15-440: Project 4

Clustering Data Points and DNA Strands Using MapReduce

Project posted on: November 28, 2012

Due date: December 15, 2012

Intended Learning Outcomes

This project applies the theory of the distributed programming model, MapReduce. The learning outcomes of the project are two-fold:

1. Apply MapReduce to a popular real problem, namely cluster analysis using K-Means algorithm.
2. Compare and contrast MPI-based (from Project 3) and MapReduce-based implementations of K-Means in terms of performance and development effort.

Project Objectives

The overall goal of this project is to get a clear understating on how different parallel implementations for the same algorithm using different programming models can provide different performance and entail different development efforts. Students will conduct and analyze some scalability studies on various degrees of parallelism and data set sizes. The project will provide our students with a practical experience augmented with a methodology for solving clustering problems (and alike) on a distributed system using MapReduce.

Implementation Guidelines

In this project, you will provide a MapReduce implementation for K-Means with two types of data sets, a data set of data points and a data set of DNA strands (datasets same as in Project 3). Please use the datasets you generated in Project 3 to write, run and test your K-means MapReduce-based implementation. For a complete explanation of the K-Means algorithm, please refer to the write-up of Project 3.

For this project, we will provide you with the following:

- A cluster of 4 VMs, each with 4 vCPUs, 8GB of RAM, 60GB of storage and 64-bit Fedora 15 OS.
- Hadoop 0.20.2 installed, configured and ready to use on your cluster.

Experimentation and Analysis

Please conduct and provide the following:

- A comparison between your 3 different K-Means implementations (the sequential and the MPI-based ones from Project 3 and MapReduce-based one) in terms of performance and development effort.

- Two scalability studies on:

    - The number of processes for your MPI version developed in Project 3 with a fixed data set of 2D data points. Specifically, use 2, 4, 8, and 12 processes.
    - The number of points in your data set of data points with a fixed number of processes (say 8) and 1 reducer for your MPI and MapReduce applications, respectively. Specifically, use 10 million, 20 million, and 30 million data points.

- A discussion on:

    - Your experience in applying MapReduce to the K-Means clustering algorithm.

- Your insights concerning the performance trade-offs of MPI and MapReduce with K-Means.
- Your thoughts on the applicability of K-Means to MapReduce.
- Your recommendations regarding the usage of MapReduce for algorithms similar to K-means.

Final Deliverables

As final deliverables, you should submit:

1- An archive containing a fully tested and debugged code for your MapReduce K-Means implementation.

2- An article with a maximum of 5 pages (similar to research articles) that presents your solution, findings, observations and analysis.

Handing In the Project

Submit your documents and code to CMU Autolab.

Late Policy

- If you hand in on time, there is no penalty (duh!).
- 0-24 hours late = 25% penalty.
- 24-48 hours late = 50% penalty.
- More than 48 hours late = you lose all the points for this project.

NOTE: You CANNOT use your grace-days quota for this final project.