StratioDeep

An integration layer between Cassandra and Spark

Álvaro Agea Herradón
Antonio Alcocer Falcón
StratioDeep

An integration layer between Cassandra and Spark

Álvaro Agea Herradón
Antonio Alcocer Falcón
SDS

- Cassandra
- Stratio Real Time
- Stratio Meta
- Acquisition
- Stratio Deep
- BI Integration
An efficient data mining solution

“Two and two are four? Sometimes... Sometimes they are five.”

G. Orwell
GOALS

1. Why do you need Cassandra?
2. What is the problem?
3. Why do you need Spark?
4. How do they work together?
CASSANDRA

1. Based on DynamoDB...
2. Replication, Key/Value, P2P...
3. Based on Big Table...
4. Column oriented
1. Based on DynamoDB...
2. Replication, Key/Value, P2P...
3. Based on Big Table...
4. Column oriented
Each Cassandra server is assigned a unique token that determines what keys it is the first replica for.
Cassandra repairs missing data in three ways:

1. HintedHandoff
2. ReadRepair
3. AntiEntropy

Cassandra allows clients to specify the desired consistency level on reads and writes:
ONE, QUORUM, ALL...
Cassandra repairs missing data in three ways:

1. HintedHandoff
2. ReadRepair
3. AntiEntropy

Cassandra allows clients to specify the desired consistency level on reads and writes: ONE, QUORUM, ALL...
Cassandra repairs missing data in three ways:

1. HintedHandoff
2. ReadRepair
3. AntiEntropy

Cassandra allows clients to specify the desired consistency level on reads and writes: 
ONE, QUORUM, ALL...
CONSISTENCY

Cassandra allows clients to specify the desired consistency level on reads and writes:
ONE, QUORUM, ALL...

Cassandra repairs missing data in three ways:

1. HintedHandoff
2. ReadRepair
3. AntiEntropy
CONSISTENCY

Cassandra allows clients to specify the desired consistency level on reads and writes:
ONE, QUORUM, ALL...

Cassandra repairs missing data in three ways:

1. HintedHandoff
2. ReadRepair
3. AntiEntropy
2i: SECONDARY INDEX

Cassandra allows clients to create secondary index. You can search information in other column not only in the primary key.

Stratio has developed a new secondary index technology based on Apache Lucene:
- Full text search
- Full support to logic operator (AND, OR...)
Cassandra allows clients to create secondary index. You can search information in other column not only in the primary key.

Stratio has developed a new secondary index technology based on Apache Lucene:
- Full text search
- Full support to logic operator (AND, OR...)
YET ANOTHER DATABASE?
WHY?
WHY?
BIG DATA USE CASE A

FEW USERS / HIGH VOLUME OF DATA
BIG DATA USE CASE B

MANY USERS WITH LOW VOLUME OF DATA
BIG DATA USE CASE C

MANY USERS WITH HIGH VOLUME OF DATA
BUT…
1. In Cassandra, you need to design the schema with the query in mind

2. Every other type of query is either very inefficient or impossible to resolve
LET´S GET DIRTY
WHAT OPTIONS DO WE HAVE?

1. Run Hive Query on top of C*
2. Write an ETL script and load data into another DB
3. Clone the cluster
1. We can use CassandraStorageHandler

2. Create an external table in hive to use the data in Cassandra.

3. Queries are resolved in two phases:
   - Load data to Hadoop.
   - Run the HIVE batch process.

4. This solution DOES NOT WORK
1. We can use CassandraStorageHandler
2. Create an external table in hive to use the data in Cassandra.
3. Queries are resolved in two phases:
   - Load data to Hadoop.
   - Run the HIVE batch process.
4. This solution DOES NOT WORK
1. We need to carefully select the other DB:
   - Scalability

2. Prepare an ad-hoc data migration script...

3. It’s a waste of time:
   - Needs duplicated hardware.
   - Single threaded
   - Unreliable
   - Will still flush cache on cassandra nodes
1. We need to carefully select the other DB:
   • Scalability

2. Prepare an ad-hoc data migration script…

3. It’s a waste of time:
   • Needs duplicated hardware.
   • Single threaded
   • Unreliable
   • Will still flush cache on cassandra nodes
We immediately discard this choice

Why?

1. Worst possible network load
3. No incremental update
4. Need duplicate hardware.
WHAT OPTIONS DO WE HAVE?

- Run Hive Query on top of C*
- Write ETL scripts and load into another DB
- Clone the cluster
AND NOW… WHAT CAN WE DO?

“We can't solve problems by using the same kind of thinking we used when we created them”

Albert Einstein
CHALLENGE
ACCEPTED
SPARK
SPARK
SPARK

1. Alternative to MapReduce
2. A low latency cluster computing system
3. Designed for very large datasets
5. Up to 100 times faster than MapReduce for:
   - Interactive algorithms.
   - Interactive data mining
SPARK

1. Alternative to MapReduce
2. A low latency cluster computing system
3. Designed for very large datasets
5. Up to 100 times faster than MapReduce for:
   • Interactive algorithms.
   • Interactive data mining
SPARK IS UNIQUE IN UNIFYING

ALL IN A SINGLE FRAMEWORK

1. Batch, Interactive, Streaming computation models.
PERFORMANCE

Latency (sec)

https://amplab.cs.berkeley.edu/benchmark/
PERFORMANCE

Latency (sec)

- BI-Like
- Intermediate
- ETL-Like

https://AMPLab.cs.berkeley.edu/benchmark/
THE MOST ACTIVE BIG DATA APACHE PROJECT

1. Past 6 months: more active devs than Hadoop MapReduce
2. Better and faster evolutions: elegant and simple, fraction of code lines Vs. previous stack
THE **MOST ACTIVE** BIG DATA APACHE PROJECT

1. Past 6 months: more active devs than Hadoop MapReduce
2. Better and faster evolutions: elegant and simple, fraction of code lines Vs. previous stack
THE MOST ACTIVE BIG DATA APACHE PROJECT

1. Past 6 months: more active devs than Hadoop MapReduce
2. Better and faster evolutions: elegant and simple, fraction of code lines Vs. previous stack

![Graph showing code size comparison between different projects]

- Hadoop MapReduce: 350,000
- Storm (Streaming): 120,000
- Impala (SQL): 65,000
- Giraph (Graph): 90,000
- Spark: 75,000

non-test, non-example source lines
SPARK WORKING OVER CASSANDRA

“Two plus two is four? Sometimes… Sometimes it is five.”

G. Orwell
DIFFERENT OPTIONS
1 CASSANDRA’S HDFS ABSTRACTION LAYER

Advantages

- Easily integrates with legacy systems.

Drawbacks

- Very high-level: no access to Cassandra’s low level features.
- Questionable performance.
Thrift protocol

- Obsolete

CQLPagingInputFormat

- Under the hood keeps using thrift protocol
- No control over implemented behaviour
Thrift protocol
  • Obsolete

CQLPagingInputFormat
  • Under the hood keeps using thrift protocol
  • No control over implemented behaviour
Uses the modern Datastax Java driver

- Uses only CQL3 (No thrift)
- Full control over data locality
- No dependency on Hadoop
- No dependency on unmaintainable code
- Fine grained control over Cassandra splits
Uses the modern Datastax Java driver

- Uses only CQL3 (No thrift)
- Full control over data locality
- No dependency on Hadoop
- No dependency on unmaintainable code
- Fine grained control over Cassandra splits
Stratio Deep provides two user interfaces:

- Domain object based
- Generic cell API
1. API let’s you map your Cassandra tables to object entities (POJOs), just like if you were using any other ORM.

2. This abstraction is quite handy, it will let you work on RDD (under the hood StratioDeep will map columns to entity properties).
1. Allows to work on RDD where Cells is a collection of Cell objects. Column metadata is automatically fetched from the datastore.

2. This interface is a little bit more cumbersome to work with, but has the advantage that it does not require the definition of additional entity classes.
SPARK-CASSANDRA ALLOWS SELECTING JUST THE INITIAL DATA SPARK NEEDS FROM THE CASSANDRA DATA STORE

With the Spark-Cassandra integration we can leverage the power of Cassandra’s main indexes, and especially secondary indexes in order to only fetch the data we need efficiently.
Stratio RT extension

• Built on-top of Spark Streaming API.
• Registration of new channels/consumer/producers

Cross-module integration with StratioMeta

• Lets us create flows of data between StratioDeep ↔ StratioRT
• Materialized views, live queries, alerts, etc...
Stratio RT extension
• Built on-top of Spark Streaming API.
• Registration of new channels/consumer/producers

Cross-module integration with StratioMeta
• Lets us create flows of data between StratioDeep ↔ StratioRT
• Materialized views, live queries, alerts, etc...
Stratio Meta is a SQL-Like language:

- Built on-top of Spark’s API.
- Unique interface to all Stratio platform modules
- Smart engine to decide the best way to resolve the queries
- Mix stream processing and batch in one language
With Spark-Cassandra we are covering a more complete use case

A lot of users accessing a lot of data from applications or predefined reports. The needs of data analysts that can transform, analyze, and query high volumes of data openly with a more powerful data manipulation tool in their hands.
CONCLUSION

With Spark-Cassandra we are covering a more complete use case

A lot of users accessing a lot of data from applications or predefined reports. The needs of data analysts that can transform, analyze, and query high volumes of data openly with a more powerful data manipulation tool in their hands.