



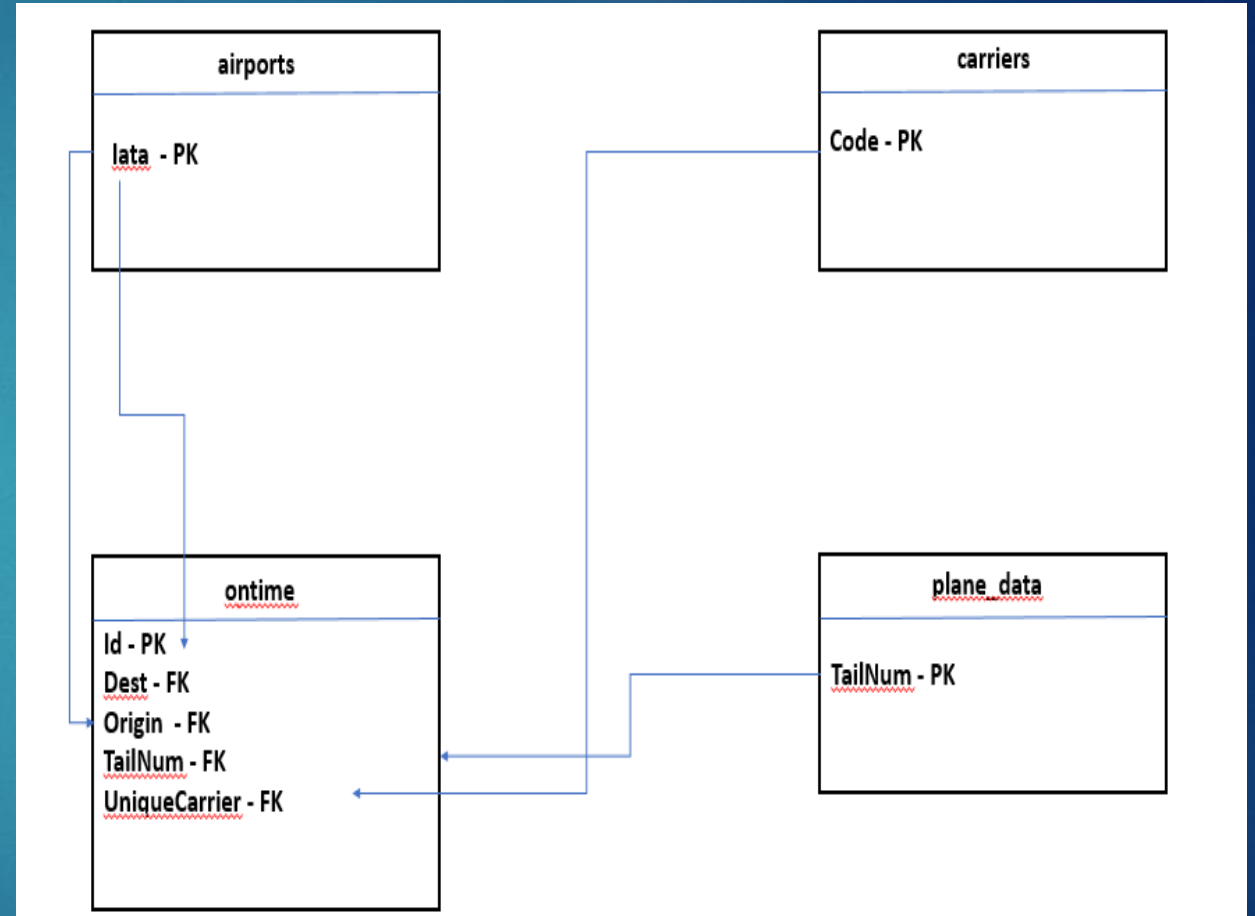
Big Data

Airlines on-time performance

A COMPARATIVE STUDY OF HADOOP SOLUTIONS

Dataset

Name	Date modified	Type	Size
1987.csv	20/02/2017 13:50	WinZip File	12,356 KB
1988.csv	20/02/2017 13:51	WinZip File	48,339 KB
1989.csv	20/02/2017 13:51	WinZip File	48,050 KB
1990.csv	20/02/2017 13:54	WinZip File	50,822 KB
1991.csv	20/02/2017 12:55	WinZip File	48,709 KB
1992.csv	20/02/2017 12:57	WinZip File	48,869 KB
1993.csv	20/02/2017 13:01	WinZip File	48,938 KB
1994.csv	20/02/2017 12:55	WinZip File	49,926 KB
1995.csv	20/02/2017 12:59	WinZip File	73,127 KB
1996.csv	20/02/2017 13:00	WinZip File	74,110 KB
1997.csv	20/02/2017 13:00	WinZip File	74,908 KB
1998.csv	20/02/2017 12:58	WinZip File	74,887 KB
1999.csv	20/02/2017 13:01	WinZip File	77,588 KB
2000.csv	20/02/2017 13:02	WinZip File	80,604 KB
2001.csv	20/02/2017 12:56	WinZip File	81,523 KB
2002.csv	20/02/2017 13:00	WinZip File	74,129 KB
2003.csv	20/02/2017 12:13	WinZip File	93,093 KB
2004.csv	20/02/2017 12:16	WinZip File	108,228 KB
2005.csv	20/02/2017 12:17	WinZip File	109,815 KB
2006.csv	20/02/2017 12:15	WinZip File	112,324 KB
2007.csv	20/02/2017 11:56	WinZip File	118,408 KB
2008.csv	09/02/2017 20:47	WinZip File	111,088 KB
airports	20/02/2017 11:30	Microsoft Excel Co...	209 KB
carriers	17/01/2017 14:01	Microsoft Excel Co...	43 KB
plane-data	17/01/2017 14:01	Microsoft Excel Co...	419 KB



Experiment

Download data from the web

Location: <http://stat-computing.org/dataexpo/2009/the-data.html>

Store the data on a disk

Move the data to the cluster file system

a) Use SSH Secure File Transfer
b) For each zip file that require unzipping
>> `bzip2 -z [filename]`

Copy the files to hdfs and unzip

c) For each file in the file system that require a copy to hdfs:
>> `hadoop fs copyFromLocation [fileName]`

d) Open a Hive connection
>> `hive`

e) Start running the HiveQL code
>> `SELECT * FROM ...`

Hive

Create the Schemas

Load the data

Write HiveSQL

Run report queries

Load the data

Write the Scala query code

Run the Scala queries

Spark

d') Open a Spark connection and run the code
>> `spark-shell --driver-memory 10G -i bigdata/SparkProto.scala`

Results High Level Functional Comparison

Hive

	Comments
Maintenance	UDFs / views
Robustness & Reliability	Disaster-recovery
Extendibility/ Interfacing	Large datasets stored in HDFS/Amazon S3 filesystem, RCFile, HBase, etc.) HiveQL -> transformed into MapReduce, Tez and/or Spark jobs. CLI, a UI, and Thrift Server.
Performance	HiveQL Indexes & Partitions. HiveQL optimiser (optimized DAG).
Throughput	Distributed storage / Parallel MapReduce jobs for extra data load.
Learning Curve	Reduced learning curve from ANSI SQL to HiveQL queries. SQL skills is well spread in the community.

Spark

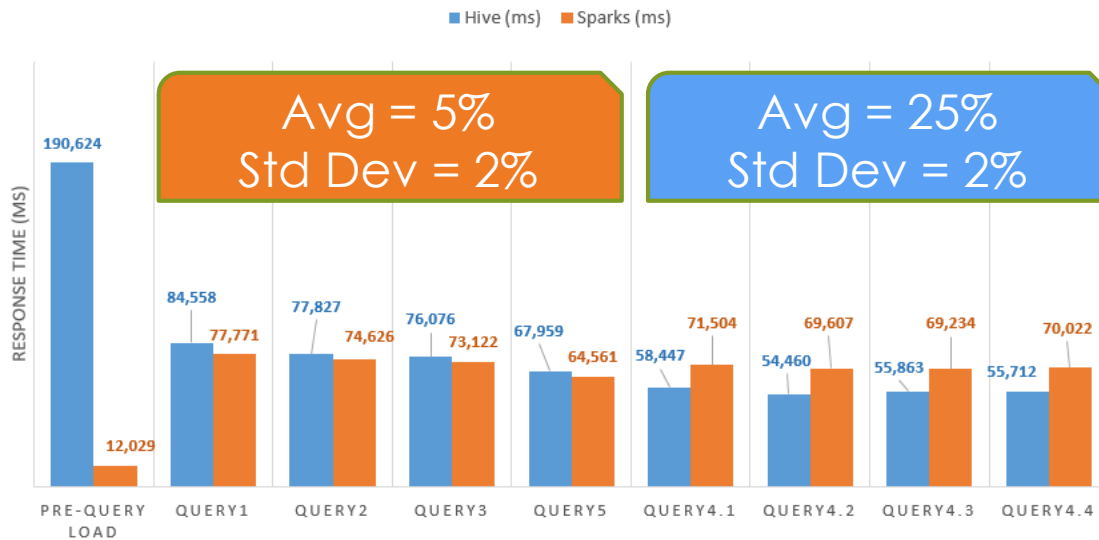
	Comments
Maintenance	Spark RDDs -> iterative algorithms & explanatory data analysis (database query style). Spark RDDs -> Java/Scala/Python and other Object Oriented languages.
Robustness & Reliability	Robustness/Reliability -> architecture engineering. RDDs -> saved/recovered at given point in time.
Extendibility/ Interfacing	Spark natively communicate with: <ul style="list-style-type: none"> - Spark SQL - Spark Streaming - MLlib - GraphX Interfaces with a wide variety of components such as , including HDFS, Cassandra, OpenStack Swift, Amazon S3etc.
Performance	No need for slow MapReduce program. Data can be lazy loaded in memory. -> Reduce read/write to disk time, network traffic and process intercommunication.
Throughput	Clusters -> to self-adapt to high-throughput situations.
Learning Curve	Requires more advanced development skills that may harder to find on the job market.
Environment	Can be run on the Hadoop platform interfacing through Yarn or as a standalone executor, independent from Hadoop.

Results **Key Facts**

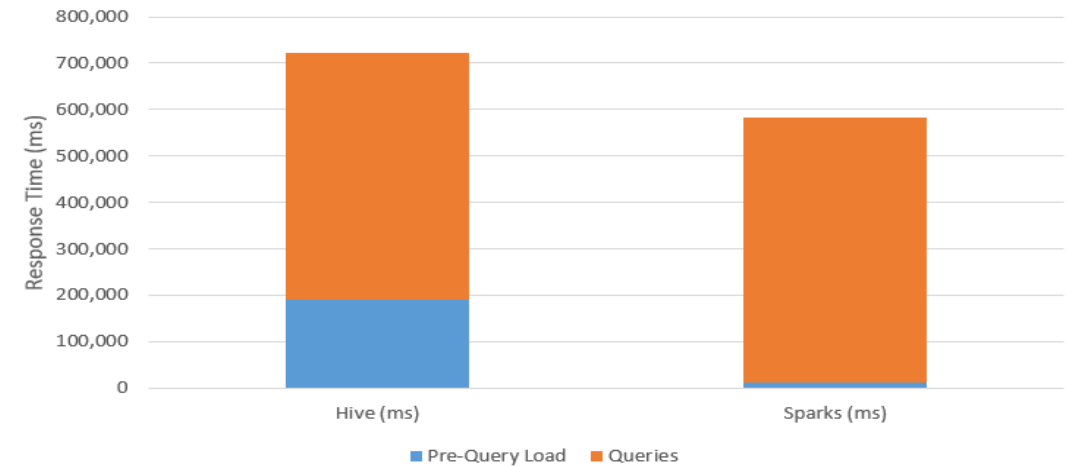
Questions	Answers
What are the routes the most affected by cancellations (by year)?	The San Francisco International – Los Angeles route and reverse between 1998 and 2000.
What are the destinations the most affected by delays (by year)?	Chicago O'Hare International between 2006 and 2008
What is the airline that suffered from the longest delays (by year)	Hawaiian Airlines Inc between 2005 and 2007
What were the best 5 years to fly to minimise delays?	1987,1992,2002,1991 and 1993 in order
What were the best 5 months to fly to minimise delays?	September, October, April, May and November in order.
What were the best 5 days of week to fly to minimise delays	Saturday, Tuesday, Wednesday, Thursday, Monday and Sunday in order
Did older planes suffer more delays?	Generally, yes. Although it is difficult to draw an objective conclusion.

Results Pre-Fetch & Query Response Time Comparison

HIVE VS SPARK
RESPONSE TIME



HIVE VS SPARK
CUMULATED
RESPONSE TIME



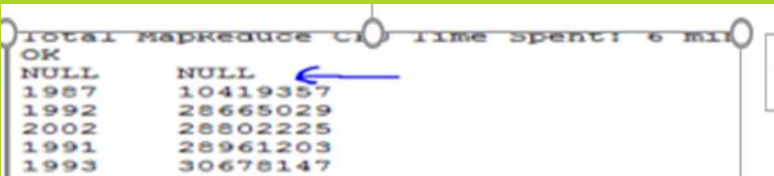
Queries with Joins		Avg	Std Dev
Hive Response Time (ms)		76,605	6,826
Spark Response Time (ms)		72,520	5,648
Queries without Joins			
Hive Response Time (ms)		56,121	1,674
Spark Response Time (ms)		70,092	995

Lessons Learnt & Challenges

Hive

- Unexpected disconnection -> Kill Hive process
- Extra Null row for non string columns

```
SELECT COUNT(Year)
FROM airport_flights.ontime
GROUP BY Year;
```



OK	NULL	NULL
1987	10419357	
1992	28665029	
2002	28802225	
1991	28961203	
1993	30678147	

- Column headers are not printed by default
-> SET hive.cli.print.header=TRUE;

Spark

- Process memory -> spark-shell --driver-memory 10G -i bigdata/SparkProto.scala
- Dataframes requires some lib to loaded in specific order

```
config.set("fs.defaultFS", uri);
var folderPath = "hdfs:///user/fmare001/"
//Create the SQLContext
val sc: SparkContext;
val sqlContext = new org.apache.spark.sql.SQLContext(sc);
// For implicit conversions like converting RDDs to DataFrames
import org.apache.spark.implicits._
import sqlContext.implicits._
```

- Spark/Scala Syntax writing -> counter intuitive

```
//Now join on the airport_dest_df and airport_origin_df to get the readable airport names
cancellations_df.join(airport_origin_df, cancellations_df("Origin") === airport_df("Iata"), "left outer").join(
airport_dest_df, cancellations_df("Dest") === airport_df("Iata"), "left outer").select("Year", "Origin_Airport", "Origin",
"Dest_Airport", "Dest", "SUM(Cancelled)").show();
```


Conclusion

Hive -> DB + Batch / biased towards end users.

Sparks -> Server Side + Real-time/ need programming skills.

Q&A

