Introduction to Apache Cassandra

Jon Haddad, Luke Tillman

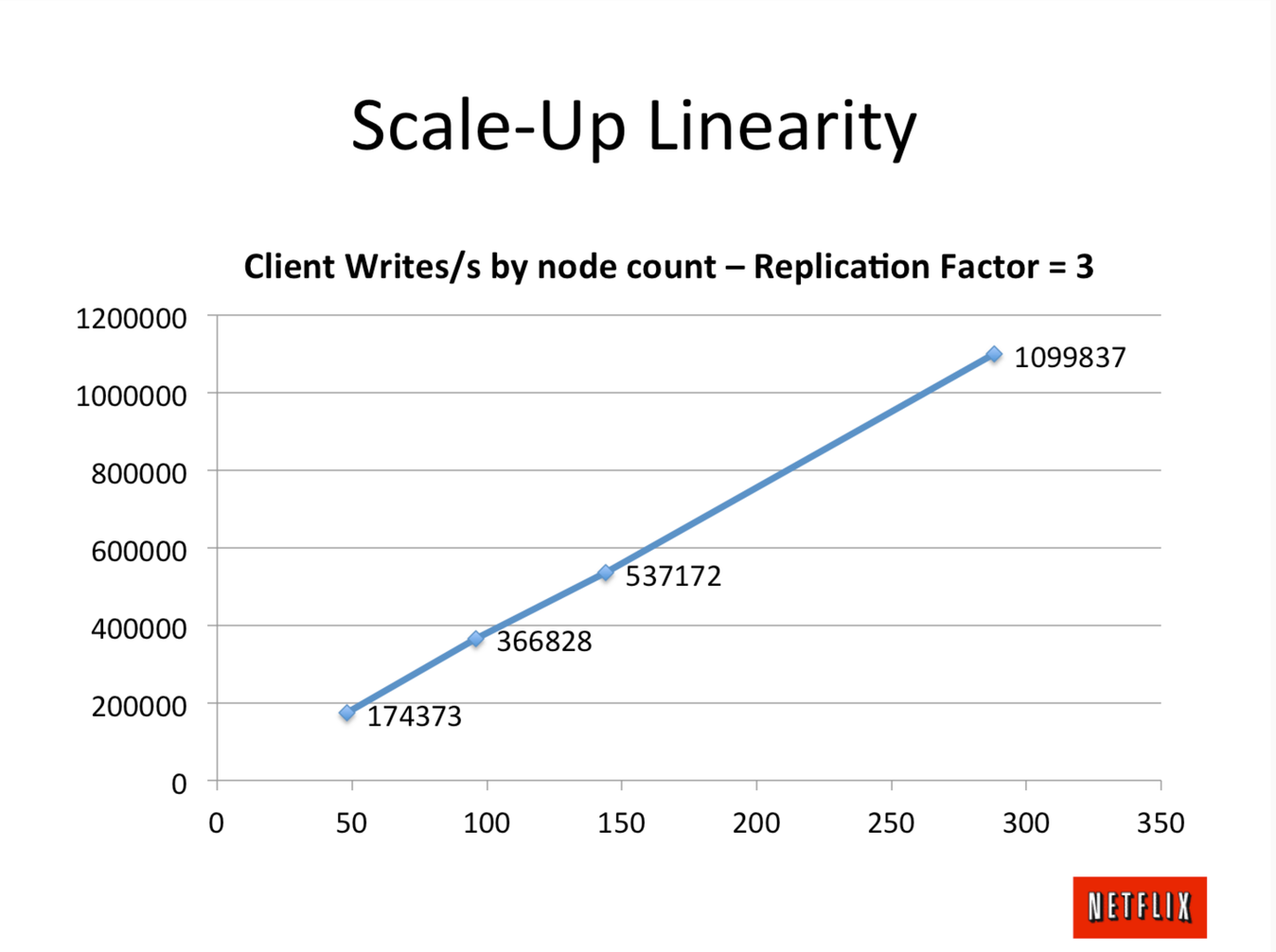
Technical Evangelists, DataStax

@rustyrazorblade, @LukeTillman

What is Apache Cassandra?

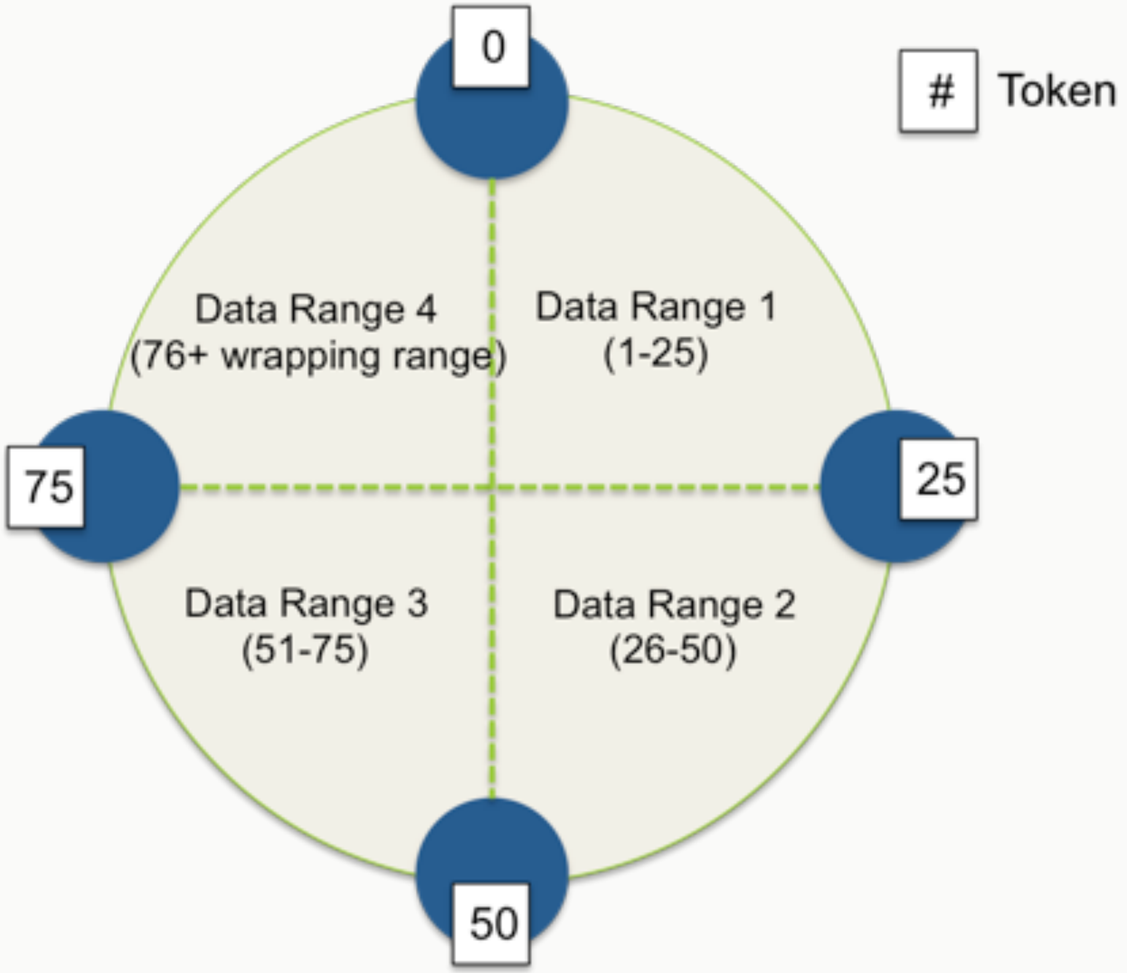
DATASTAX

- Fast Distributed Database
- High Availability
- Linear Scalability
- Predictable Performance
- No SPOF
- Multi-DC
- Commodity Hardware
- Easy to manage operationally



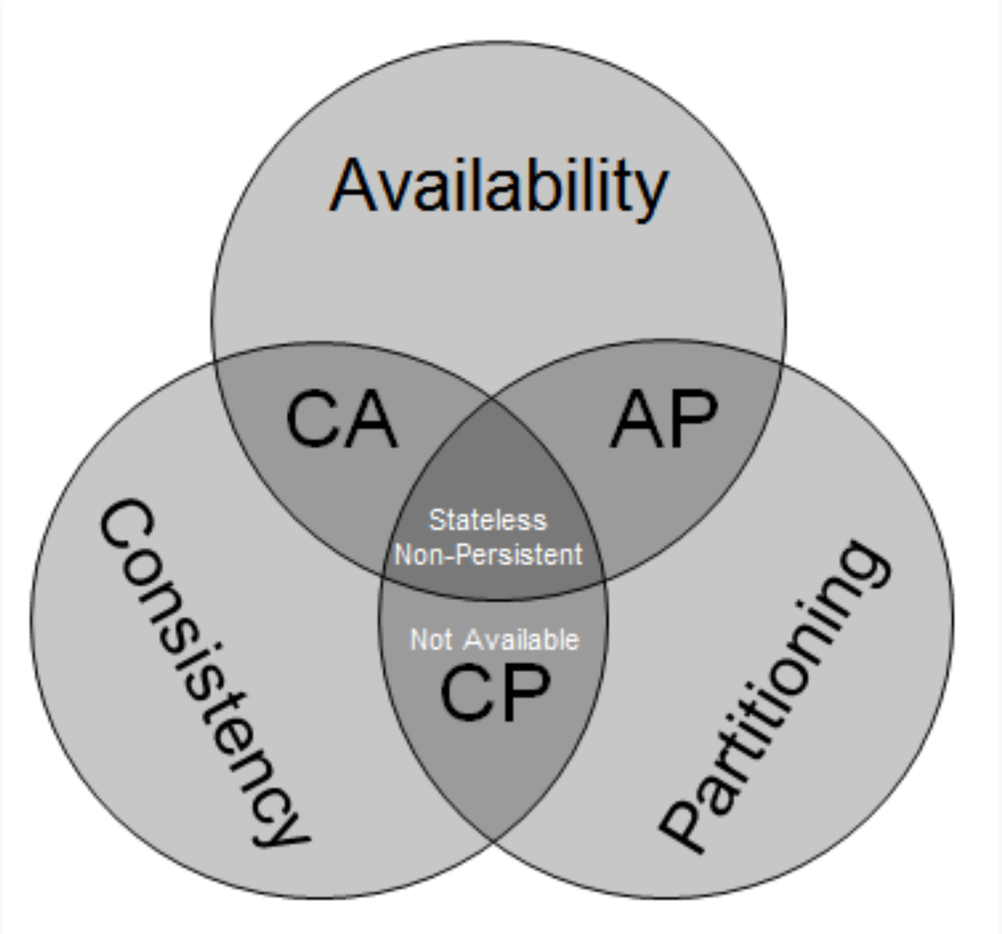
Hash Ring

- No master / slave / replica sets
- No config servers, zookeeper
- Data is partitioned around the ring
- Data is replicated to RF=N servers
- All nodes hold data and can answer queries (both reads & writes)
- Location of data on ring is determined by partition key



CAP Tradeoffs

- Cassandra chooses Availability & Partition Tolerance over Consistency
- Queries have tunable consistency level
- ALL, QUORUM, ONE
- Hinted Handoff to deal with failed nodes



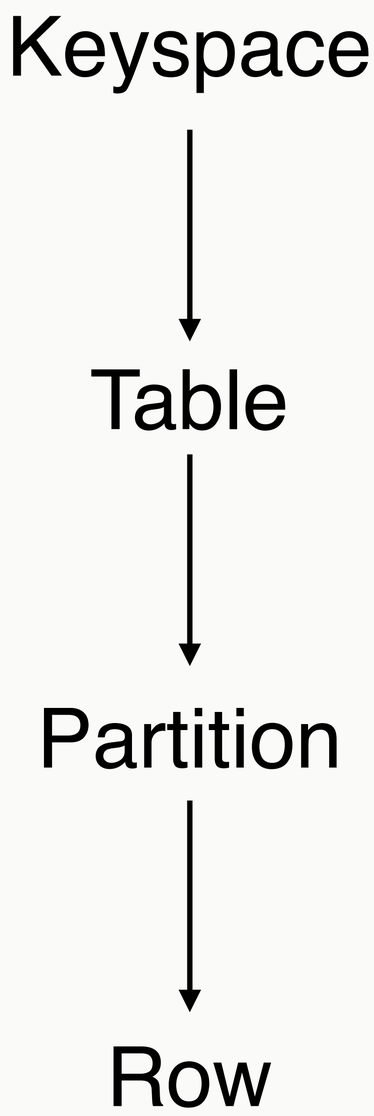
DATASTAX

Data Modeling

Data Structures

DATASTAX

- Like an RDBMS, Cassandra uses a Table to store data
- But there's where the similarities end
- Partitions within tables
- Rows within partitions (or a single row)
- CQL to create tables & query data
- Partition keys determine where a partition is found
- Clustering keys determine ordering of rows within a partition



Example: Single Row Partition

DATASTAX

- Simple User system
- Identified by name (pk)
- 1 Row per partition
- This is familiar territory

```
cqlsh:demo> create table user
(name text primary key,
age int,
job text);
```

name	age	job
jon	33	evangelist
luke	33	evangelist
old pete	108	retired
s. seagal	62	actor
JCVD	53	actor

```
cqlsh:demo> select * from user WHERE name = 'JCVD'
```

Example: Multiple Rows

DATASTAX

- Comments on photos
- Comments are always selected by the photo_id
- There are only 4 rows in 2 partitions
- In the real world, use UUIDs instead of int for PK

```
create table comment
( photo_id int,
comment_id int,
user text,
comment text,
primary key (photo_id, comment_id));
```

photo_id	comment_id	user	comment
5	1	jon	hi
5	2	luke	oh hey
5	3	JCVD	AHHHHH!!!
6	4	jon	great pic

```
select * from comment where photo_id=5
```

Partition with Clustering

DATASTAX

- Multiple rows are transposed into a single partition
- Partitions vary in size
- Old terminology - "wide row"

photo_id	comment_id	user	comment	comment_id	user	comment	comment_id	user	comment
5	1	jon	hi	2	luke	oh hey	3	JCVD	AHHHHH!!!
6	4	jon	great pic						

Model Tables to Answer Queries

DATASTAX

- This is not 3NF!!
- We always query by partition key
- Create many tables aka materialized views
- Manage in your app code
- Denormalize!!

```
CREATE TABLE age_to_user (
  age int,
  user text,
  primary key (age, user)
);
```

user	age
jon	33
luke	33
JCVD	53

age	user	user
33	jon	luke
53	JCVD	

CQL Data Types

DATASTAX

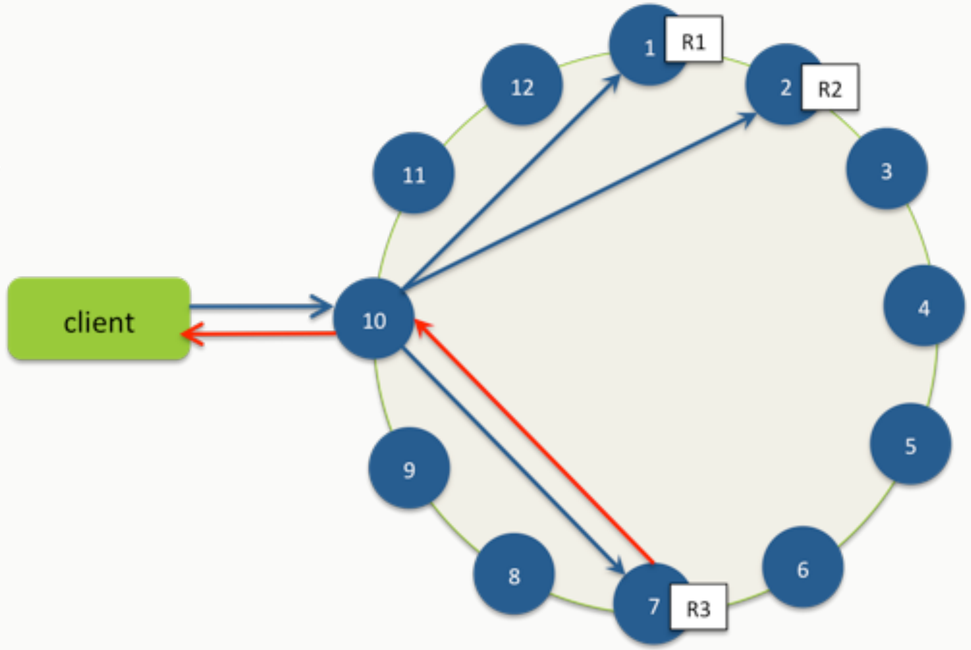
Basic Types			Collections
text	uuid	counter	map
int	timeuuid		list
decimal			set
blob			

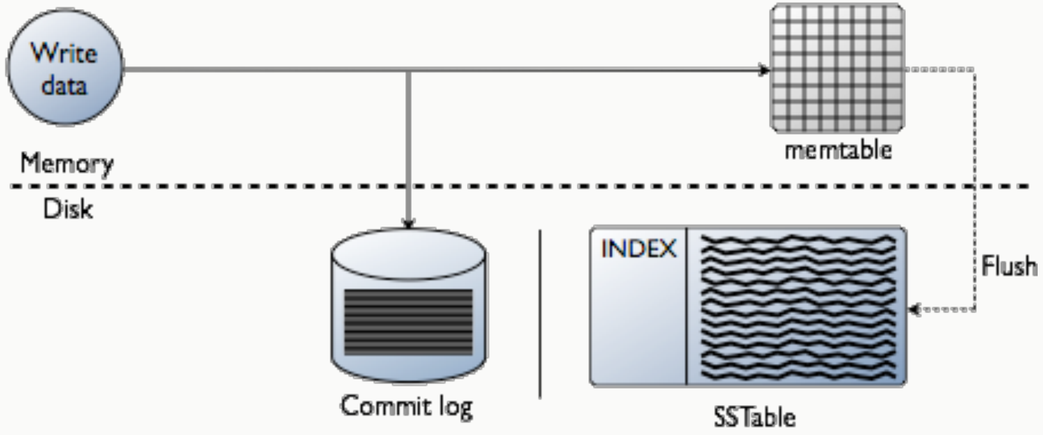
Read the CQL documentation for the full list of types

Reads & Writes

The Write Path

- Writes are written to any node in the cluster (coordinator)
- Writes are written to commit log, then to memtable
- Every write includes a timestamp
- Memtable flushed to disk periodically (sstable)
- New memtable is created in memory
- Deletes are actually a special write case, called a “tombstone”

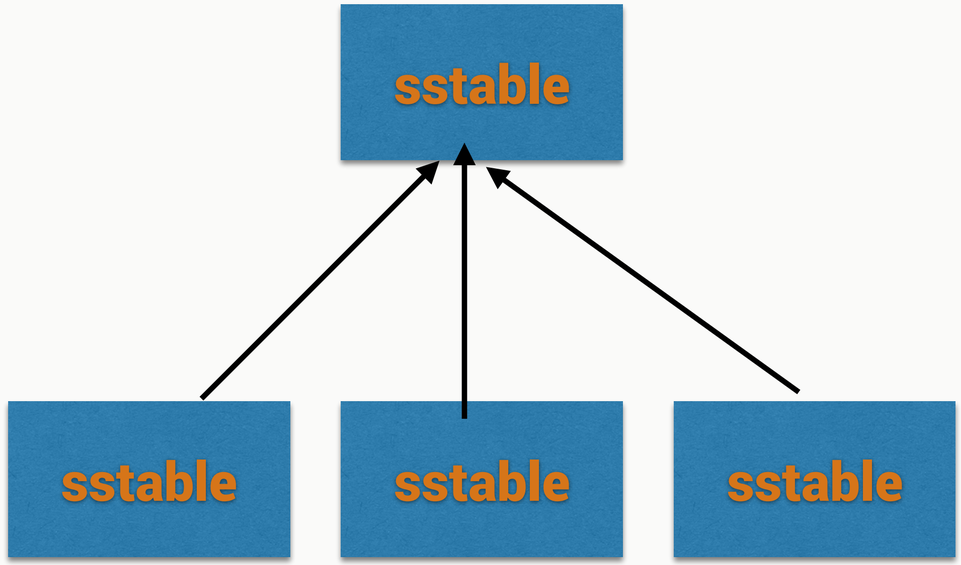




What is an SSTable?

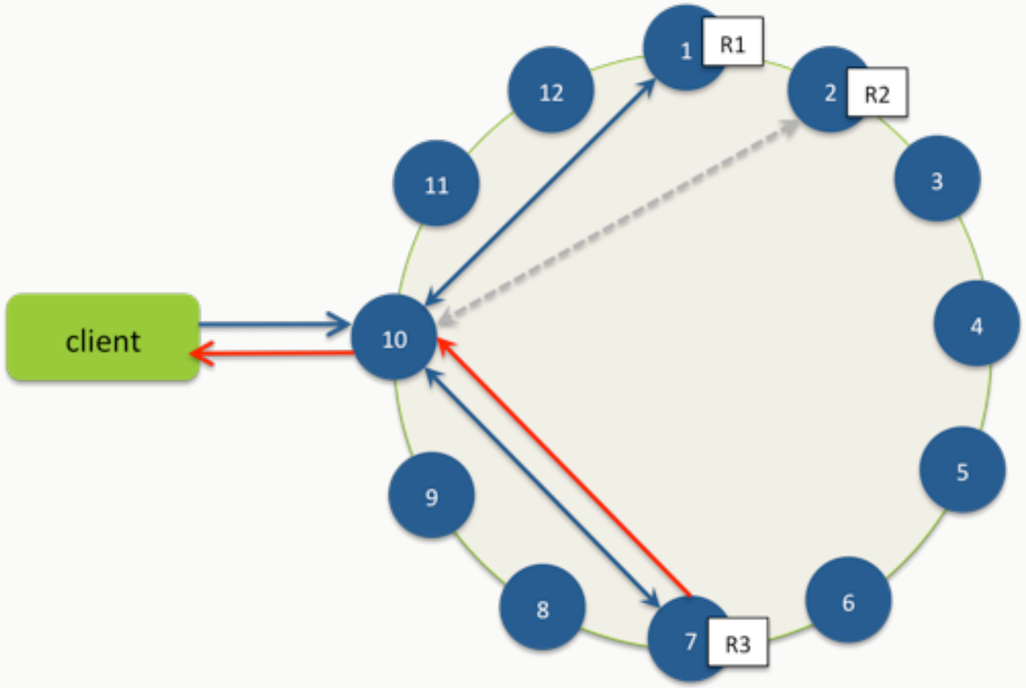
DATASTAX

- Immutable data file for row storage
- Deletes are written as tombstones
- Every write includes a timestamp of when it was written
- Partition is spread across multiple SSTables
- Same column can be in multiple SSTables
- Merged through compaction, only latest timestamp is kept
- Easy backups!



The Read Path

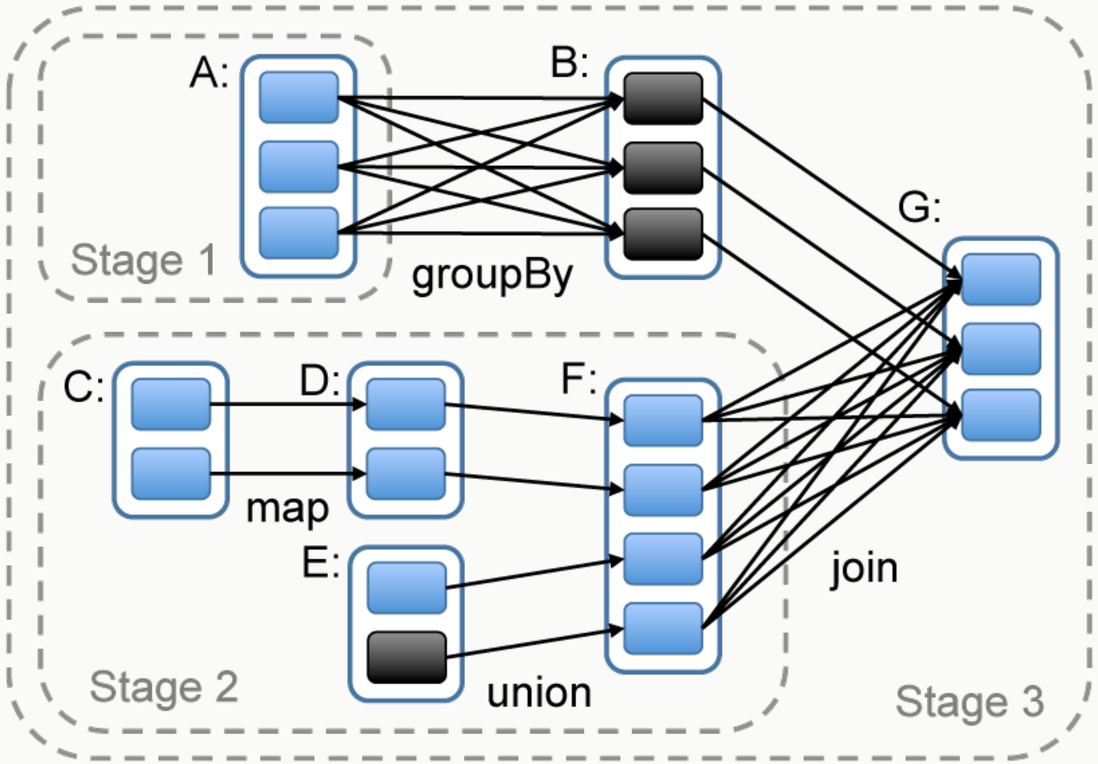
- Any server may be queried, it acts as the coordinator
- Contacts nodes with the requested key
- On each node, data is pulled from SSTables and merged
- Consistency< ALL performs read repair in background (read_repair_chance)



Analytics with Spark

Spark at a Glance

- Scala, Python, Java
- Hadoop alternative - batch analytics
- Distributed SQL
- Real time analytics via streaming
- Machine learning
- GraphX (in progress)
- Open source connector available
- Built into DSE



Summary

- How do I query my data if I can only query by key?
- Denormalize!
- Create multiple views into your data (multiple tables)
- Cassandra is built for fast writes
- Use fast writes to do as few reads as possible
- Use Spark for advanced analytics and real time analysis