**Project Proposal**

**Introduction**

This document presents the first version of our project proposal for the *2016 Projects on Big Data Software* class. It outlines scope and objectives of the project and gives an overview of project team, used technologies, datasets and provisional project schedule.

**Project working title**

Parallelization and NoSQL Benchmarking

**Project description**

The project team plans to build a distributed *Extract Transform Load (ETL)* tool / environment using Spark that will process large data input files and populate data into NoSQL Open Source databases such as Cassandra or MongoDB. The team will implement and test performance statistics in the context of the following environments:

* Single node
* multiple nodes (2-4)

The project team will also use Cassandra and MongoDB as targets and benchmark how these NoSQL databases perform under the same load conditions. Various read/write queries will be used to conduct studies on both Cassandra and MongoDB (see *Appendix* for details).

The following high level execution steps will be performed:

1. Install and configure Spark on single node and multi-node cluster.
2. Install and configure MongoDB and Cassandra on single and multi-node cluster.
3. Write Python code to read data from input files and populate NOSQL tables.
4. Query data on MongoDB and Cassandra on single and multi-node cluster to benchmark the effects of parallelization.
5. Query input files using SPARK SQL with Spark running on single node and multi-node clusters.

**Project objectives**

Main objective: To conduct a benchmark study of parallelization for different NoSQL databases with special emphasis on the following aspects:

* Distributed ETL and how parallelization helps and performs under various conditions
* Performance Testing and Benchmarking of various Open Source NoSQL databases

**Project Team**

* **Srinivasa Gorijavolu** (Project Lead), IU Username: **sgorijav**, IU Github: **sgorijav**
* **Michael Mzyk**, IU Username: **mmzyk**, IU GitHub: **mmzyk**
* **Satish Patharkar**, IU Username: **Spathark**, IU Github: **Spathark**

**List of Technologies**

* Spark
* Cassandra
* MongoDB

**Compute Resources**

OpenStack in FutureSystems

**System Requirements**

* Size: 3 VM instances
* Storage: 6 GB

**List of DataSets**

Population from the college data set or Dummy Insurance data

**Development language**

Python will be used wherever necessary as development language.

**Schedule**

* March 7th : Initial Meeting
* March 12th : Project proposal creation
* March 15th : Project proposal finalization
* March 16th : Project proposal submission
* March 24th: Architecture design : First run
* March 28th: Architecture finalization
* April 1st: Develop modules, test run
* April 7th: First run
* April 15th: Second Run
* April 21st: Final Run
* April 25th: Upload Results
* April 28th: Final Submission
* TBD: Demo

**Appendix: Benchmarking Queries**

* Baseline Results from Single node cluster

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Spark | MongoDB | Cassandra |
| Populate data |  |  |  |
| Update multiple Records |  |  |  |
| Update single Record |  |  |  |
| Delete one record |  |  |  |
| Delete multiple records |  |  |  |
| Range Queries |  |  |  |
| Read records with single condition |  |  |  |
| Read Records from table with multiple conditions |  |  |  |

* Baseline Results from Multi node cluster

|  |  |  |  |
| --- | --- | --- | --- |
| Item | Spark | MongoDB | Cassandra |
| Populate data |  |  |  |
| Update multiple Records |  |  |  |
| Update single Record |  |  |  |
| Delete one record |  |  |  |
| Delete multiple records |  |  |  |
| Range Queries |  |  |  |
| Read records with single condition |  |  |  |
| Read Records from table with multiple conditions |  |  |  |