Progress Report

Progress:

- **Warm-up study and finalizing the dataset and framework required.**

We aim to analyze the dataset of music records and extract useful information from it to support ad-hoc querying. We plan to use the dataset provided by Lab Rosa at Columbia University [1] to support answering ad-hoc user queries. Since there can be infinite variants, we plan to limit the custom queries and variants to facilitate ad-hoc querying within the scope of the project.

There are a couple of ways to implement this. We have studied the approach and have explored options to do it in an optimal way - We will leverage the Data querying prowess of HIVE and Map-Reduce framework. One major hurdle to look for in this approach is the latency. Depending on how the latency turns out to be, we plan to use the same or Impala for fast querying. Apart from this, if time permits, we will be working on the GUI part of the project.

- **Figure out project specifics and understand the framework, namely: Map Reduce and Apache Hive.**

We have set up a mini cluster in AWS EC2 with HIVE and Hadoop configured. We worked on some basic sample queries using HIVE to understand the framework. We are yet to test the querying on large datasets. We are currently processing a subset of the original dataset of music records.

We have also selected a set of queries that can be used (because of the availability of required fields in the dataset, we cannot facilitate a completely ad-hoc system) and worked on identifying the parts of dataset that can be used to answer such queries. So we have selected required subset and narrowed down the information required depending on variants of queries.

- **what do you think will be your contributions**

Use of Big data tools to facilitate ad-hoc queries of large database like that of songs (million songs DB). Also, fast processing of the big dataset to support on the fly user requests specific to music domain.
how you will be experimentally (or otherwise) evaluating your projects

We plan to check the efficiency by the latency time of response for each query. We will also vary the sample subset size and evaluate if the response time is reasonable for interactive big data querying.

We can’t think of ways to verify the accuracy of the results, except by manually verifying. We have thought of a way to break each query into parts and sub-query the database, combine the results and check the results for accuracy. However, this may not be fool-proof.

whether you can complete in time and how you will scope the project if you cannot

We have kept design of GUI as a plan only if we can complete the ad-hoc query part of the project. Otherwise we would not be creating the GUI and connectivity to back-end. Since the main goal of the project is to use big data tools to do analysis on the songs database, we would focus on that as the main scope and then build upon it on completion.

what are the major difficulties, what has been straightforward

1) Setting up the stack and the basic data query is fairly straightforward - HIVE, Hadoop are available as Apache Projects.
2) Importing S3 data into HIVE table is not trivial, due to some dependency problems which is not properly documented (HDF5 extension, etc).
3) Understanding the dataset for custom-queries. Even though the project is concerned with ad hoc queries, there might be components of queries that may not be present in the union of dataset given.
4) We are yet to figure out the back-end design and how it can be doing using HIVE. On executing such queries on a large dataset, we are not aware of the difficulties that might come up as opposed to the smaller samples that we have tried the queries on. HIVE might not be the most optimal solution for realtime processing, hence the data
must be pre processed and then be queried real time. We will be working on how to improve the latency of interactive queries.

References: