SPARK VERSUS FLINK: PEFORMANCE COMPARISONS IN BIG DATA ANALYSIS FRAMEWORK

February 18, 2019

Shayan Manoochehri 27232438
Xueying Li 40036265
SOEN 691 Big Data Project Proposal

1 Abstract:

Spark and Flink are two Apache-hosted data analytics frameworks that facilitate the analysing of the big dataset. The in-depth understanding of the underlying architecture choices are important for increasing the performance of processing data in terms of different dataset. This project is aimed to justify the performance of Spark and Flink by directly evaluating their performances on processing streaming dataset and static dataset. Different parameter configurations will also be considered for evaluating. Additionaly, other aspects including machine learning, CLI, memory management will also be briefly compared.

2 Introduction:

Streaming computing and real-time analytic are two important fields in the topic of big data. When it comes to streaming data, there is no avoiding the importance of the two most powerful data processing engines: Flink and Spark. Basically, streaming data means data that comes in continuously, and the typical use case is querying tweets in real time. Other streaming data examples are traffic, stock and weather etc. The way of Flink and spark works is that they do not persist their data to storage but keep in-memory and use it right now[1].

From the aspect of the designs, The Apache Flink is specifically built for streaming while Apache Spark is designed as a replacement of the batch-oriented Hadoop system but Apache Spark Streaming is included. It seems that Flink and Spark are the same, however the main difference are that Flink is build from the ground up for streaming processing while Spark added streaming to their products, which was built ,like Hadoop, to run over static data sets. And in Flink, if the input data stream is bounded, the effect of batch processing results naturally[2].

Technically, for streaming data,Spark continuously loop to divide streaming data into micro batches while Flink takes a checkpoint on streaming data to mark it as a finite sets. That means streams are not necessarily opened and closed. And as we could see,it is always true that processing live data will always cause a latency.

In other words, the reducer operation will run on a map dataset which was created a few seconds ago. It is noted that Flink

For running over static data sets, flink process data the same was no matter it is finite or not. But for Spark, Discretized Streams(Dstream) are used for streaming data and Resilient distributed dataset(Rdd) for batch data.

Additionally, there are various differences between Spark and Flink including Memory Management, Machine Learning, CLI, Cluster Operations and etc.

In this project, We will provide direct and in-depth comparison between Spark and Flink in the aspects of processing streaming dataset and static dataset respectively. We will also compare the other aspects such as memory management, CLI etc.

3 Materials and Methods:

Firstly, we will introduce a methodology[3] to understand the performance in Big Data analytic frameworks. Specifically, we will:

- Set constraints on streaming dataset(static dataset) and compare the problem fixed on each node.
- Set constraints on node and compare the time taken to process the same dataset (streaming and static respectively).
- Constraints the number of nodes, respectively compare the same processing of streaming and static dataset in the aspect of Resource Usage.

Secondly, we will briefly introduce the dataset. Our datasets will access from Twitter API. Twitter API functions could be used to acquire a Twitter dataset including ① Retrieving tweets from the user timeline, ② Searching tweets and ③ Filtering real-time tweets. So the following is our Streaming Dataset:

- Tweets in twitter.

Because the access to historical tweets is extremely limited. We will use GW Libraries to retrieve historical tweets, which are collections on number of topics including Congress, the federal government, the news organizations.So the following is our Static Dataset:

- Historical Tweets in twitter.

0pt 0.1

References

[1] http://www.developintelligence.com/blog/2017/02/comparing-contrasting-apache-flink-vs-spark/ *Comparing and Contrasting Apache Flink vs. Spark.*

[2] https://hackernoon.com/in-search-of-data-dominance-spark-versus-flink-45cefb28f377 *In Search of Data Dominance: Spark Versus Flink.*

[3] Ovidiu-Cristian Marcu,Alexandru Costan,Gabriel Antoniu,Mar ıa S. Pe rez-Herna ndez `Spark versus Flink: Understanding Performance in Big Data Analytics Frameworks` 2016 IEEE International Conference on Cluster Computing.