

Batch mode and Machine Learning canvas

Kafka

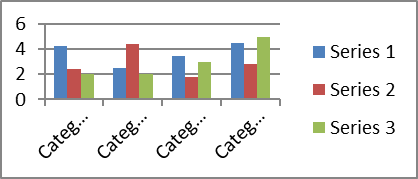
Flume

Spark Stream

Enrich Tables

NoSQL Server

HDFS



Real time canvas

**Use-Case: Airline on-time performance**

**Reference Link:** <http://stat-computing.org/dataexpo/2009/>

**The data:**

The data consists of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008. This is a large dataset: there are nearly 120 million records in total, and takes up 1.6 gigabytes of space compressed and 12 gigabytes when uncompressed.

***Solution***

**Batch Ingestion & Processing –**

* 1. **Hadoop directory Structure created.**

|  |  |  |
| --- | --- | --- |
| **Layer** | **Directory Path** | **File Format** |
| RAW | /data/raw/ | As is (e.g. TXT, CSV, XML, JSON, etc.,) |
| Decomposed | /data/decomposed/ | Avro |
| Modelled | /data/modelled/ | Parquet |
| Schema (Meta data) | /data/schema/ | AVSC schema |

* 1. **Source data details**

Stats created for year 2008 & 2007.

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

**Supplemental Data:** <http://stat-computing.org/dataexpo/2009/supplemental-data.html>

* 1. **Data preparation**
  + Moved data to Kafka cluster
  + Topics in Kafka
    - Airports
    - Carriers
    - Planedate
    - OTP
  + Stream data to Kafka cluster.
  1. **Batch Ingestion (HDFS)**
* **Raw layer (Store data AS-IS)**
  + Use *Apache Flume* to consume messages from Airports & Planedate Kafka Topic to HDFS Raw folder
  + Use *Spark Streaming* to consume messages from Carriers and OTP Kafka Topic to HDFS Raw folder
* **Decomposed layer (Append UUID and timestamp to the AS-IS data)**
  + For each message in the Airports & Planedate data from raw directory, append UUID and timestamp using Pig Latin.
  + For each message in the Carriers & OTP data from raw directory, append UUID and timestamp using Pig Latin.
  1. **Modelling and processing**
* Cleanse the data (trim, null, removing duplicates) and load it in Parquet format as modelled using Spark/Scala
  1. **Solution using R/Python/Spark MLlib tools and Big Data platform to answer the following questions.**
* Which carrier performs better?
* When is the best time of day/day of week/time of year to fly to minimise delays?
* Do older planes suffer more delays?
* Can you detect cascading failures as delays in one airport create delays in others? Are there critical links in the system?
* Create a model to predict flight delays
* How well does weather predict plane delays?

