Report on Evaluation Programs of Data Skew Performance Debugging

Research 199

Student: Katherine Kang

UID: 304808025

Date: Aug 10, 2018

Project Description:

This project is intended to focus on performance debugging of big data systems, specifically data skew and build on top of the existing Big Data Analytics and Debugging project. The core is to use Titian, a data lineage/data provenance tool to track skewed input and output data. My current focus is on benchmarks and evaluation programs. I investigated the cause of skewed performance across data sets and real-world applications that demonstrate the problem. I looked for different types of applications in Apache Spark that have skewed performance and analyzed their appropriateness to serve as a motivating example or evaluation programs. After investigating some potential candidates, PageRank was chosen as the motivating example. In the meanwhile, I got familiar with the Scala syntax and Apache Spark API. It's necessary to understand their basics in order to complete the tasks.

The code and data sets used for the following programs can be accessed at https://github.com/nkang0503/Performance-Debug-Benchmarks. Among these programs, PageRank is the most promising motivating example for the project, and others are test programs that can be used to evaluate the debugging tool.

1. PageRank

Description:

PageRank is a well-known algorithm that naturally demonstrates data skew, so it would be a good fit as the motivating example. PageRank is the first algorithm used by Google to rank websites in their search engine results. PageRank measures the importance of a website page by counting the number and quality of inbound links to the page to determine how important the website is. The idea is that more important websites are likely to receive more links from other websites.

At the first step, every web page is assigned an equal probability number. Then, in each iteration, a web page's importance is calculated based on the weights of inbound links.

Sample record of input data:

url inbound_url

Data Skew Characteristics:

The groupByKey operation in the program will group a web page's inbound links together and the rank of each web page is the sum of its inbound links' contributions. In the data set sample, "pagerank_data.txt", I gave an example of data skew: there are two-way links between the central web page and the other three web pages. In this case, the rank of the central web page will take three times longer than the other three. Such data sets can be generated at a large scale with one central web page or multiple web pages as hot spots.

Location of Larger Data Sets:

https://github.com/nkang0503/Performance-Debug-Benchmarks/blob/master/PageRank/hollins_num.dat (waiting to be tested)

2. Word Count

Description:

WordCount takes in an input of data file and output the number of occurrences of each word appeared in the file. The map operation emits <word,1> tuples, and reduce adds up the counts for each word from all map tasks and outputs the final count.

Data Skew Characteristics:

During the map phase, a sleep command is added before emitting the tuple, so that the long lines with many words will take much longer to process than the shorter lines. This is similar to the effect of having some words that needs additional processing in a line.

Location of Larger Data Sets:

ScAi cluster HDFS /clash/dataset/bigsift

Or four Wikipedia datasets of size 50GB, 140 GB, 150GB and 300GB at:

https://engineering.purdue.edu/~puma/datasets.htm

3. Weather Analysis

Description:

Weather Analysis converts the zip code of the input snowfall data to state and calculates the difference between the maximum and the minimum snowfalls on each day for every state.

Sample record of input data:

18090,30/12/2012,1902 mm

18090,30/12/2011,10.431476 ft

Zip code, data/month/year, snowfall measurement

Data Skew Characteristics:

A sleep command is executed for any input that has a snowfall smaller than 500 millimeters. As a result, the input data with small snowfalls will take significant longer than other input data.

Another modification that could create data skew is to add a sleep command for any snowfall data that have some other units rather than millimeter.

Both of these cases demonstrate the situation that a group of input data with some specific characteristics need additional processes, thus result in longer processing time.

Location of Larger Data Sets:

ScAi cluster HDFS /clash/dataset/bigsift

4. Inverted Index

Description:

The program takes a list of documents as input and generates word-to-document indexing. The map operation emits <word, document_id> tuples with each worsd emitted once per document_id. The reduce operation combines all tuples on key <word> and emits <word,list of (document_id)> tuples after removing duplicates.

Data Skew Characteristics:

A sleep command is added for every word in a line regardless if there are duplicates. Because the length of the lines in the input data file varies greatly, some lines take significantly longer time to process. The sleep command exaggerates the difference.

Location of Larger Data Sets:

ScAi cluster HDFS /clash/dataset/bigsift

Or four Wikipedia datasets of size 50GB, 140 GB, 150GB and 300GB at:

https://engineering.purdue.edu/~puma/datasets.htm

5. Student Data Analysis

Description:

The program takes an input of student information that can be generated by a script in the repository and outputs the average age of students by grade.

Sample record of input data:

ytbipjjsy vljsfjft female 18 0 IndustrialEngineering

first_name last_name gender age grade major

Data Skew Characteristics:

This program will demonstrate data skew if there are more students in a particular grade. For example, we can accomplish this by increasing the chance of generating students in grade 0 in the date generation script.

Another possibility of skew can be created by adding additional operations for students in a particular major, for instance, computer science.

Location of Larger Data Sets:

Scalable data generation script: https://github.com/nkang0503/Performance-Debug-Benchmarks/blob/master/StudentInfo/dataGenerator.py

6. Term Vector

Description:

This program determines the most frequent words in a set of documents and is useful in the analyses of a host’s relevance to a search. The map operation emits <host,termvector> tuples where termvector is itself a tuple of the form <word, 1>. The reduce operation discards the words whose frequency is below a certain value, sorts the rest of the list per key in a descending order with respect to the counts and emits tuples of the form <host, list of termvector>.

Data Skew Characteristics:

A sleep command is added for each word that has appeared before, so a line with many old words will be much slower to process than other lines.

Location of Larger Data Sets:

ScAi cluster HDFS /clash/dataset/bigsift

Or four Wikipedia datasets of size 50GB, 140 GB, 150GB and 300GB at:

https://engineering.purdue.edu/~puma/datasets.htm

Future Works

1. Choose a few more popular programs and modify them to fit as evaluation programs.

2. Make a data generation script for PageRank that can generate large scale input data with a specified number of hot spots.

3. Many of the current programs are planted sleep command in order to show data skew behaviors. Continue to work on the existing programs and see if we could come up with some situations that happen in real life.