Twitterlytics

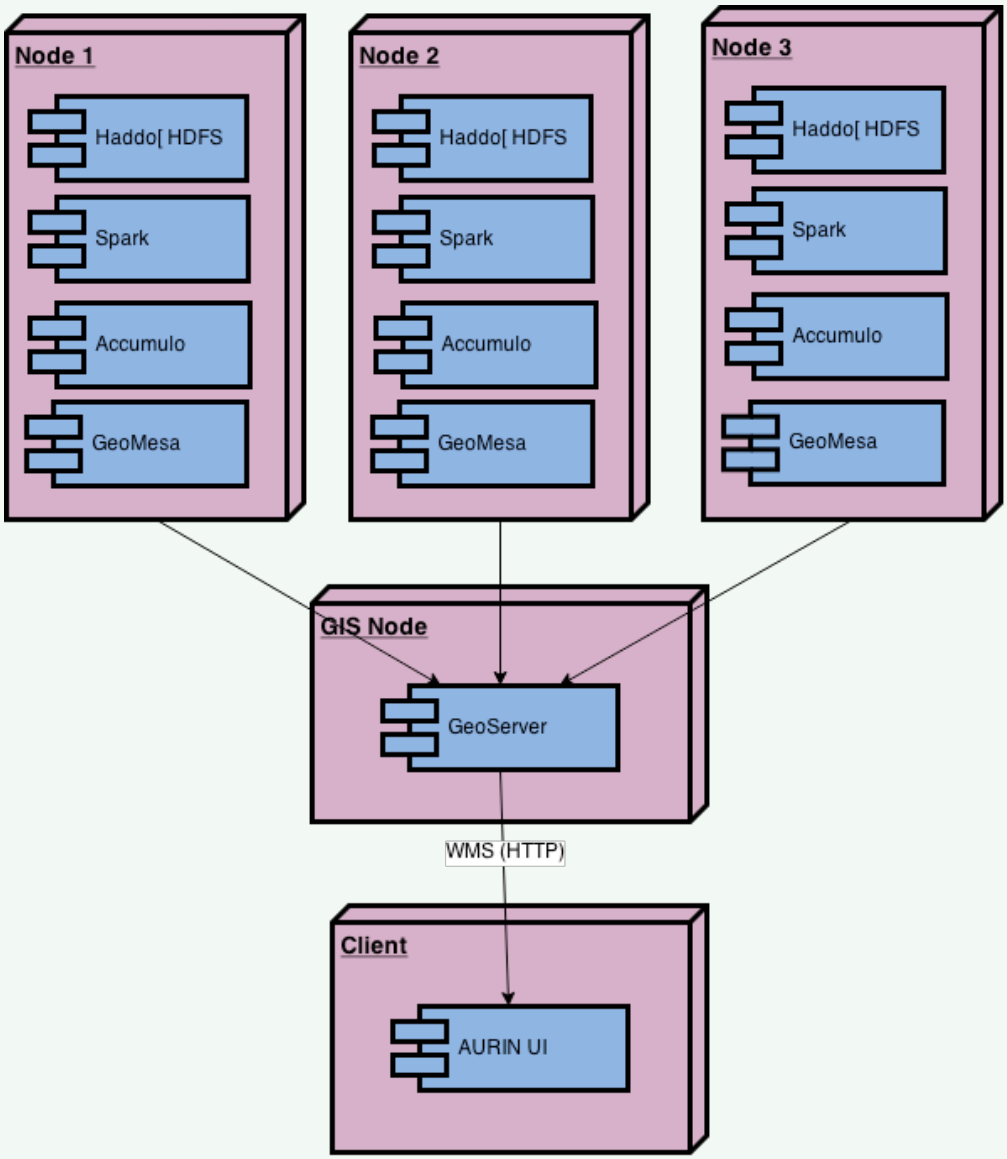
Welcome to your new documentation space!

A few technical notes on the projects.

Architecture

1. Apache Hadoop's HDFS as file system on a small cluster (5 ?)
2. Apache Spark as distributed computing engine
3. Apache Accumulo to store Tweets and Weather data
4. GeoMesa as geo-spatial processing engine (it runs on Spark and can store data in Apache Accumulo)
5. GeoServer as WMS server (to serve maps to the user interface)
6. AURIN's Portal as user interface

A draft of the architecture as UML deployment diagram:



Data Flow

The data are collected and processed using two flows

- **Analytical data flow**: building the training sets and machine learning execution)
- **Streaming data flow**: collecting weather data and Tweet to categorize Tweets for display on a map using AURIN UI

See the Data Flow Diagram for more details.

Training sets

1. Weather data for 2014 (must be the same variables we can scrape from their website)
2. Tweets from 2014 stored in HDFS. Some cleaning in order though:
    a. Tweets are to be re-parsed to get rid of stop-words, hash-tags, and, more generally, to have them parsed sensibly
    b. Tweets with no location or time-stamp, or with no meaningful text (say, a Tweet has only has-tags in it) have to be filtered out of the training set
    c. A sentiment index has to be computed on the re-parsed and filtered Tweets (see point 2)

Analytical data flow

1. A view in CouchBase is created to list some information from harvested Tweets
2. Data from CouchBase are transferred to an HDFS dataset, with HDFS-friendly formatting
3. A Spark process is performed to tokenize texts, get rid of stop-words, stem texts (compute sentiment ?) and write outputs to Accumulo
4. Historical weather data are scraped from the BoM and written to Accumulo

Analysis (Machine Learning)

1. Execution of topic modelling analysis using Latent Dirichlet Allocation on the Tweets (using Apache Spark MLlib)
2. Topics are baptised by an human being
3. Computation of a model correlating sentiment and weather on the training set (Decision Tree available in Apache Spark MLlib?)
4. The results of the two computations above are:
    a. A list of topics (cluster of words) the are more common in Tweets
    b. A matrix of probabilities linking words to topics
    c. A set of rules to predict sentiment based on some Weather parameters

Streaming data flow

1. Daily weather forecast data are harvested via web-scraping from the BoM's website and stored in Accumulo
2. New Tweets are harvested, parsed, assigned a topic, and the expected (based on weather) and observed sentiment is computed

Presentation (User Interface)

1. Connection of GeServer to GeoMesa/Accumulo
2. A WMS service returning density map of the expected sentiment given a time-stamp (day);
3. A WMS service returning a density map of the observed sentiment given a time-stamp (day);
4. A WMS service returning a density map of Tweets, given a time-stamp (month) and a topic (chosen amongst a list);
5. An extension of the "overlays" in AURIN's UI that, upon selection, ask the user for required parameters (time-stamp, and, if applicable, topic) and add a WMS map layer to AURIN.