HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Department of Computer Science

Spark Code Camp Project Report

Rapid prototype using Spark streaming & Spark SQL API

Maninder Singh (maninder.singh@cs.helsinki.fi)
Ayesha Ahmad (ayesha.ahmad@helsinki.fi)
Md. Mesbahul Islam (mdislam@cs.helsinki.fi)
Department of Computer Science
University of Helsinki
Finland.

1 Project Objectives

In this project our aim was to familiarize ourselves with the Apache Spark cluster computing environment. We decided to focus on Spark Streaming and Spark SQL. Twitter was selected as the streaming data source and our objective was to collect live tweets and analyse them to find out popular hashtags, discover the tweet frequency per location, and to discover tweeting trending over time.

2 Project Overview

Several projects were developed in order to explore the different features of Spark Streaming and Spark SQL. The projects included: analysing the live twitter stream from the last 30 seconds and displaying the most popular hashtags in a dynamically updated chart. Exploring different ways to store and process the streamed data, for example, using window-based streaming vs direct streaming, extracting stream information using the Twitter APIś, or in JSON format, and storing streams in files or in SQL tables.

2.1 Spark Streaming Project

Window Based Streaming Spark Streaming provides a window based streaming. Every time the window slides over a source DStream within a specified interval, a set of RDD's are collected and operated to filter required Twitter tweets. We processed live feed from Twitter using a sliding window to collect top hashtags. These top hashtags are organized dynamically into a live graph. The implementation is available in following github link: github/SparkStreming

Direct Streaming In our project we used spark streaming to sample twitter's live data. We can fetch the live tweets or User information from the stream periodically. From the live feed we can format the data according to our necessity and then we can manipulate the data. For example, we can get the User information like User's name, location, followers count, recent tweets, friends count etc. The inplementation is available in following github link: github/spark-streaming-twitter-hashtag

2.2 Spark SQL Project

In addition to processing live data streams, Spark Streaming can also be used to build large datasets. Data can be written to filesystems or databases to be analyzed later. We were able to push the live data to both the filesystem and to Spark SQL tables. The project that stores tweets to the filesystem and in SQL tables can be found at: https://github.com/ayef/SparkStreamTwitterFeed.git
Another project was developed to test out how SQL statements could be run on the tweet data stored in files. This project is available at:
https://github.com/ayef/SparkSQLProject.git

3 Open–Source Software Stack

The streaming application is developed on an open source software stack. The linux environment is used to organize different technologies as a layered structure application. We are using Apache Spark[5], Spark SQL, Twitter4j[7], Akka[6], Chart.js[2] and Socko[3] web server.

Apache Spark We are using Apache Spark as an open source software for large-scale data processing. Apache Spark provide both core and streaming apiś. The application focused on streaming using Twitter streaming as a data source.

Akka Akka is a toolkit and runtime for building highly concurrent distributed event-driven appliation on the JVM. Akka provide actors which are very lightweight concurrent entities which are used for handling business logic. Akka actors process messages asynchronously using an event-driven pattern.

Twitter4j Twitter4j is a Java library for Twitter API. It provide utilities to handle Twitter stream data.

Socko web server A Scala based web server is used as a http server. It is powered by Netty networking and Akka processing capabilities. Socko web server serve static HTML5 files to our busines logic implemented in Akka.

Spark SQL Spark SQL exhibit qualitie to load and query data from different souces. It provide utilities to save and process data in structured format. The Spark SQL queries directly access and process distributed dataset(RDD).

Chart.js Chart.js is an open source HTML5 charts. It is based on JavaScript charts. It is simple, reponsive and flexible way to represent data. It provide variety of options to visualize data.

4 Challenges & Learning

The streaming application developed during the course comes up with lot of challenges and domain learning. As Spark APIś are completely new to everyone in a Team, it took significant time to explore and learn. We explored Streaming APIś available in Apache Spark. The challenges can be organized in different context. Some are decision challenges and other are related to learning curve bind to different open source software stack. Short learning curve played important factor while selecting software stack.

Streaming Source There are variety of sources available to analyse and process. We have chosen quantity and quality of the streaming data, therefore, we selected Twitter as a data source to perform analysis in application.

Build environment As we are using Scala as programming language, we used SBT as our build environment. SBT is an interactive build tool for Scala. We found it as an easy tool compared to Maven. The learning curve is quite small in SBT build tool.

Scala Middleware As per our project require interaction between steam content and web server, we found out Akka actors as a useful entity. The event-driven asynchronous pattern was useful for our application.

Web Server There are variety of containers available as a web server. We were looking for a Scala based simple container which can iteract with Akka actors. We explored Socko web server useful for our project.

Graph There are few libraries which are simple and open source. Chart.js was first choice due to its impressive features and wide variety of graphs.

Storage Large stream of data need to store in a file system or in a database. Using a filesystem it may be difficult to handle a massive stream of data. Considering other software stack, we explored Spark SQL as a useful tool and compatible with Apache Spark stream content.

5 Results

We were able to fulfil all of the project objectives. Specifically, we discovered the most popular hashtags in the last n seconds of the live Twitter stream using sliding window streaming, plotted a dynamic graph with live feeds from Twitter, generated a large dataset of tweets in text files and in Spark SQL tables, ran some SQL statements on the tweet dataset, and used Actor-based interaction to connect the streaming content and a Web Server.

References

[1] AMPLab UC Berkeley. Amp camp two - big data bootcamp strata 2013, 2013.

[2] Nick Downie. Chart.js, 2013.

[3] Vibul Imtarnasan. Socko, 2014.

[4] University of Berkeley. ampcamp: Stream processing with spark streaming, 2014.

[5] Apache Spark. Spark streaming examples, 2014.

[6] Akka Team. Akka.

[7] Twitter4J Team. Twitter4j java library.

[8] Patrick Wendell. Sampling twitter using declarative streams, 2013.