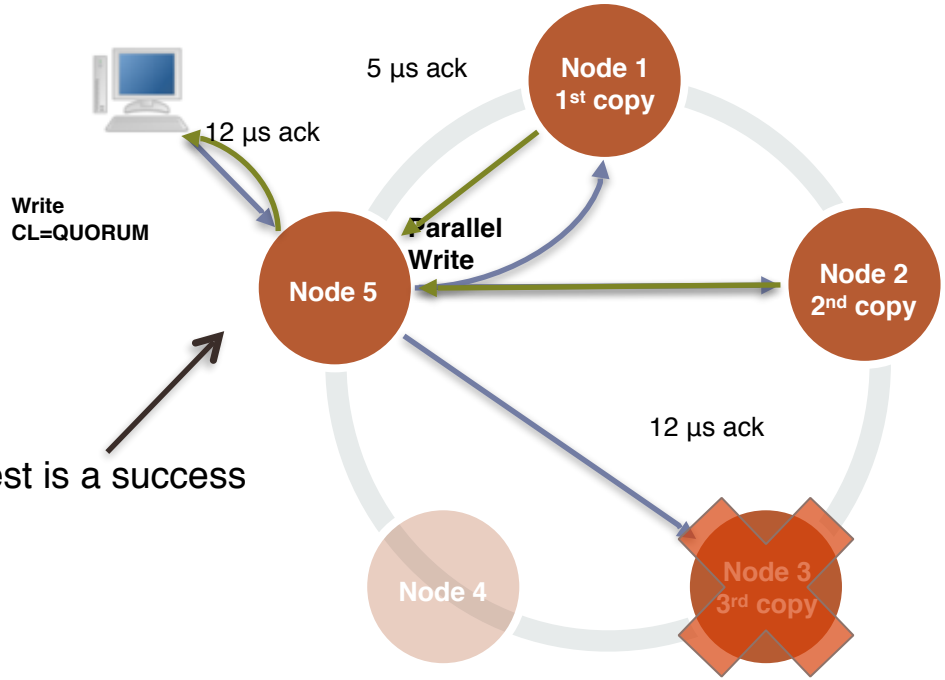


Cassandra - Node failure

DATASTAX

- **A single node failure shouldn't bring failure.**
- Replication Factor + Consistency Level = Success

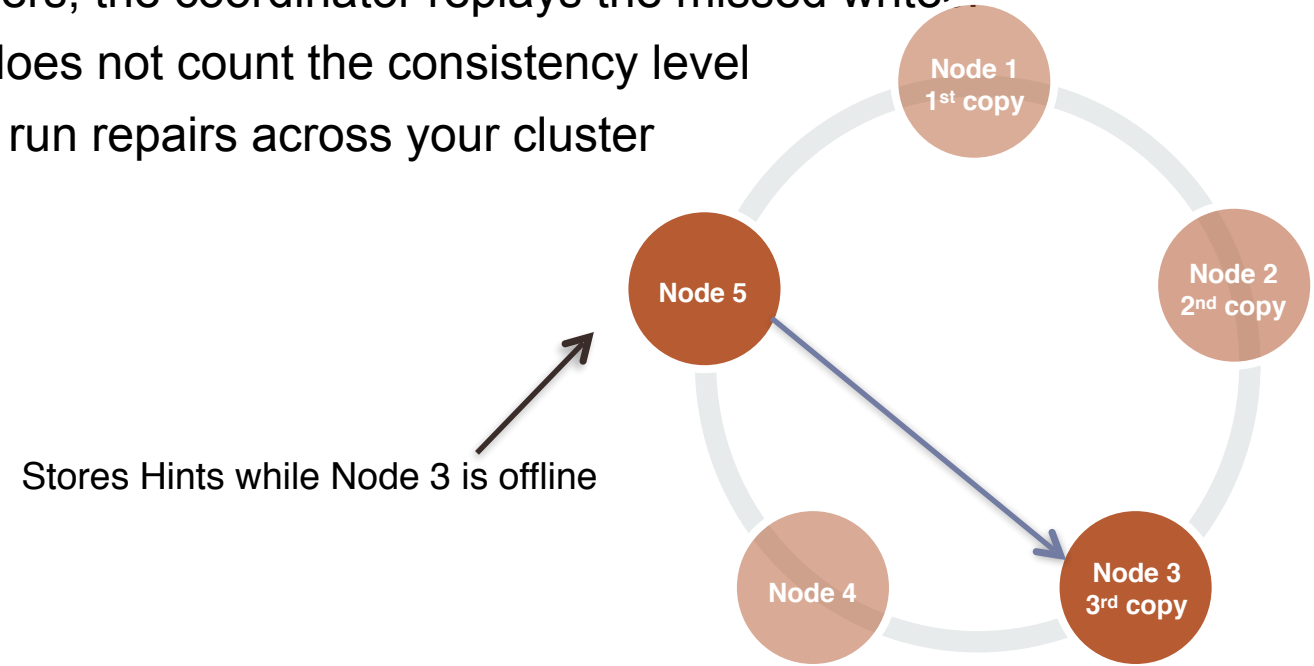
- This example:
  - RF = 3
  - CL = QUORUM



Cassandra - Node Recovery

DATASTAX

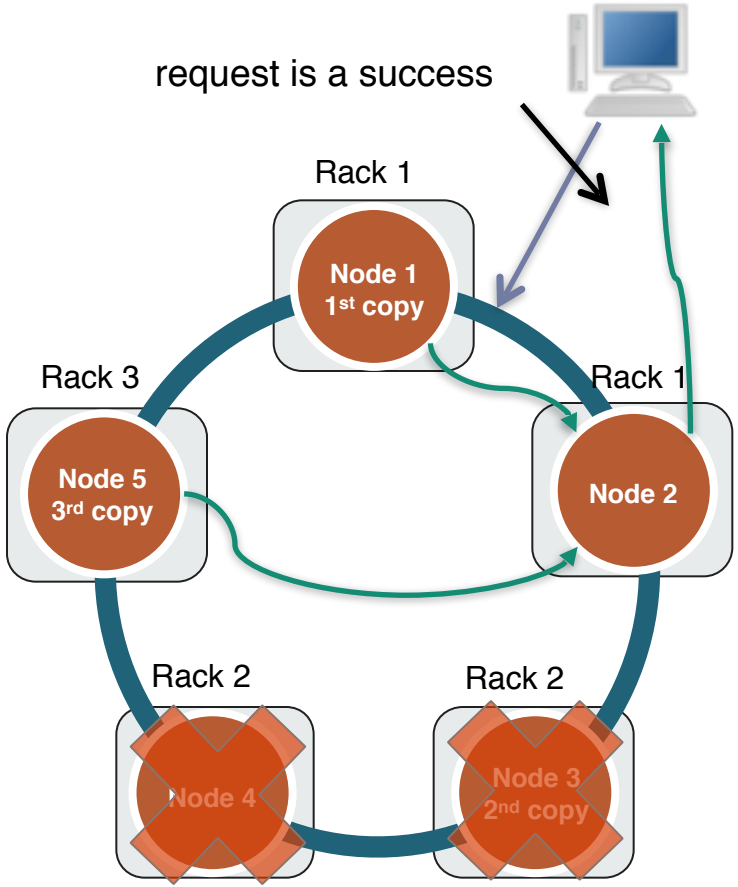
- When a write is performed and a replica node for the row is unavailable the coordinator will store a hint locally (3 hours)
- When the node recovers, the coordinator replays the missed writes
- **Note:** a hinted write does not count the consistency level
- **Note:** you should still run repairs across your cluster



Cassandra Rack/Zone Failure

DATASTAX

- Cassandra will place the data in as many different racks or availability zones as it can.
- This example:
  - RF = 3
  - CL = QUORUM
  - AZ/Rack 2 fails
- Data copies still available in Node 1 and Node 5
- Quorum can be honored i.e. **> 51% ack**



Cassandra is fast!

DATASTAX

- University of Toronto study:

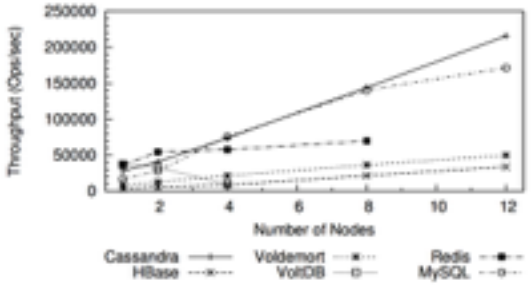


Figure 6: Throughput for Workload RW

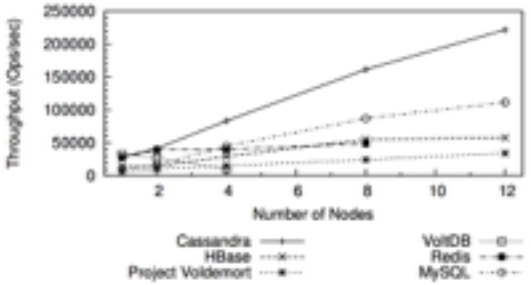


Figure 9: Throughput for Workload W

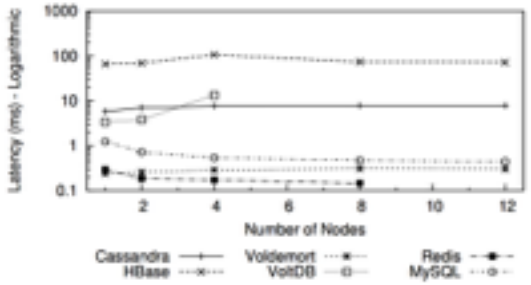


Figure 7: Read latency for Workload RW

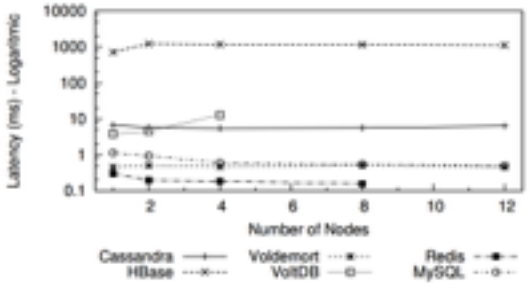
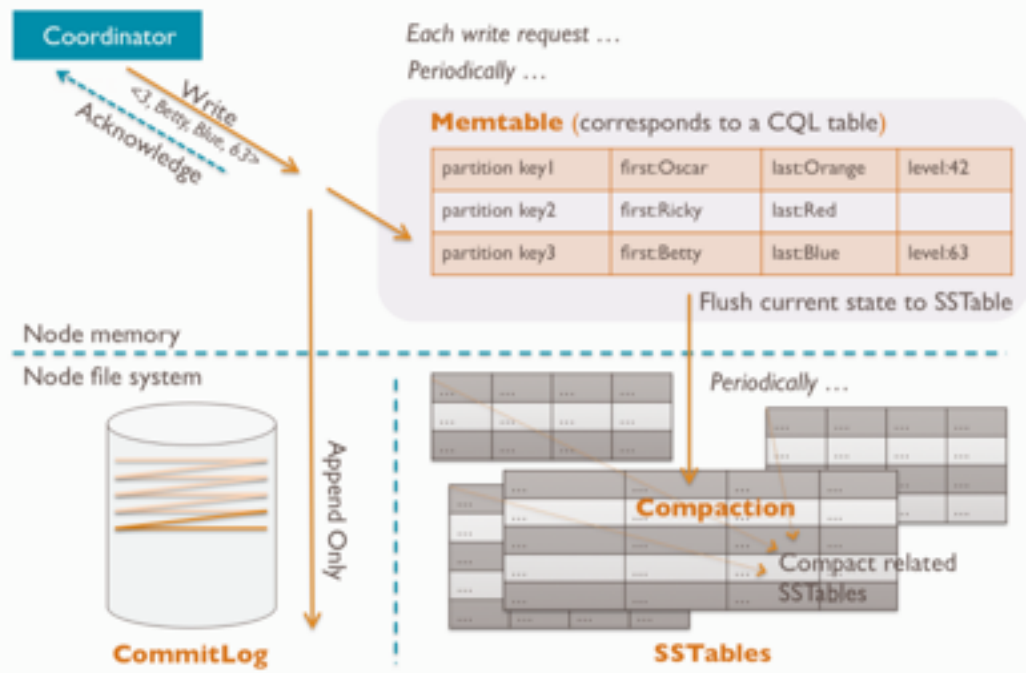


Figure 10: Read latency for Workload W

Why is Cassandra so fast?

- write-optimised - sequential writes to disk
- fast merging - when SSTable big enough merged with existing
- single layout on disk



Operational Simplicity

DATASTAX

- Cassandra is a complete product – there is not a multitude of components to install, set-up and monitor.
- Extremely simple to administer and deploy
- Backups are instantaneous and simple to restore
  - Supports snapshots, incremental backups and point-in-time recovery.
- Cassandra can handle non-uniform hardware and disks.
  - This enables the mixing of solid state and spinning disks in a single cluster and pinning tables to workload-appropriate disks.
- No downtime is required in Cassandra for upgrades or adding/removing servers from the cluster. Scale-Up and Scale-Out are easy to manage.

Cassandra Query Language

CQL

DATASTAX

- **C**assandra **Q**uery **L**anguage
- CQL is intended to provide a common, simpler and easier to use interface into Cassandra - and you probably already know it!
- **e.g. SELECT * FROM users**
- Usual statements:
  - CREATE / DROP / ALTER TABLE / SELECT

CQLSH

DATASTAX

- Command line interface comes with Cassandra
- Allows some other Statements

Command	Description
CAPTURE	Captures command output and appends it to a file
CONSISTENCY	Shows the current consistency level, or given a level, sets it
COPY	Imports and exports CSV (comma-separated values) data
DESCRIBE	Provides information about a Cassandra cluster or data objects
EXIT	Terminates cqlsh
SHOW	Shows the Cassandra version, host, or data type assumptions
SOURCE	Executes a file containing CQL statements
TRACING	Enables or disables request tracing

CQL Basics

DATASTAX

```
CREATE KEYSPACE league WITH REPLICATION = {‘class’:’NetworkTopologyStrategy’, ‘DataCentre1’:3, ‘DataCentre2’: 2};

USE league;

CREATE TABLE teams (
   team_name   varchar,
   player_name varchar,
   jersey      int,
   PRIMARY KEY (team_name, player_name)
);

SELECT * FROM teams WHERE team_name = ‘Mighty Mutts’ and player_name = ‘Lucky’;

INSERT INTO teams (team_name, player_name, jersey) VALUES ('Mighty Mutts','Felix’,90);
```

CQL Data Types

DATASTAX

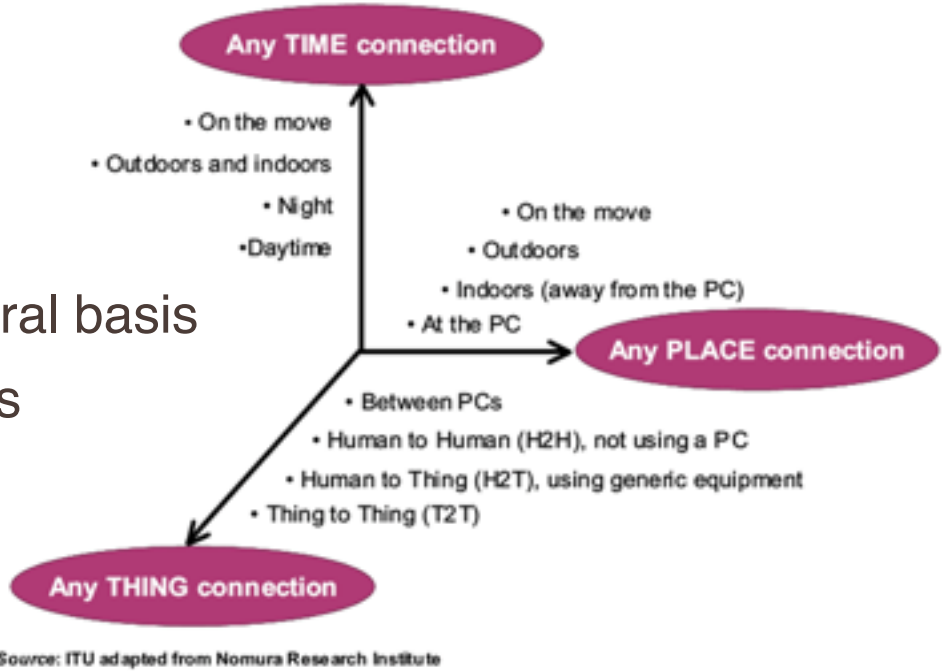
CQL Type	Constants	Description
ASCII	strings	US-ASCII character string
BIGINT	integers	64-bit signed long
BLOB	blobs	Arbitrary bytes (no validation), expressed as hexadecimal
BOOLEAN	booleans	true or false
COUNTER	integers	Distributed counter value (64-bit long)
DECIMAL	integers, floats	Variable-precision decimal
DOUBLE	integers	64-bit IEEE-754 floating point
FLOAT	integers, floats	32-bit IEEE-754 floating point
INET	strings	IP address string in IPv4 or IPv6 format*
INT	integers	32-bit signed integer
LIST	n/a	A collection of one or more ordered elements
MAP	n/a	A JSON-style array of literals: { literal : literal, literal : literal ... }
SET	n/a	A collection of one or more elements
TEXT	strings	UTF-8 encoded string
TIMESTAMP	integers, strings	Date plus time, encoded as 8 bytes since epoch
UUID	uuids	A UUID in standard UUID format
TIMEUUID	uuids	Type 1 UUID only (CQL 3)
VARCHAR	strings	UTF-8 encoded string
VARINT	integers	Arbitrary-precision integer

Sensor/Time Data - Data Model

It´s about the data

DATASTAX

- Sensors
  - CPU, Network Card, Electronic Power Meter, Resource Utilization, Weather
- Clickstream data, WebAnalytics
- Historical trends
  - Stock Ticker
- Anything that varies on a temporal basis
  - Top Ten Most Popular Videos

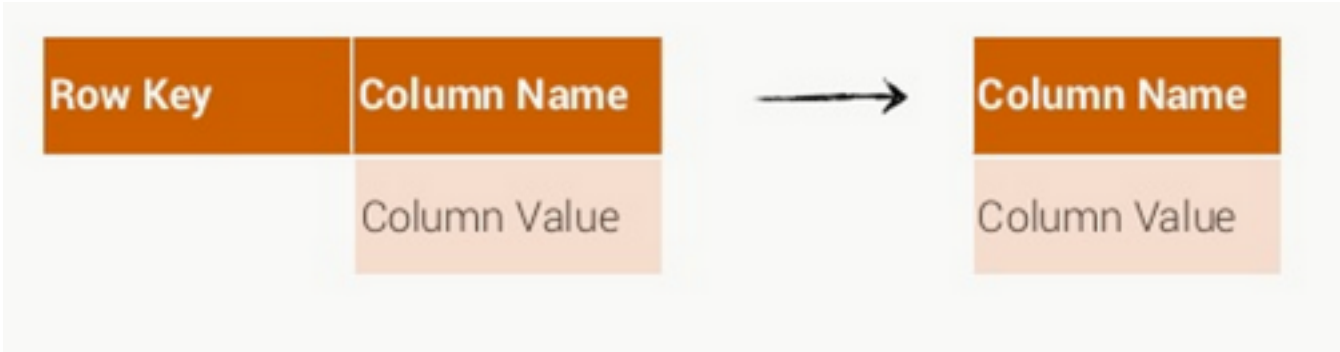


Source: ITU adapted from Nomura Research Institute

Why Cassandra for time series data

DATASTAX

- Cassandra is based on BigTable storage model
  - One key row and lots of (variable) columns
  - Single layout on disk
  - Cassandra works very well with data in sequence



Time series - table definition

DATASTAX

- Data partitioned by weather station ID and time
  - WeatherStationID is PRIMARY KEY (CQL) = PARTITION KEY (Cassandra), event_time is Clustering Column (Together = Compound Primary Key)
  - Clustering determines clustering (storage process that creates index and keeps data in order based on the index)
  - When rows for a partition key are stored in order based on clustering columns retrieval is very efficient

```
CREATE TABLE temperature (
   weatherstation_id text,
   event_time timestamp,
   temperature text,
   PRIMARY KEY (weatherstation_id,event_time)
);
```

Time series - example

DATASTAX

- Storing weather data, One weather station
- Temperature measurement every minute
- Retrieving data by row key (WeatherStationID) and column key (event_time) is efficient

Time series - INSERT and QUERY

DATASTAX

- Inserts are simple and easy

```
INSERT INTO
temperature(weatherstation_id,event_time,temperature)
VALUES ('1234ABCD','2013-04-03 07:01:00','72F');
```

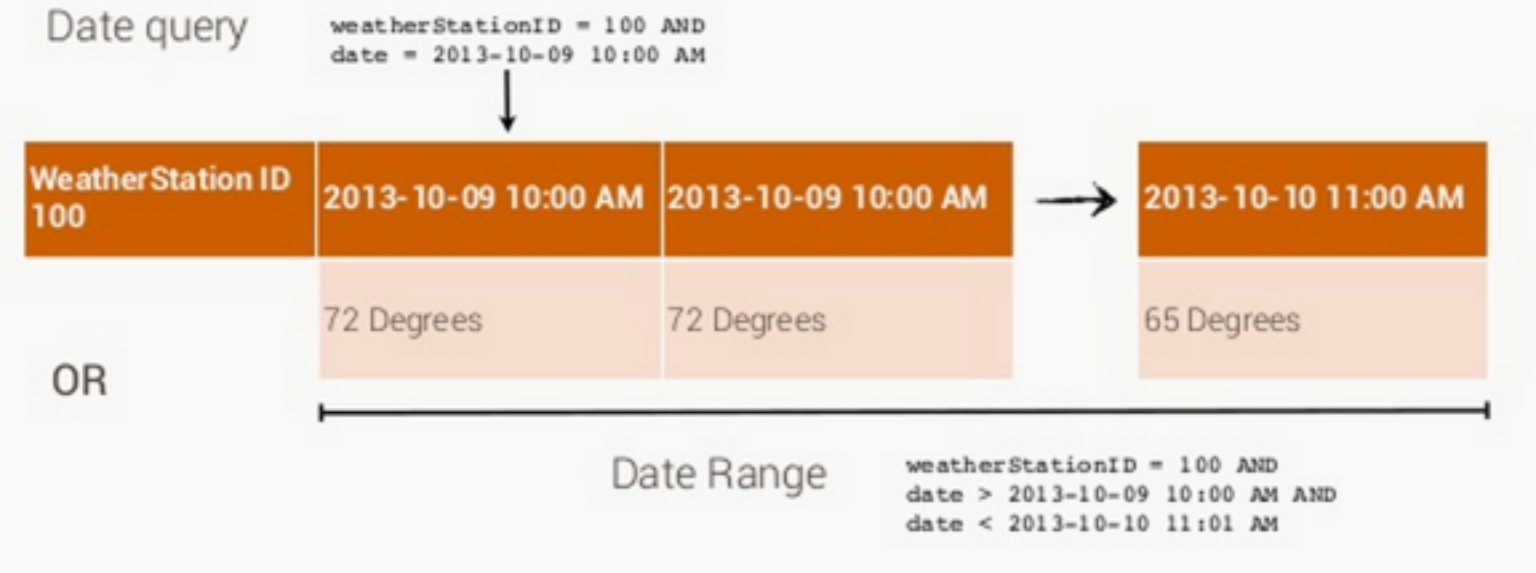
- Row can be retrieved by Row Key
- Column Value can be retrieved by Row Key and Column Key
- WHERE Statement possible on Primary Key and Indexed Columns (event_time)

```
SELECT event_time,temperature
FROM temperature
WHERE weatherstation_id='1234ABCD';
```

Time series - query data

DATASTAX

- Queries based on Date and Date ranges are easy



Time series - Partitioning

DATASTAX

- With the previous table, you can end up with a very large row on 1 partition i.e. (per millisecond for example)
- This would have to fit on 1 node, Cassandra can store 2 billion columns per storage row (on one node reads = hotspots)
- The solution is to have a **composite** Partition Key **(date)** to split things up:

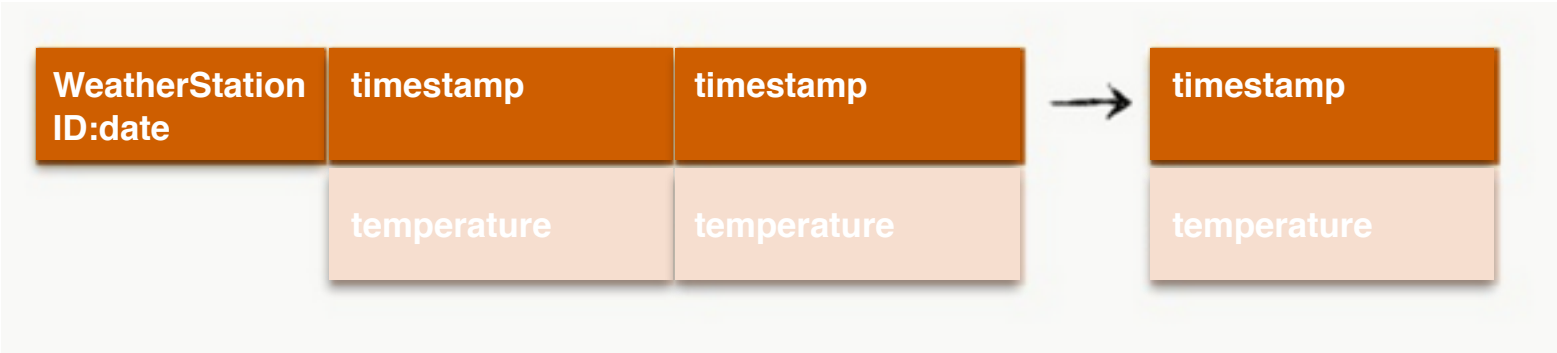
```
CREATE TABLE temperature_by_day (
   weatherstation_id text,
   date text,
   event_time timestamp,
   temperature text,
   PRIMARY KEY ((weatherstation_id,date),event_time)
);
```

Time series - Partitioning

DATASTAX

- Using date (portion of timestamp) as available value
- Query all data from a single day

```
SELECT *
FROM temperature_by_day
WHERE weatherstation_id='1234ABCD'
AND date='2013-04-03';
```



Data Modeling

DATASTAX

- Any questions?
- Feel free to learn more about data modeling online:

Part 1: The Data Model is Dead, Long Live the Data Model
http://www.youtube.com/watch?v=px6U2n74q3g

Part 2: Become a Super Modeler
http://www.youtube.com/watch?v=qphhxujn5Es

Part 3: The World's Next Top Data Model
http://www.youtube.com/watch?v=HdJlsOZVGwM

What ´s up with DataStax?

DataStax at a glance

DATASTAX

370+
Employees

~27
Percent

600+
Customers

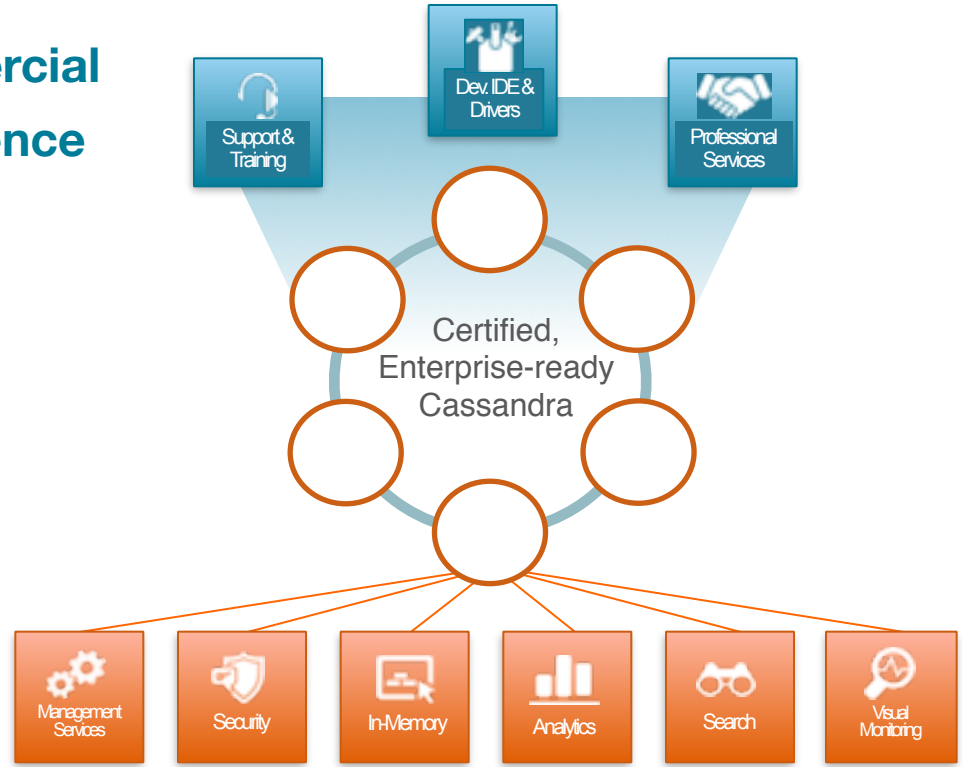
Founded in April 2010

Santa Clara, Austin, New York, London, Sydney, Paris

FORTUNE 100

DATASTAX

Commercial Confidence

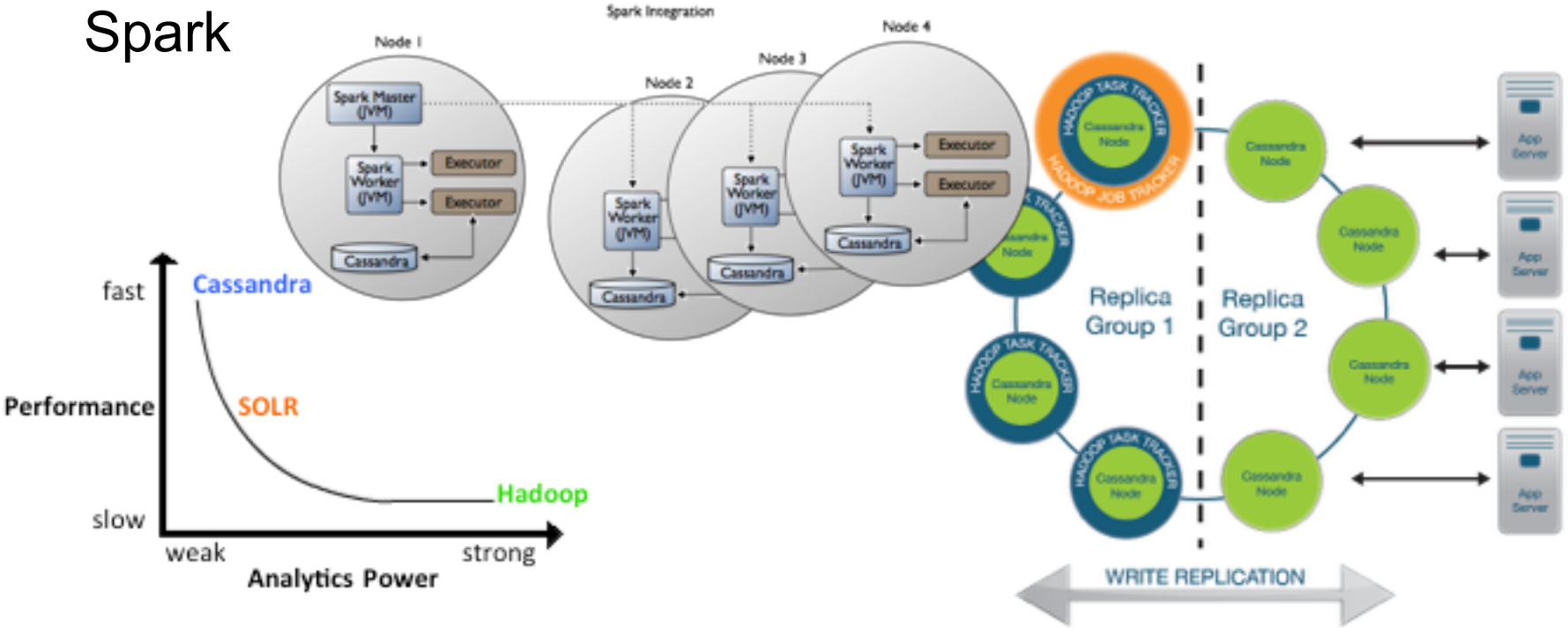


Enterprise Functionality

Enterprise Integrations

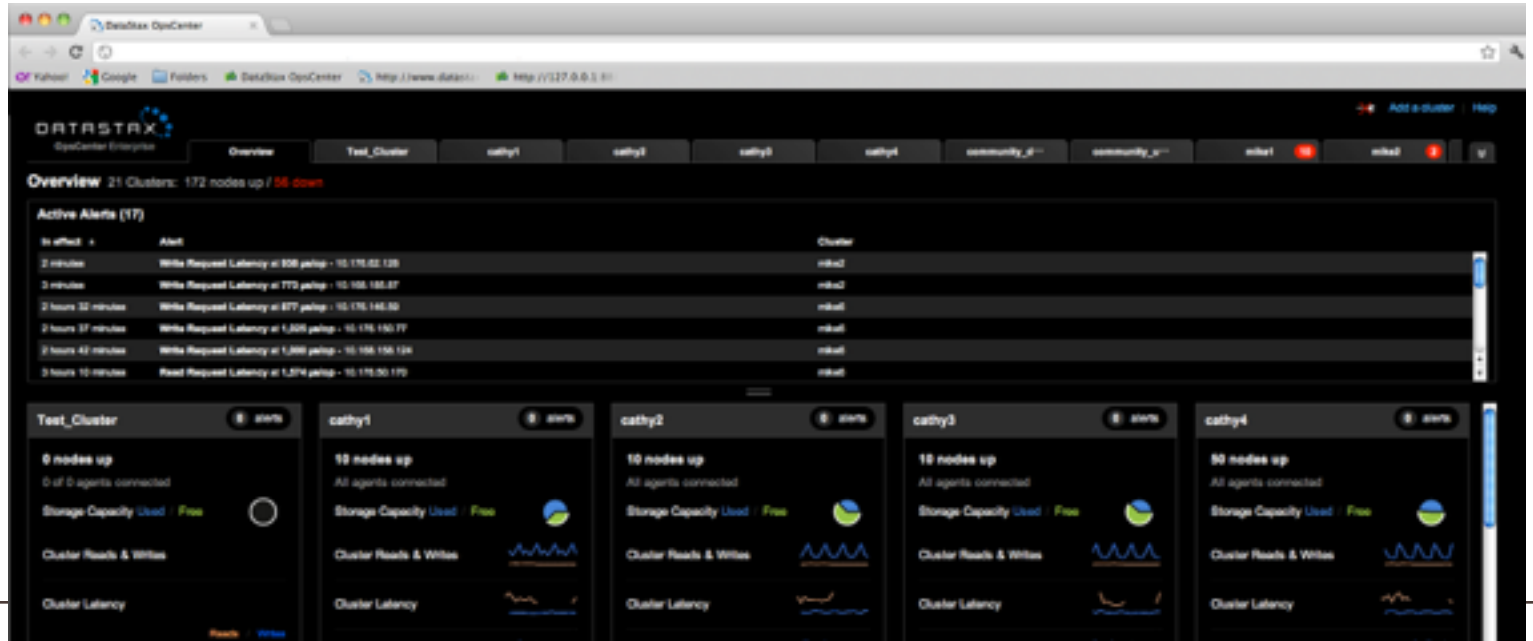
DATASTAX

- DataStax adds Enterprise Features like: Hadoop, Solr, Spark



OpsCenter 5.0

- OpsCenter is a browser-based, visual management and monitoring solution for Apache Cassandra and DataStax Enterprise
- Functionality is also exposed via HTTP APIs

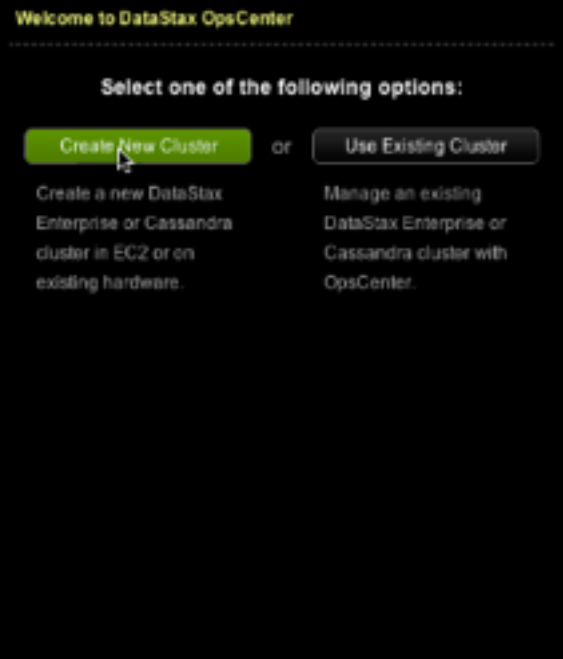


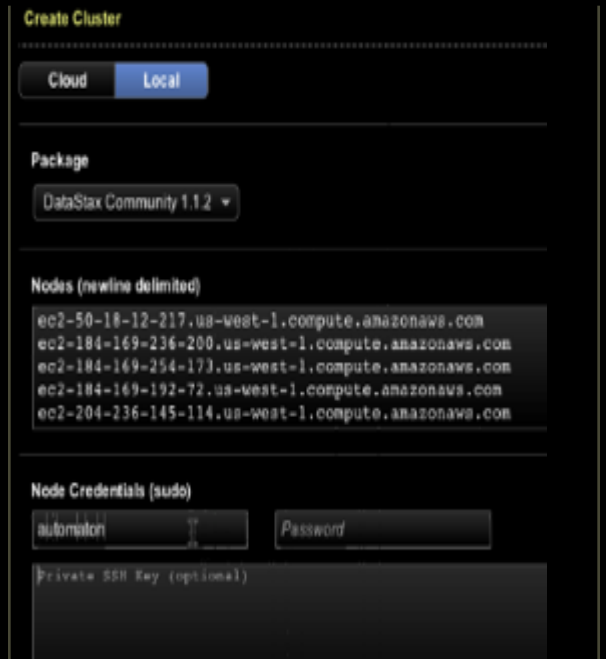
OpsCenter - New Cluster Example

DATASTAX

A new, 10-node DSE cluster with OpsCenter running on AWS in **3 minutes**…

1 → 2 → 3 **Done**





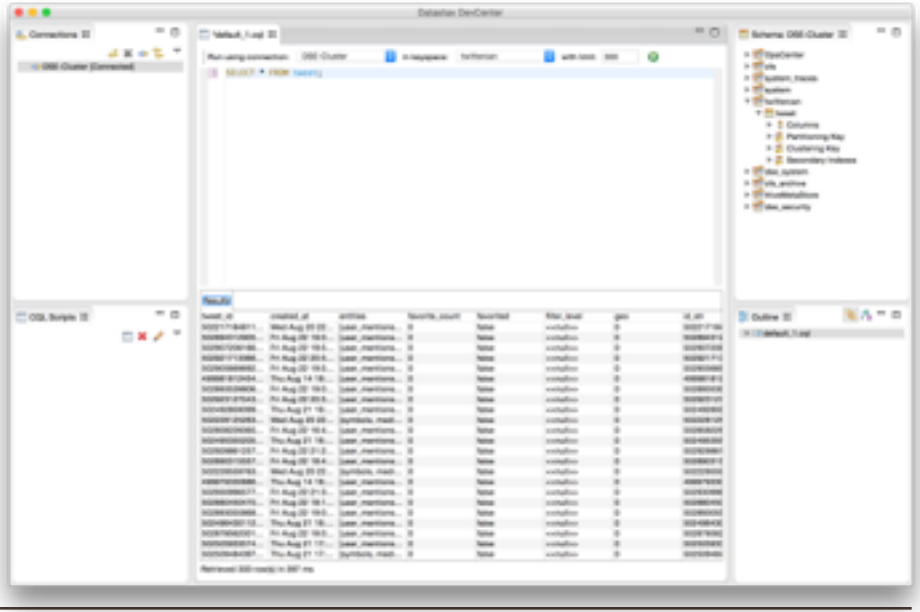
OpsCenter 5.0

- Manage multiple clusters and nodes
- Add and remove nodes
- Administer individual nodes or in bulk
- Configure clusters
- Perform rolling restarts
- Automatically repair data
- Rebalance data
- Backup management
- Capacity planning

DevCenter 1.1

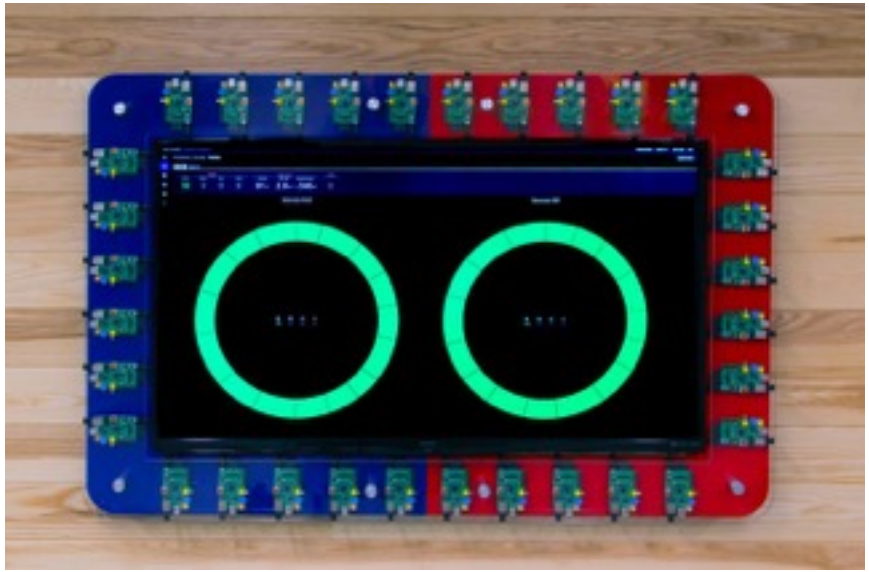
DATASTAX

- Visual Query Tool for Developers and Administrators
- Easily create and run Cassandra Queries
- Visually navigate database objects
- Context-based suggestions



DataStax Office Demo

- 32 Raspberry Pi´s
- 16 per DataStax Enterprise 4.5 Cluster
- Managed in OpsCenter 5.0
- “Red Button” downs one DataCenter
- Not the Performance-Demo but
  - Availability
  - Commodity Hardware



Native Drivers

DATASTAX

- Different Native Drivers available: Java, Python etc.
  - Load Balancing Policies (Client Driver receives Updates)
  - Data Centre Aware
  - Latency Aware
  - Token Aware
- Reconnection policies
- Retry policies
- Downgrading Consistency
- Plus others..

- http://www.datastax.com/download/clientdrivers

DataStax Enterprise

DATASTAX

Feature	Open Source	Datastax Enterprise
Database Software		
Data Platform	Latest Community Cassandra	**Production Certified Cassandra**
Core security features	Yes	**Yes**
Enterprise security features	No	**Yes**
Built-in automatic management services	No	**Yes**
Integrated analytics	No	**Yes**
Integrated enterprise search	No	**Yes**
Workload/Workflow Isolation	No	**Yes**
Easy migration of RDBMS and log data	No	**Yes**
Certified Service Packs	No	**Yes**
Certified platform support	No	**Yes**
Management Software		
OpsCenter	Basic functionality	**Advanced functionality**
Services		
Community Support	Yes	**Yes**
Datastax 24x7x365 Support	No	**Yes**
Quarterly Performance Reviews	No	**Yes**
Hot Fixes	No	**Yes**
Bug Escalation Privilege	No	**Yes**
Custom Builds	No	**Option**
EOL Support	No	**Yes**
Licensing	Free	**Subscription**

DataStax Comparison

DATASTAX

	Standard	Pro	Max
Server Data Management Components			
Production-certified Cassandra	Yes	Yes	Yes
Advanced security option	Yes	Yes	Yes
Repair service	Yes	Yes	Yes
Capacity planning service	Yes	Yes	Yes
Enterprise search (built-in Solr)	No	Yes	Yes
Analytics (built-in Hadoop)	No	No	Yes
Management Tools			
OpsCenter Enterprise	Yes	Yes	Yes
Support Services			
Expert Support	24x7x1	24x7x1	24x7x1
Partner Development Support	Business	Business hours	Business
Certified service packs	Yes	Yes	Yes
Hot fixes	Yes	Yes	Yes
Bug escalation	Yes	Yes	Yes
Quarterly performance reviews	No	No	Yes
Bi-weekly call with support team	No	No	Yes
Custom builds	No	No	Option

Netflix Delights Customers with Personal Recommendations

World's leading streaming media provider with digital revenue $1.5BN+
Tailors content delivery based on viewing preference data captured in Cassandra
Increased market cap by 600% since 2012
Introduction of 'Profiles' drove throughput to over 10M transactions per second
Replaced Oracle in six data centers, worldwide, 100% in the cloud

NETFLIX

Use Case: Personalization

Christos Kalantzis
@chriskalan

Follow

Just completed our #Cassandra migration to DSE. 80+ clusters, 2500+ nodes done by 2 guys. Want to be one of them? #Netflix

Reply Retweet Favorite ••• More

RETWEETS	FAVORITES
114	70

9:27 AM - 23 May 2014

Comcast Invents the “Future of Awesome”

Future of TV: cloud-based X1 platform connects viewers with more content
App messaging to track your favorite team’s score while watching a movie, or TV show
DVR scheduling, recording, playback
Playlists and personalized recommendations
SOLID project - centralized NoSQL delivery

comcast

Use Case: Personalization

DATASTAX

The Weather Channel

*"If you had a look in the past, you may have found Cassandra had a high learning curve and a fair amount of complexity. CQL3, the native drivers, and virtual nodes have **changed the game entirely**, making Cassandra a much more accessible and friendly platform.*

*While I have years of experience using Cassandra, my team was mostly new to it; **CQL made their transition essentially painless**. But where Cassandra really shines is in **speed and operational simplicity**, and I would say those two points were critical."*

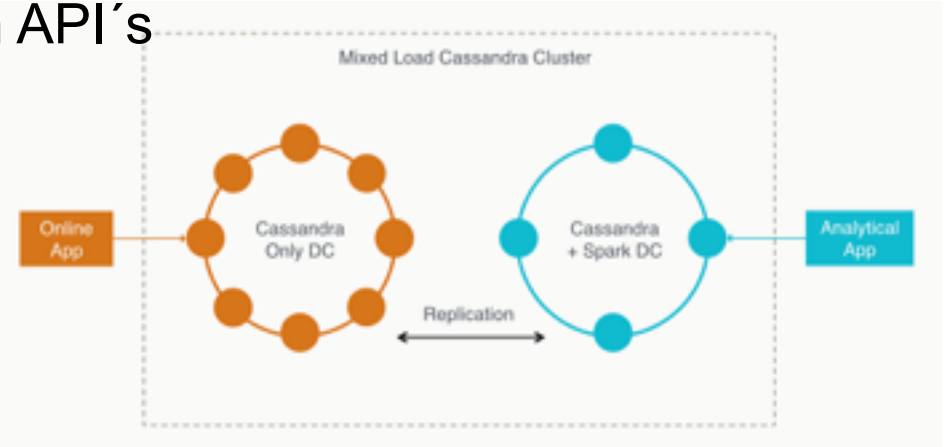
ROBBIE STRICKLAND *Software Dev Manager*

Realtime Analytics

What is Spark?

DATASTAX

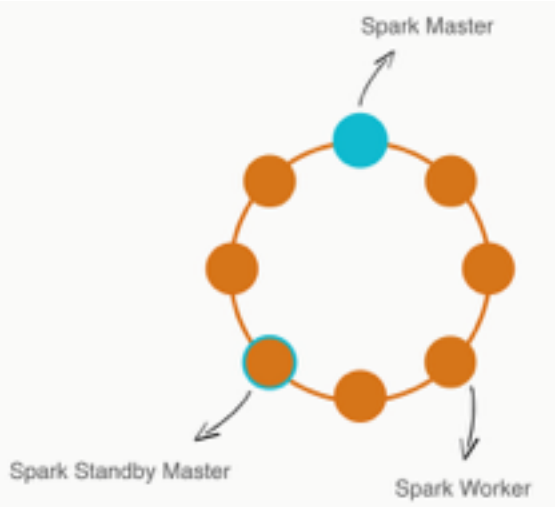
- Apache Project since 2010 - Analytics Framework
- 10-100x faster than Hadoop MapReduce
- In-Memory Storage for Read&Write data
- Single JVM Processor per node
- Rich Scala, Java and Python API´s
- 2x-5x less code
- Interactive Shell



Why Spark on Cassandra?

DATASTAX

- Data model independent queries
- cross-table operations (JOIN, UNION, etc.)!
- complex analytics (e.g. machine learning)
- data transformation, aggregation etc.
- stream processing (coming soon)
- all nodes are Spark workers
- by default resilient to worker failures
- first node promoted as Spark Master
- Standby Master promoted on failure
- Master HA available in Datastax Enterprise



How to Spark on Cassandra?

DATASTAX

- DataStax Cassandra Spark driver
  - OpenSource: https://github.com/datastax/cassandra-driver-spark

- Compatible with
  - Spark 0.9+
  - Cassandra 2.0+
  - DataStax Enterprise 4.5+

What´s new?!

2.1 Release - User Defined Types

DATASTAX

```
CREATE TYPE address (
  street text,
  city text,
  zip_code int,
  phones set<text>
)

CREATE TABLE users (
  id uuid PRIMARY KEY,
  name text,
  addresses map<text, address>
)

SELECT id, name, addresses.city, addresses.phones FROM users;

       id |    name | addresses.city |         addresses.phones
----------+---------+----------------+--------------------------
 63bf691f | chris   |         Berlin | {'0201234567', '0796622222'}
```

2.1 Release - Secondary Indexes on collections

DATASTAX

```
CREATE TABLE songs (
  id uuid PRIMARY KEY,
  artist text,
  album text,
  title text,
  data blob,
  tags set<text>
);

CREATE INDEX song_tags_idx ON songs(tags);

SELECT * FROM songs WHERE tags CONTAINS 'blues';

 id       | album         | artist            | tags                  | title
----------+---------------+-------------------+-----------------------+-----------------
 5027b27e | Country Blues | Lightnin' Hopkins | {'acoustic', 'blues'} | Worrying My Mind
```

How to start in production?

DATASTAX

- DataStax Enterprise or Community
- Hardware:
  - min. 8GB RAM - optimal price-performance sweet spot is 16GB to 64GB
  - 8-Core CPU - Cassandra is so efficient in writing that the CPU is the limiting factor
  - SSD-Disks - Commitlog + 50% Compaction and ext3/4 or xfs file-system
- Nodes - Cluster recommendation is 3 nodes as minimum

- Alternative: Use the Amazon Images (http://www.datastax.com/documentation/cassandra/2.0/cassandra/architecture/architecturePlanningEC2_c.html)

Thanks! Let´s see a demo!