Cassandra and Spark: Optimizing for Data Locality

Russell Spitzer
Software Engineer @ DataStax

Lex Luther Was Right: Location is Important

The value of many things is based upon it's location. Developed land near the beach is valuable but desert and farmland is generally much cheaper. Unfortunately moving land is generally impossible.

Lex Luther Was Wrong:
Don't ETL the Data Ocean

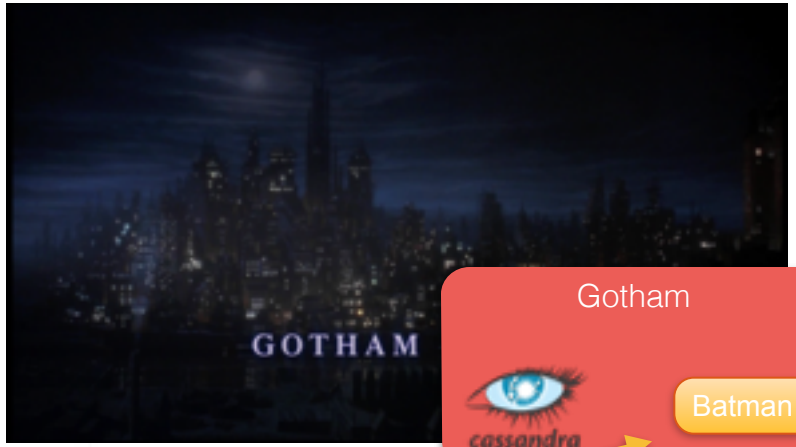
or lake, or swamp or whatever body of water is "Data" at the time this slide is viewed.

Spark is Our Hero Giving us the Ability to Do Our Analytics Without the ETL

Moving Data Between Machines Is Expensive
Do Work Where the Data Lives!

Our Cassandra Nodes are like Cities and our Spark Executors are like Super Heroes. We'd rather they spend their time locally rather than flying back and forth all the time.

Moving Data Between Machines Is Expensive
Do Work Where the Data Lives!



Metropolis

cassandra

Superman

Gotham

cassandra

Batman

Spark Executors

Our Cassandra Nodes are like Cities and our Spark Executors are like Super Heroes. We'd rather they spend their time locally rather than flying back and forth all the time.

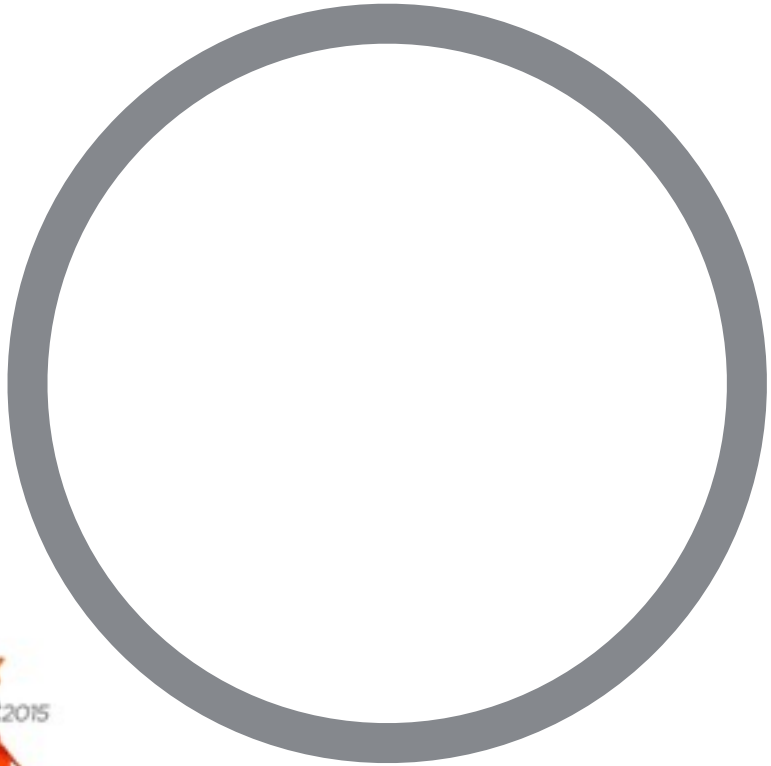
DataStax Open Source
Spark Cassandra Connector is Available on Github

- Compatible with Spark 1.3
- Read and Map C* Data Types
- Saves To Cassandra
- Intelligent write batching
- Supports Collections
- Secondary index pushdown
- Arbitrary CQL Execution!

https://github.com/datastax/spark-cassandra-connector

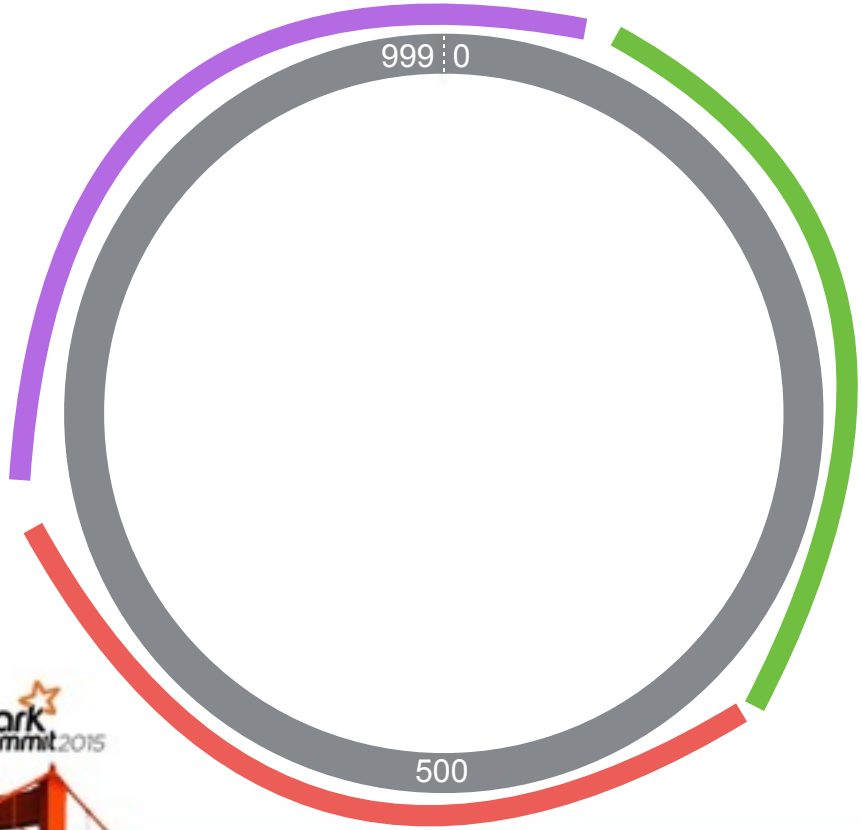
How the Spark Cassandra Connector Reads Data Node Local

Cassandra Locates a Row Based on Partition Key and Token Range



All of the rows in a Cassandra Cluster are stored based based on their location in the *Token Range*.

Cassandra Locates a Row Based on Partition Key and Token Range



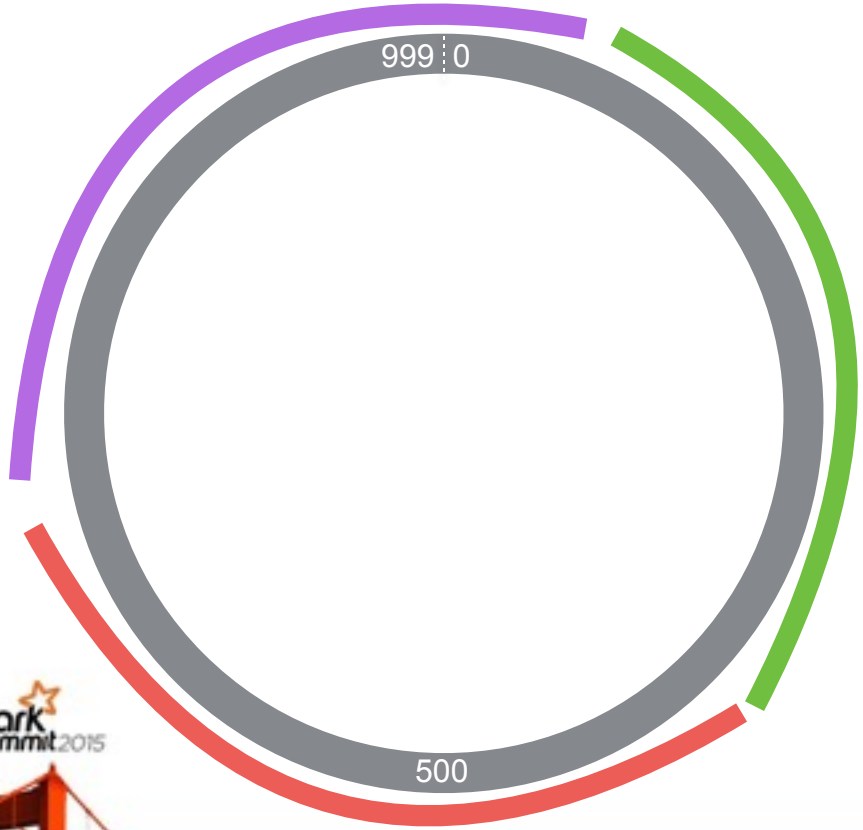
Metropolis

Gotham

Coast City

Each of the *Nodes* in a Cassandra Cluster is primarily responsible for one set of *Tokens.*

Cassandra Locates a Row Based on Partition Key and Token Range



Metropolis

750 - 99

Gotham

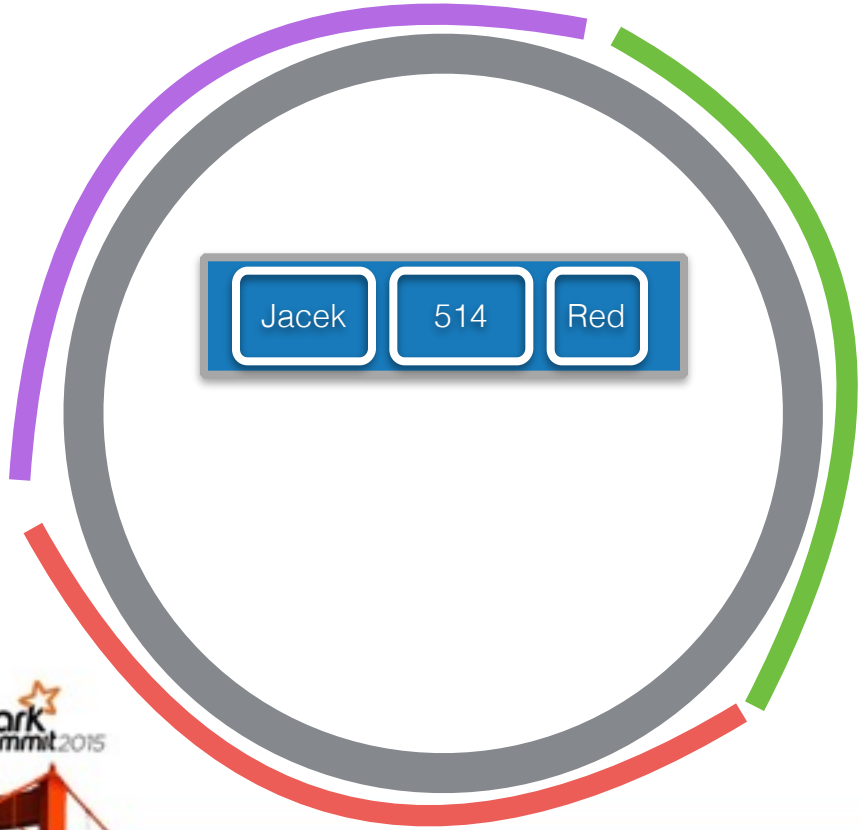
350 - 749

Coast City

100 - 349

Each of the *Nodes* in a Cassandra Cluster is primarily responsible for one set of *Tokens.*

Cassandra Locates a Row Based on Partition Key and Token Range



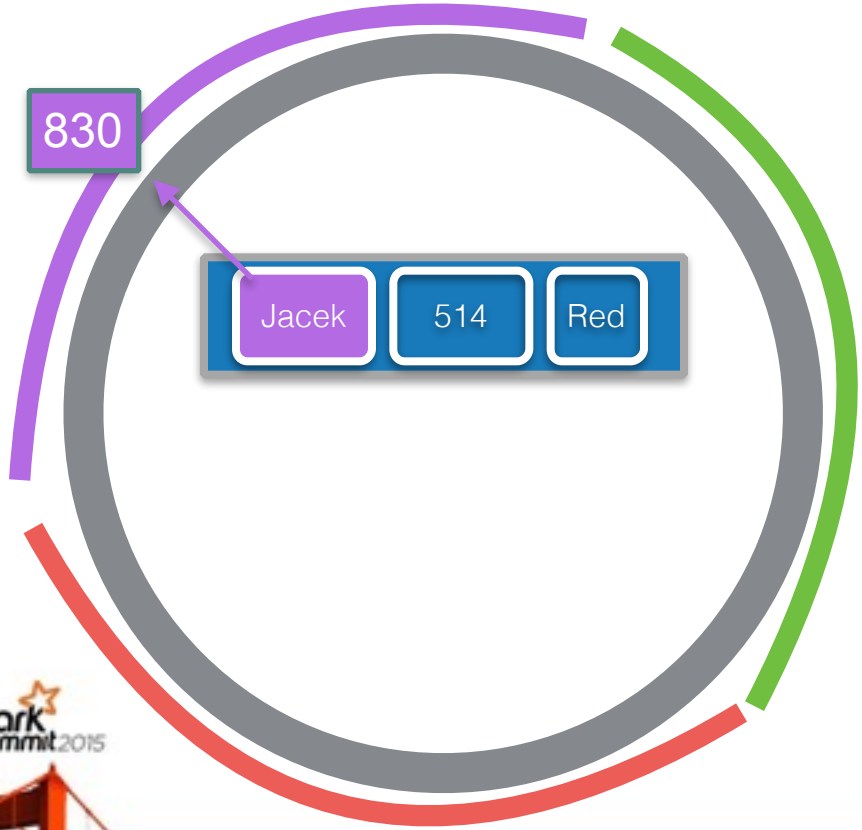
Metropolis

Gotham

Coast City

The CQL Schema designates at least one column to be the *Partition Key.*

Cassandra Locates a Row Based on Partition Key and Token Range



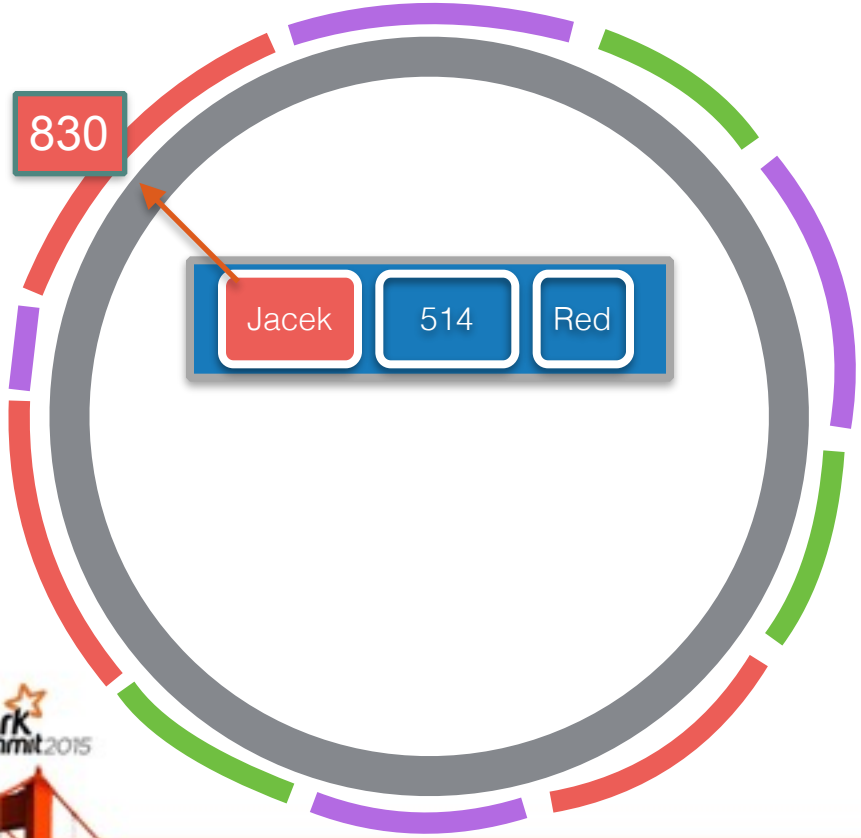
Metropolis

Gotham

Coast City

The hash of the *Partition Key* tells us where a row should be stored.

Cassandra Locates a Row Based on Partition Key and Token Range



Metropolis

Gotham

Coast City

With *VNodes* the ranges are not contiguous but the same mechanism controls row location.

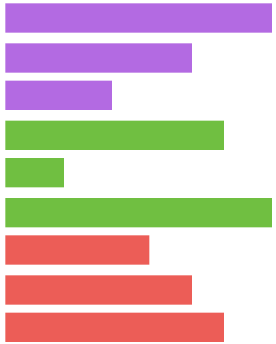
Loading Huge Amounts of Data

Table Scans involve loading most of the data in Cassandra

Cassandra RDD Use the Token Range to Create Node Local Spark Partitions

sc.cassandraTable or sqlContext.load(org.apache.spark.sql.cassandra)

Token Ranges

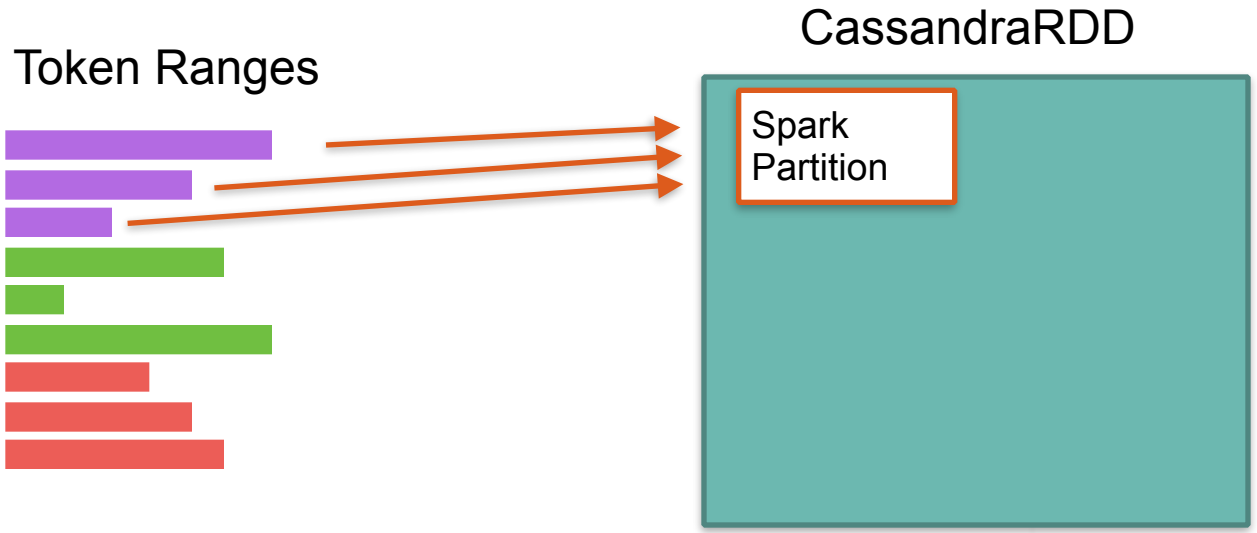


`spark.cassandra.input.split.size`

The (estimated) number of C* Partitions to be placed in a Spark Partition

Cassandra RDD Use the Token Range to Create Node Local Spark Partitions

sc.cassandraTable or sqlContext.load(org.apache.spark.sql.cassandra)

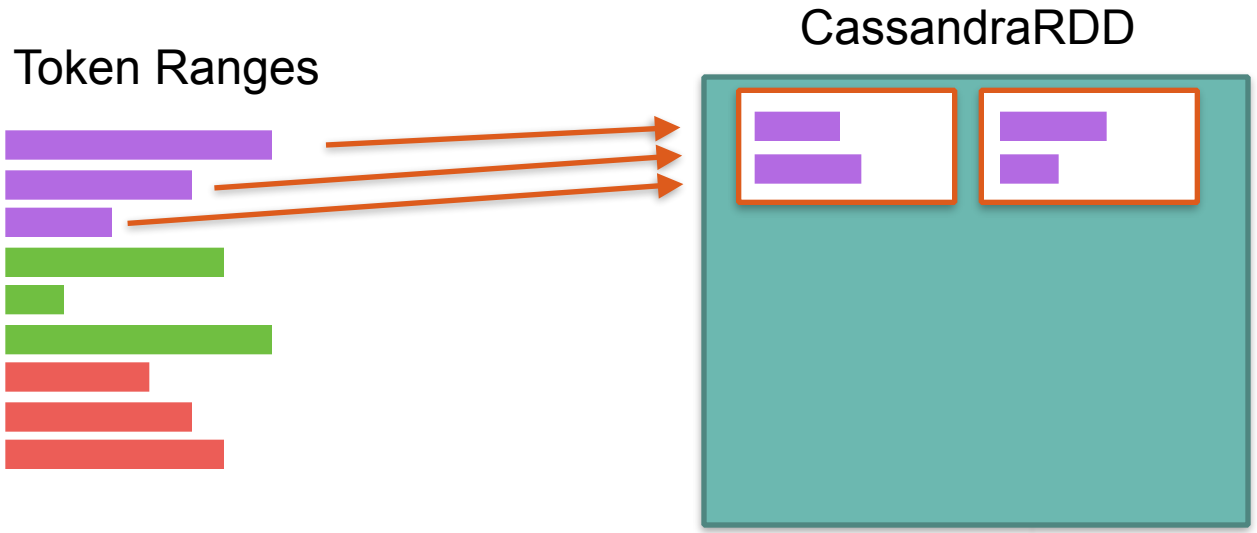


`spark.cassandra.input.split.size`

The (estimated) number of C* Partitions to be placed in a Spark Partition

Cassandra RDD Use the Token Range to Create Node Local Spark Partitions

sc.cassandraTable or sqlContext.load(org.apache.spark.sql.cassandra)

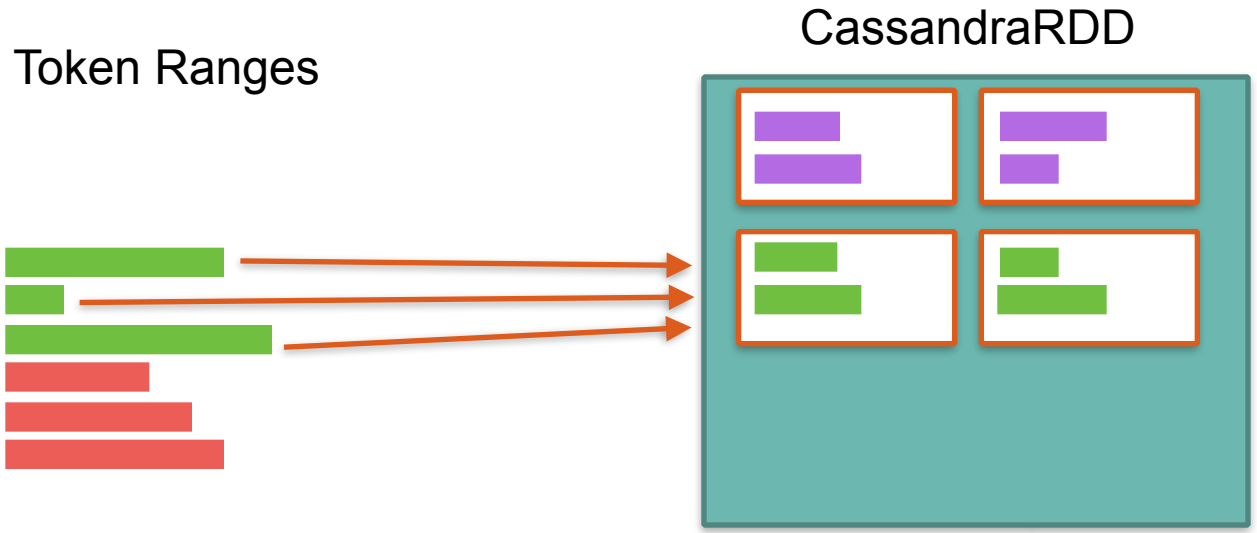


`spark.cassandra.input.split.size`

The (estimated) number of C* Partitions to be placed in a Spark Partition

Cassandra RDD Use the Token Range to Create Node Local Spark Partitions

sc.cassandraTable or sqlContext.load(org.apache.spark.sql.cassandra)

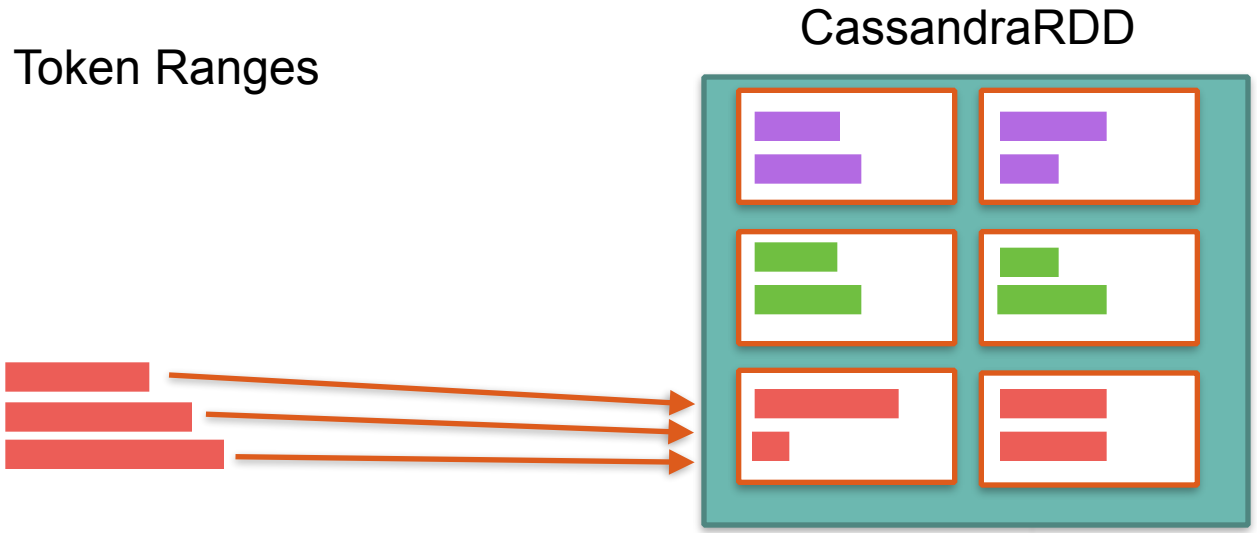


`spark.cassandra.input.split.size`

The (estimated) number of C* Partitions to be placed in a Spark Partition

Cassandra RDD Use the Token Range to Create Node Local Spark Partitions

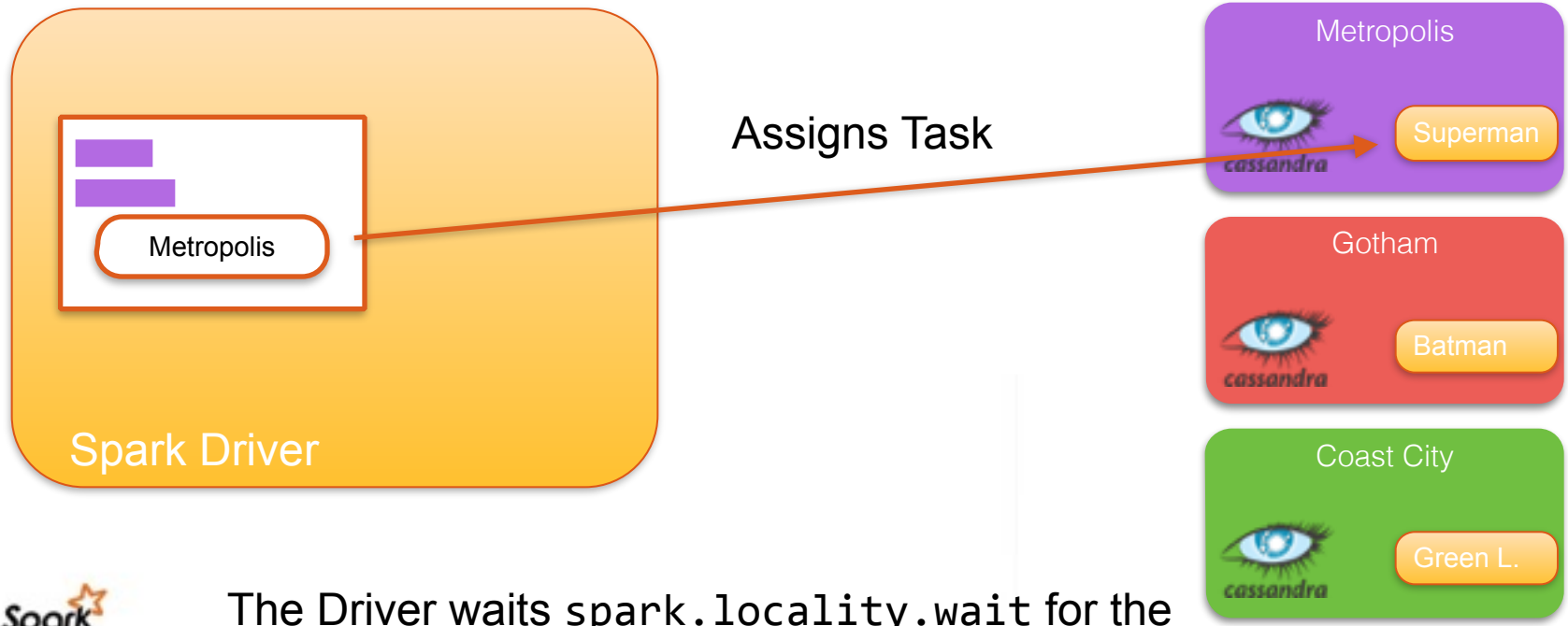
sc.cassandraTable or sqlContext.load(org.apache.spark.sql.cassandra)



`spark.cassandra.input.split.size`

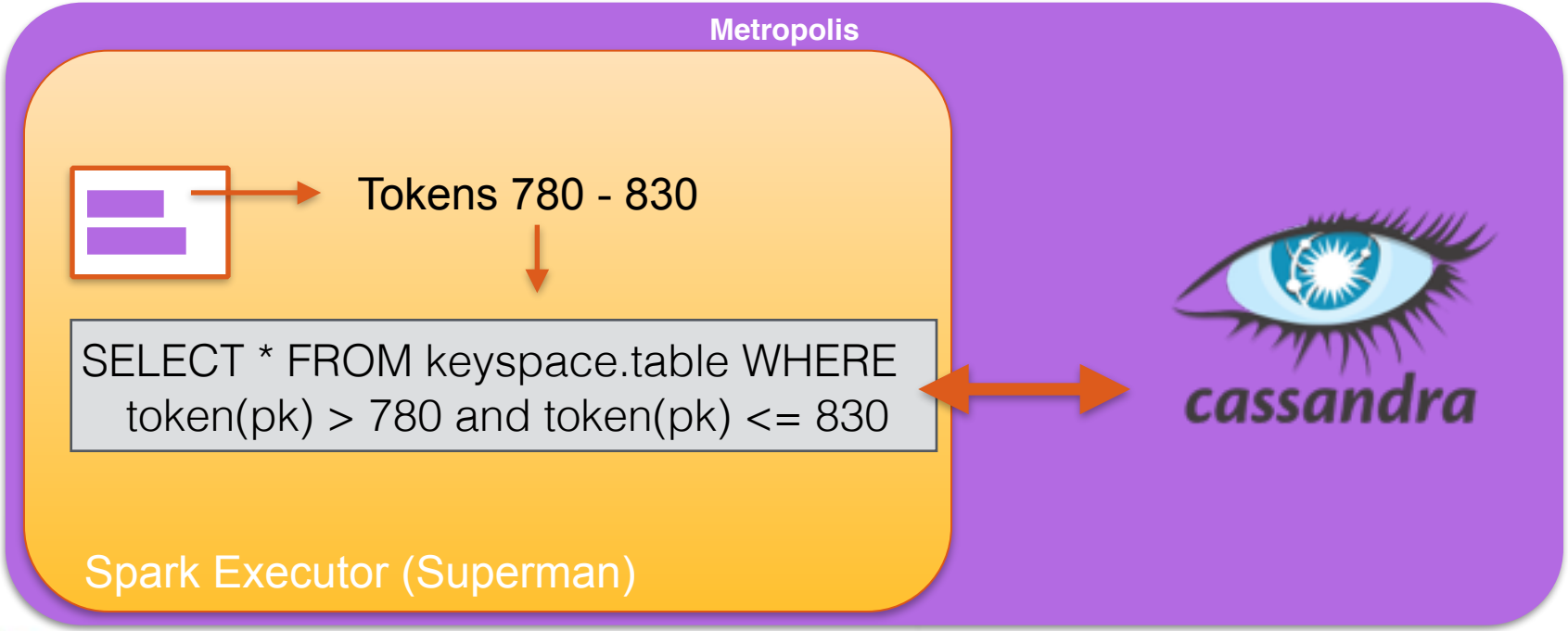
The (estimated) number of C* Partitions to be placed in a Spark Partition

Spark Partitions Are Annotated With the Location For TokenRanges they Span



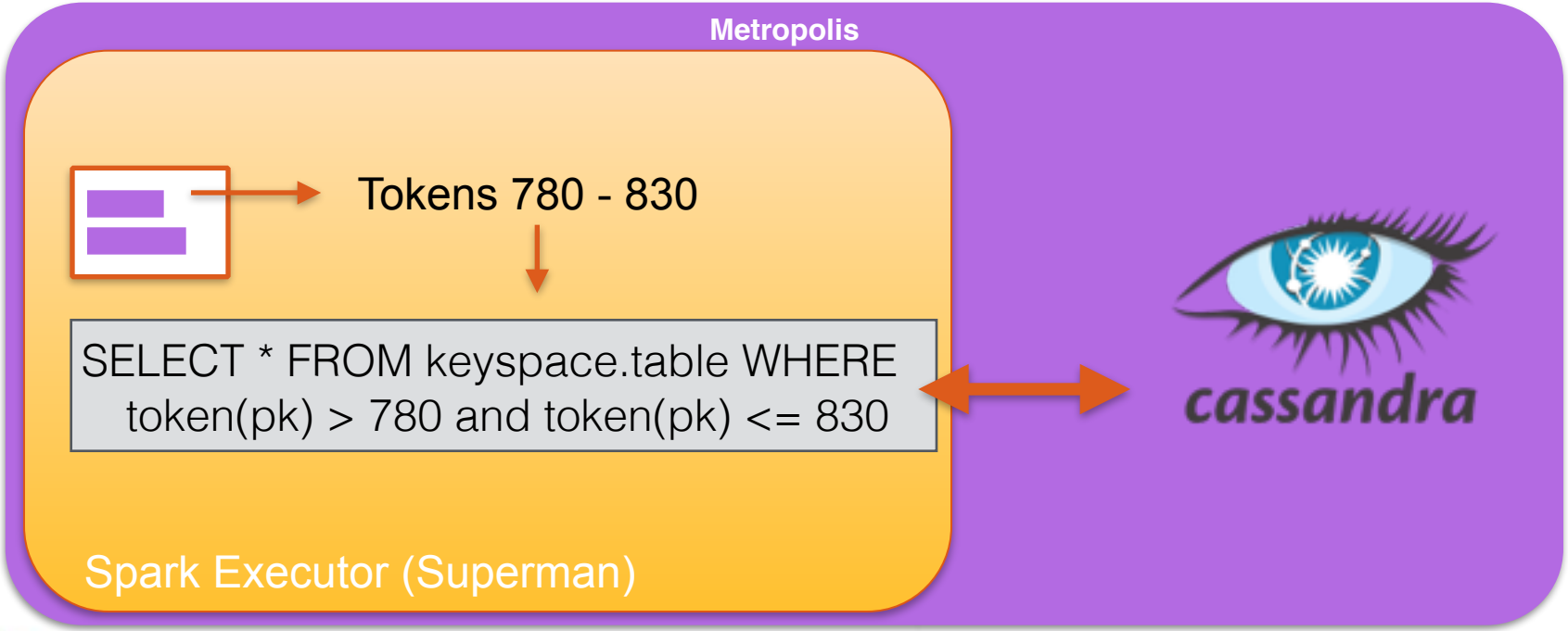
The Driver waits `spark.locality.wait` for the preferred location to have an open executor

The Spark Executor uses the Java Driver to Pull Rows from the Local Cassandra Instance



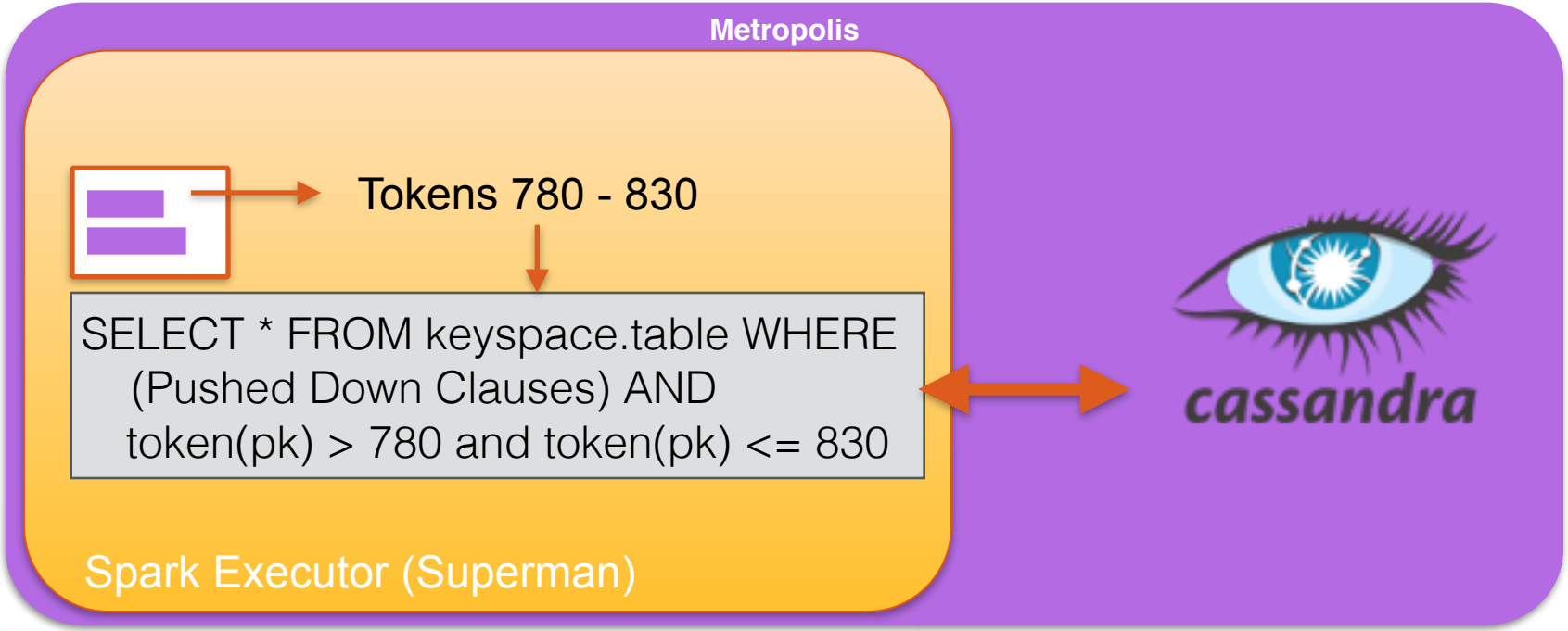
On the Executor the task is transformed into CQL queries which are executed via the Java Driver.

The Spark Executor uses the Java Driver to Pull Rows from the Local Cassandra Instance



The C* Java Driver pages `spark.cassandra.input.page.row.size` CQL rows at a time

The Spark Executor uses the Java Driver to Pull Rows from the Local Cassandra Instance



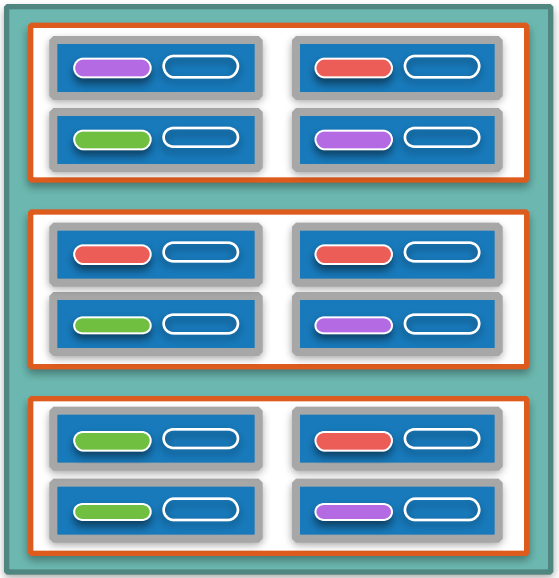
Because we are utilizing CQL we can also pushdown predicates which can be handled by C*.

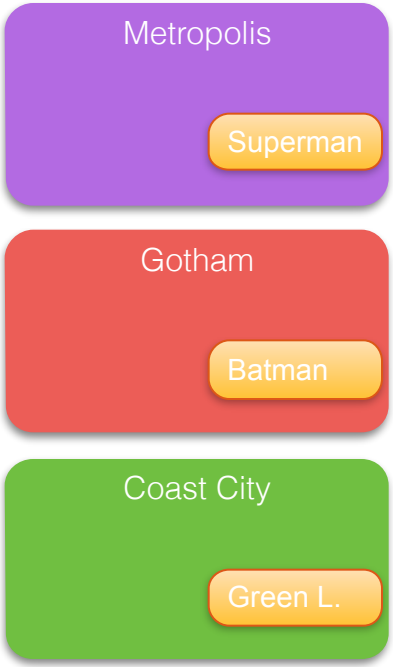
Loading Sizable But Defined Amounts of Data

Retrieving sets of Partition Keys can be done in parallel

joinWithCassandraTable Provides an Interface for Obtaining a Set of C* Partitions

Generic RDD

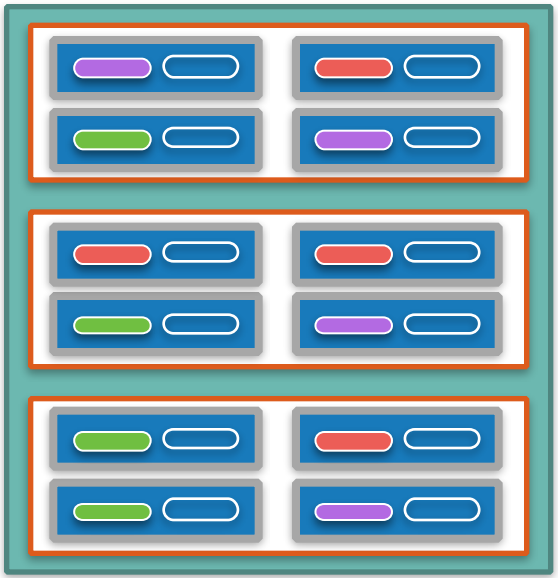


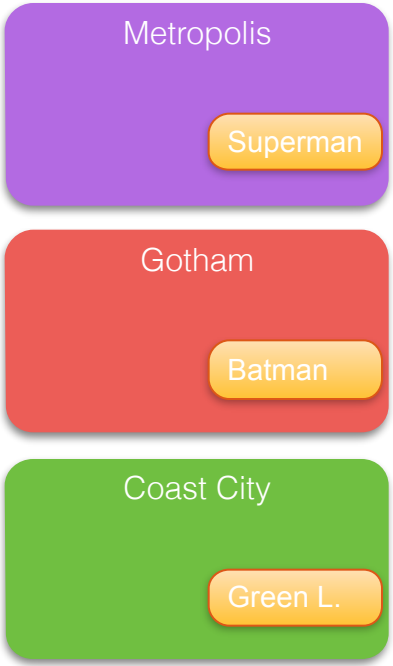


Generic RDDs can Be Joined But the Spark Tasks will Not be Node Local

joinWithCassandraTable Provides an Interface for Obtaining a Set of C* Partitions

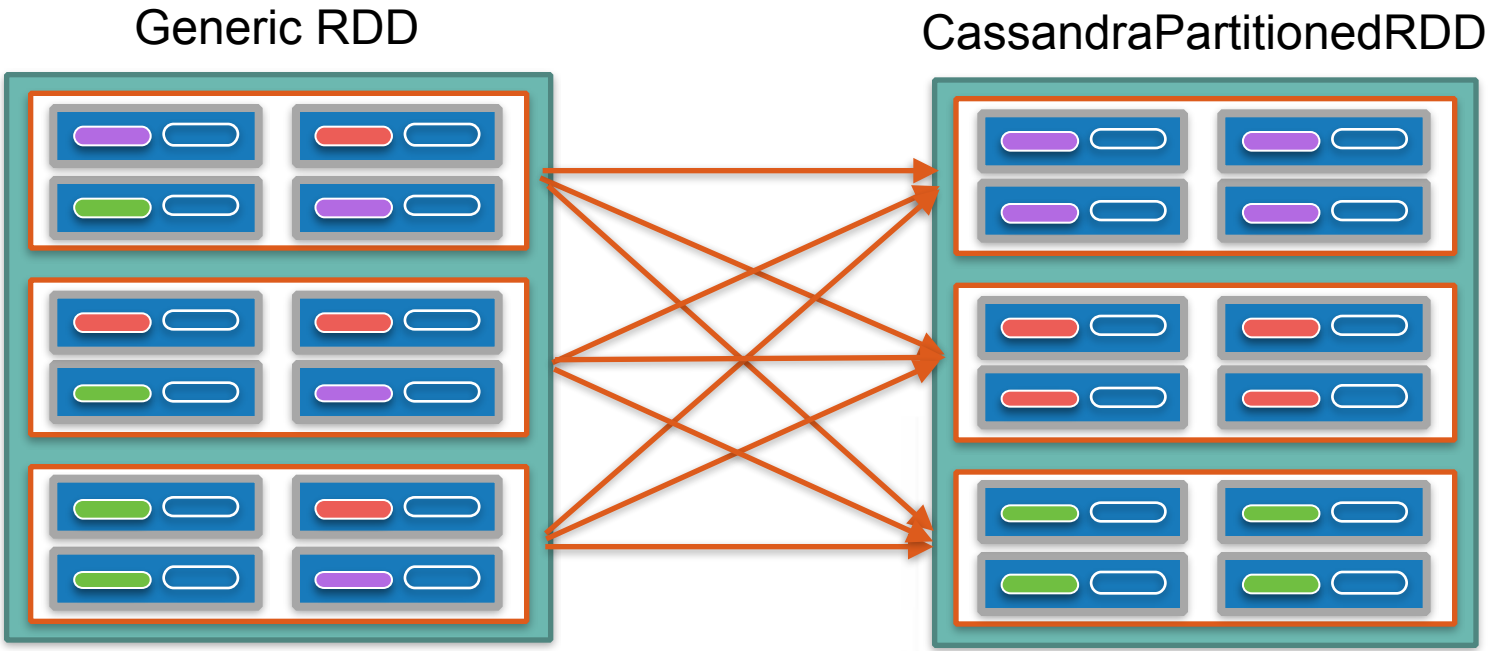
Generic RDD





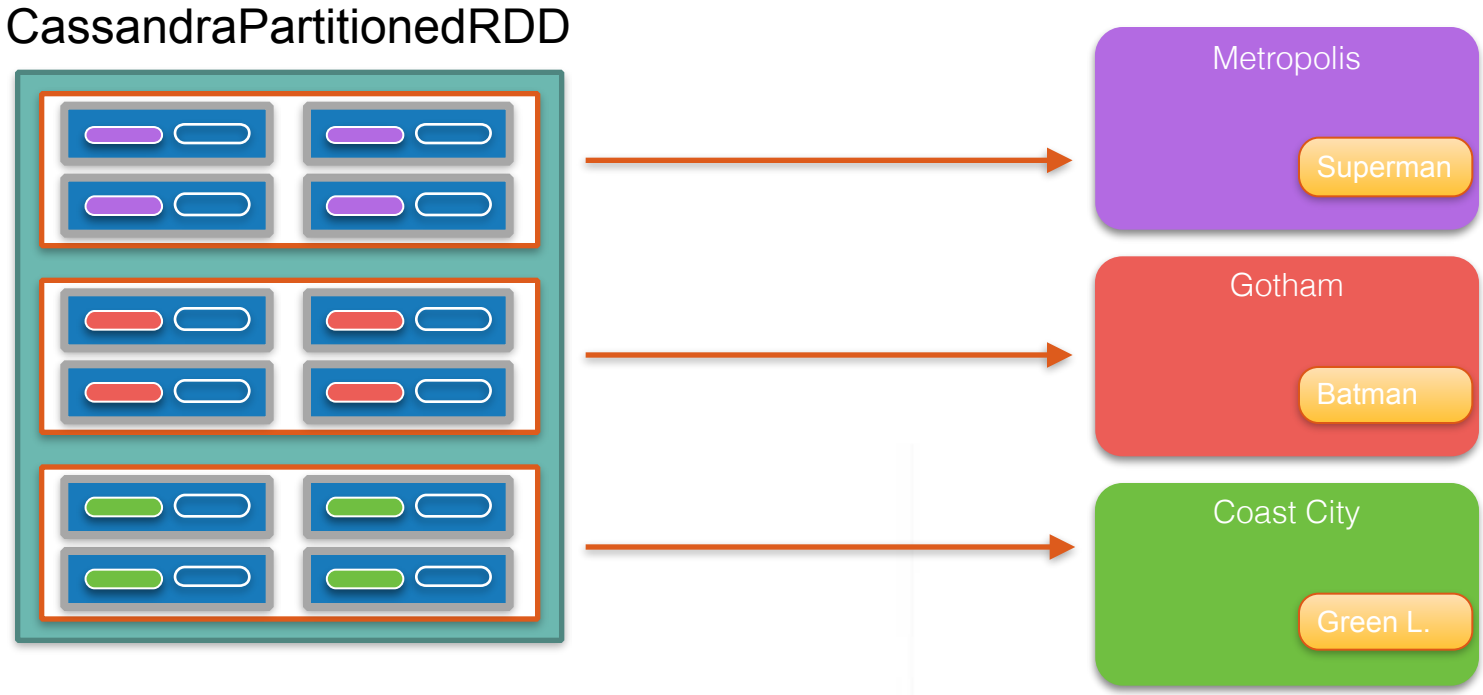
Generic RDDs can Be Joined But the Spark Tasks will Not be Node Local

repartitionByCassandraReplica Repartitions RDD's to be C* Local



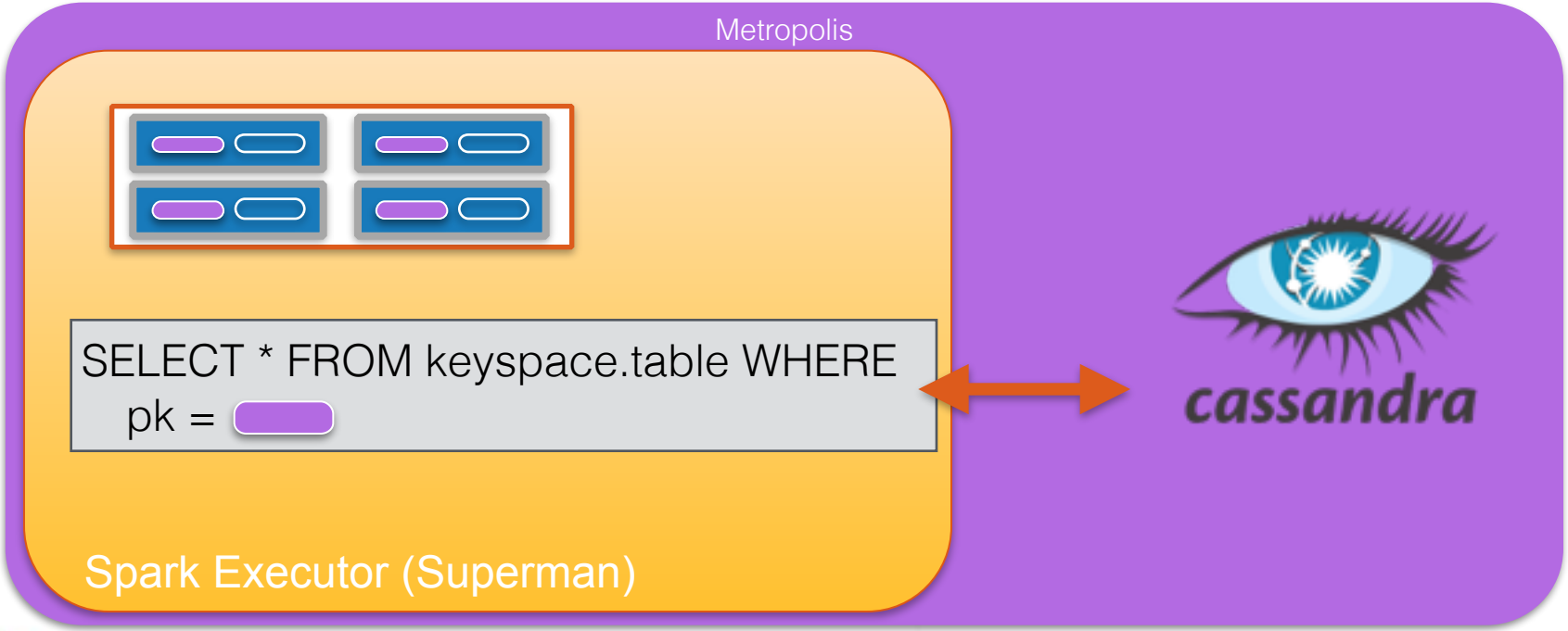
This operation requires a shuffle

joinWithCassandraTable on CassandraPartitionedRDDs (or CassandraTableScanRDDs) will be Node local



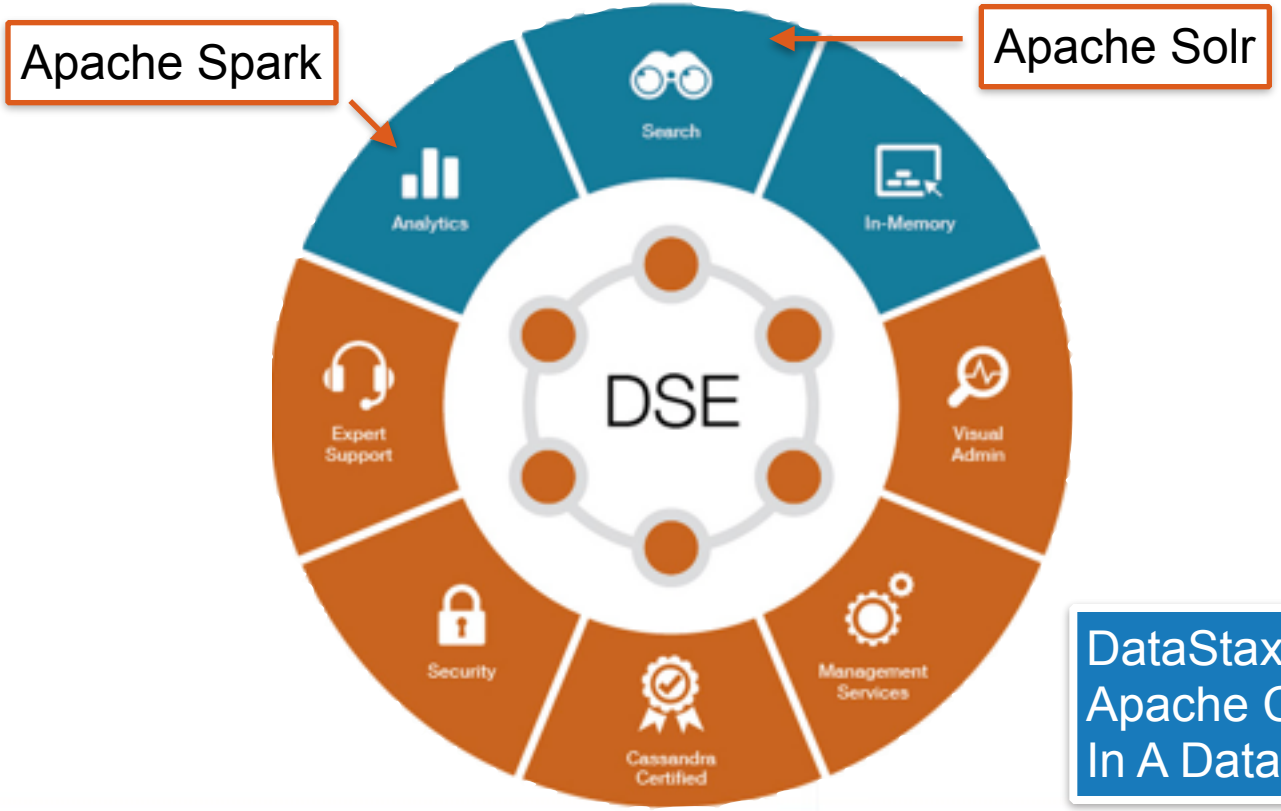
CassandraPartitionedRDDs are partitioned to be executed node local

The Spark Executor uses the Java Driver to Pull Rows from the Local Cassandra Instance



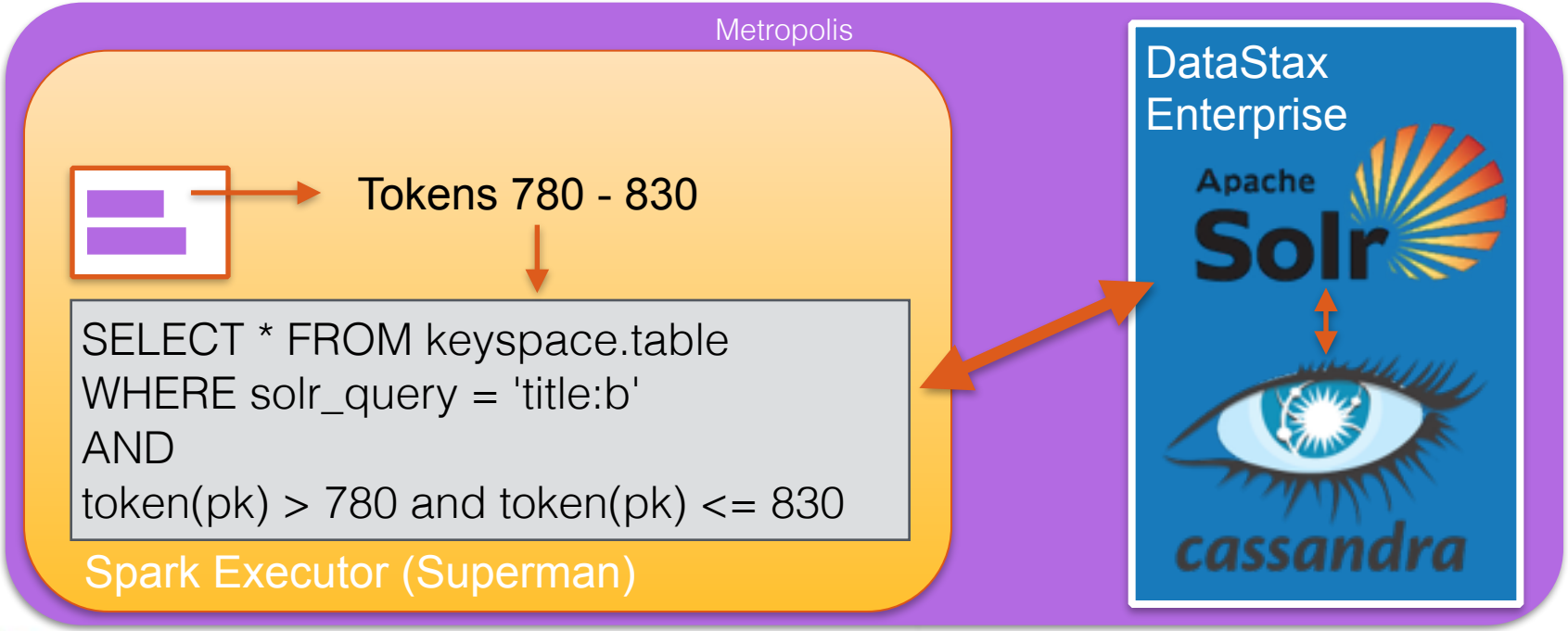
The C* Java Driver pages `spark.cassandra.input.page.row.size` CQL rows at a time

DataStax Enterprise Comes Bundled with Spark and the Connector



DataStax Delivers Apache Cassandra In A Database Platform

DataStax Enterprise Enables This Same Machinery with Solr Pushdown



Learn More Online and at Cassandra Summit

https://academy.datastax.com/