Project 2- Proposal

Implementation of Search Engine using Hadoop framework

Aparna Shikhare Poonkodi Ponnambalam Shubhangi Rakhonde

Problem Statement

This project aims to implement an efficient search engine using Hadoop framework. Today, search engines use combination of complicated algorithms to provide faster and accurate results in response to the user queries. PageRank[5] is one of the most important ranking signals.

The search engine consists of two main parts:

- **Indexing stage** - This is an offline process where all the web pages are indexed with the most relevant words. This is not a time critical process as, this is done ahead of time. PageRank is computed at the indexing stage[1].
- **Serving stage** - In this stage, the user queries the search engines and the relevant documents are retrieved from the index(created above), ranked and reordered so that the most relevant pages are at the top. Pagerank[4] is used to rank the pages. This stage is the most time critical, as the user is waiting for search results.

Since the data dump is huge, we use the Hadoop[7] framework for faster computations and results.

Survey of Previous Work

Due to huge amount of data being generated in today's era, storing such huge amount of data and transferring such data across network for processing using traditional database systems has become impractical. As a result, Hadoop was created to handle processing of huge amounts of data. It distributes the data across various nodes and facilitates parallel and cost effective processing. Hadoop design is based on Google's GFS (Google File System) and MapReduce framework[6], thus Hadoop is also made up of these two primary components namely HDFS (Hadoop Distributed File System) and MapReduce. HDFS is a distributed, scalable and portable file-system written in Java for Hadoop framework. Mapreduce is a programming model for large-scale data processing [2]. HDFS lacks the random read/write capability. It is good for sequential data access[3]. Hence, HBase is well suited if the data access is random. It is a NoSQL database that runs on top your Hadoop cluster and provides you random real-time read/write access to your data
Some of the salient features of both the systems:
HDFS

1. Optimized for streaming access of large files.
2. Follows write-once read-many ideology.

3. Doesn't support random read/write.

HBase

1. Stores key/value pairs in columnar fashion (columns are clubbed together as column families).
2. Provides low latency access to small amounts of data from within a large data set.
3. Provides flexible data model.

We will be making use of hadoop framework and HBase for efficient, parallel and cost effective computation of inverted indexes and pagerank to provide accurate search results.

Key features

- The search engine handles two types queries: One word query and free text queries. One word queries are one which contain a single word and free text queries are which contain series of words separated by words.
- Retrieve the top ten articles in the order of importance.

Assumptions

- The dataset has large set of web pages with hyper links to each other

Challenges

- Handling “Big Data” amount of dataset is a challenge in itself. We plan to use Hadoop/HBase to tackle this problem.

References

1. Aren. May 2011. How to implement search Engine Part- 1.Available online at http://www.ardendertat.com/2011/05/30/how-to-implement-a-search-engine-part-1-create-index/
2. Bappalige S, An introduction To Apache Hadoop for big data. http://opensource.com/life/14/8/intro-apache-hadoop-big-data, 2014
3. Mittra D, Apache Hadoop http://wiki.apache.org/hadoop, 2014
4. Page, Lawrence, et al. "The PageRank citation ranking: Bringing order to the web." (1999).
5. Langville, Amy N., and Carl D. Meyer. Google's PageRank and beyond: the science of search engine rankings. Princeton University Press, 2011.
6. Dean, Jeffrey, and Sanjay Ghemawat. "MapReduce: simplified data processing on large clusters." Communications of the ACM 51.1 (2008): 107-113.
7. Shvachko, Konstantin, et al. "The hadoop distributed file system." Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on. IEEE, 2010.