

Applying NoSQL solutions for Stock Market Prediction using Neural Networks

By Konstantina Noutsis

Greenfield Project

Phase 1: Installation / Initiation

- Installation of the Apache Cassandra database, transformation of the data to the Cassandra flexible schema format and population of the data.

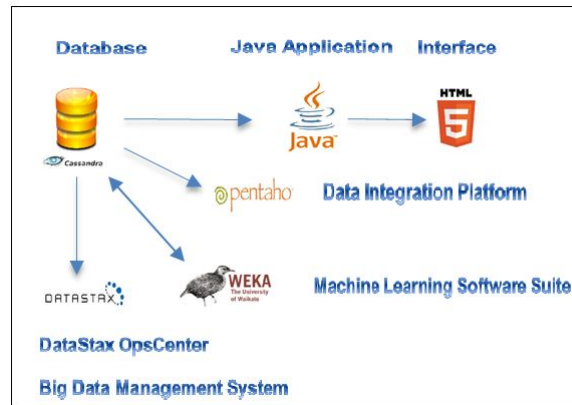
Phase 2: Data Backend/Machine Learning

- Installation of the Weka machine learning tool and plugin connectivity among Cassandra and Weka in order to train the data and build “what if” analysis for different market scenarios.

Phase 3: User Interface

- Graphical user interface build in HTML 5, using JavaScript Libraries e.g. JQuery, Flot.

The Technologies



Step By Step

- What is NoSQL?

NoSQL database provides a mechanism for storage and retrieval of data that uses looser consistency models than traditional relational databases.

Examples:

- **Column:** Hbase, Cassandra
- **Document:** MongoDB, Couch, Raven
- **Key-value :** Dynamo, Riak, Azure, Redis, Cache, GT.m
- **Graph:** Neo4J, Allegro, Virtuoso, Bigdata

Why Cassandra

Cassandra is renowned for its exceptionally fast Writes, which it accomplishes by keeping most of its data within main memory at the responsible node. It also relies on a series of clever techniques with names like read repair, hinted handoff, and anti-entropy to resolve the data inconsistencies that inevitably arise when data is being analyzed by multiple processing nodes.

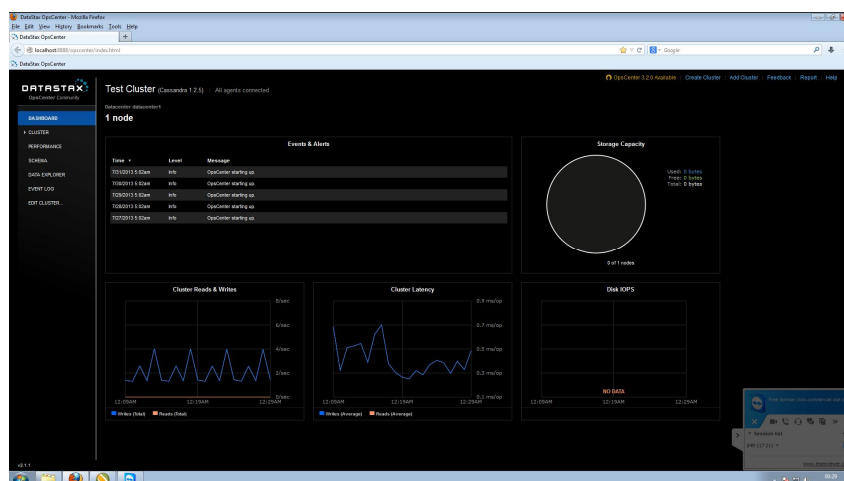
Cassandra Details

- In Cassandra, the *keyspace* is the container for your application data, similar to a database or schema in a relational database. Inside the keyspace are one or more *column family* objects, which are analogous to tables. Column families contain *columns*, and a set of related columns is identified by an application-supplied row *key*. Each row in a column family is *not* required to have the same set of columns.
- Cassandra does not enforce relationships between column families the way that relational databases do between tables: there are no formal foreign keys in Cassandra, and joining column families at query time is not supported. Each column family has a self-contained set of columns that are intended to be accessed together to satisfy specific queries from your application.

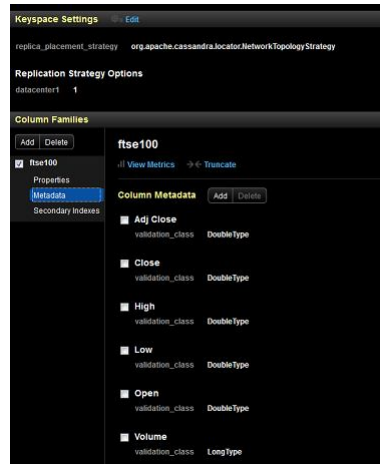
Access to the Database



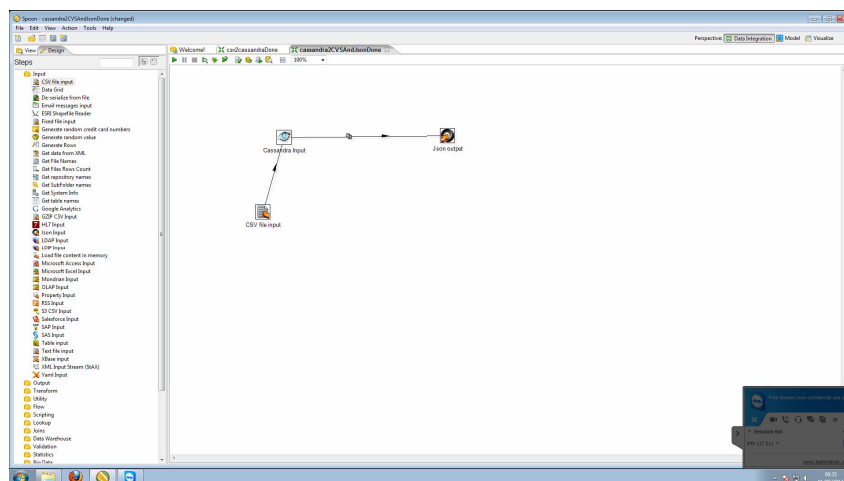
Nodes



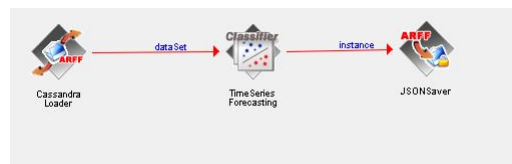
Column Families



Load Data



Forecasting



HTML5- Bootstrap

