CprE 419 Lab 9: Stream Data Processing Using Flink

Department of Electrical and Computer Engineering
Iowa State University

Purpose

In this lab, the goal is to learn how to use Apache Flink that is run of top of the YARN resource manager and learn how to solve some interesting problems in streaming fashion.

During this lab, you will write programs with pipelined jobs to:

- Using Flink platform on top of YARN
- Some real world problems that can be solved using Flink

Submission

Create a single zip archive with the following and hand it in through Canvas**:**

- Screenshots of your results for each individual experiment.
- Commented Code for your program. Include all source files needed for compilation and make sure it compiles successfully. Make sure you output the results to a specified folder of each experiment.

Flink resources

Apache Flink documentation:
https://ci.apache.org/projects/flink/flink-docs-stable/
Running a Flink job on YARN:
https://ci.apache.org/projects/flink/flink-docs-release-1.4/ops/deployment/yarn_setup.html
Batch Flink sample problems:
https://ci.apache.org/projects/flink/flink-docs-release-1.4/dev/batch/examples.html
Streaming Flink sample problems:
https://github.com/apache/flink/tree/master/flink-examples

Flink Installation and execution

Download the Apache Flink package from the following link. Note, we recommend to download the version 1.4.2 since the experiments of this lab are tested on Apache Flink 1.4.2.
http://flink.apache.org/downloads.html

After having downloaded the package, unpack it and move it to your home directory (or any sub-directory of your choice) in the cluster (hpc-class.its.iastate.edu). You can download the flink package in your local machine, then copy it to the cluser, using the following command
scp flink-1.4.2 user_name@hpc-class.its.iastate.edu:/home/user_name/

Otherwise, you can download it directly from the cluster, using following commands:

- Download the Flink:
```
wget http://mirrors.advancedhosters.com/apache/flink/flink-1.4.2/flink-1.4.2-bin-hadoop26-scala_2.11.tgz
```

- Unpack the package:
```
tar -xvzf flink-1.4.2-bin-hadoop26-scala_2.11.tgz
```

- Remove the package (after unpacked it):
```
rm flink-1.4.2-bin-hadoop26-scala_2.11.tgz
```

- Add Flink to the PATH environment from the namenode(hpc-class-hdp01) (optional):
```
export PATH=$PATH:/home/<Net_ID>/flink-1.4.2/bin
```

To test Flink, run the following command:
flink run flink-1.4.2/examples/batch/WordCount.jar

In order to be able to compile a Flink program in Eclipse, you need to have the following dependencies in your maven project:

```
<!-- https://mvnrepository.com/artifact/org.apache.flink/flink-java -->
<dependency>
	<groupId>org.apache.flink</groupId>
	<artifactId>flink-java</artifactId>
	<version>1.4.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.flink/flink-core -->
<dependency>
	<groupId>org.apache.flink</groupId>
	<artifactId>flink-core</artifactId>
	<version>1.4.2</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.apache.flink/flink-streaming-java -->
<dependency>
	<groupId>org.apache.flink</groupId>
	<artifactId>flink-streaming-java_2.11</artifactId>
	<version>1.4.2</version>
</dependency>
```

Make sure you have JVM (>=1.8.x), Maven(>=3.1.x). Sample Stream Word Count implementation with its pom.xml configurations is provided on Canvas. It is run in a streaming fashion and reads the file from HDFS. Run the following command:
flink run –m yarn-cluster –yarncontainer 1 JAR_NAME.jar
In order to check your output result, you may see it in the logs by running the following command: **yarn logs –applicationId YOUR_APPLICATION_ID**

Experiment 1 (30 points) Word Frequency Distribution

The WordCount program counts the number of occurrences of every distinct word in a document. Download the file WordCount.java and compile the program. You should have the initial results for the wordcount program.

Write a Flink program to count the frequency of every distinct word from a given stream. Output the word frequency distribution sorted by the frequency in descending order. Read the input stream from HDFS and the dataset is **/cpre419/shakespeare**. You may output your results to the log, HDFS or file.

Bonus 5 points: Output top 10 most frequent words along with their frequencies for each 10 lines of input.

Experiment 2 (40 points) Bigram Counting

A bigram is a contiguous sequence of two words within the same sentence. For instance, the text “This is a car. This car is fast.” has the following bigrams: “this is”, “is a”, “a car”, “this car”, “car is”, “is fast”. Note that the two words within a bigram must be a part of the same sentence. For example, “car this” is not a bigram since these words are not a part of the same sentence. Also note that you need to convert all upper case letters to lower case first. The sentence ends with “.”, “?” or “!”.

Write a Flink program to count the frequencies of bigrams from a given stream. Read the input stream from HDFS and the dataset is **/cpre419/shakespeare**. You may output your results to the log, HDFS or file.

Bonus 5 points: Output top 10 most frequent bigrams along with their frequencies for each 10 lines of input.