Project Retrospective

December 9, 2019

1 Wrap-Up

1.1 Project Retrospective

Provide answers to the following questions and submit a PDF via Canvas. Be sure to answer the questions completely and explain your logic.

1.2 Q1:

You've now had the chance to work with both MapReduce and Spark. In your opinion, what are the pros and cons of both?

MapReduce Pros:

- Easy to implement.
- Relax the developer job.
- can (potentially) query live data.
- can (conceptually) be highly efficient at joining data sets that are identically sharded on the join key (the joins can be pushed down into the key-value store itself).

MapReduce Cons:

- The design are limited to the 2 phases of map and reduce that limited the developer jobs.
- We need to wait to see the result to ensure the process are correct.
- Full scans (the most common pattern for map-reduce) is most likely to be much faster with raw file system access.
- because of the better decoupling of computation and storage in the Map-Reduce model resulting from MR jobs is much easier.
- Unstructured data and need to write writable for the data structure.
- key-value stores are rarely arranged to have schemas optimized for analytics.

Spark Pros:

- High speed
- Ease of Use
- Advanced analytics
- Dynamic in Nature
- Multilingual
- Apache Spark is powerful

- Increased access to Big data
- Demand for Spark Developers

Spark Cons:

- No automatic optimization processes
- File management system
- Fewer Algorithm
- Small Files Issue
- Window Criteria
- Doesn't suit for a multi-user environment
- Lazy structure

[Rozita]: I personally like to work with Spark, because of easy setup and powerful APIs. However, Spark has poor documentation.

[Hiep]:

For me, i think Spark is easier to implement because even though with DataFrame API, it still nearly the same with SQL query. Don't need to care much about what will do in mapper, what will do in reducer, just write query.

[Alper]:

I love MapReduce and Spark both. Although I spent a lot of time with this project, I still like Spark project more because I learned Python, Pandas, data frame, sentiment analysis, readability analysis and etc. All of them were fun and nice. And it was also interesting to be able to run SQL queries in the Hadoop cluster!

1.3 Q2:

Was there something that you thought would be easy to implement in Spark but it turned out that it wasn't?

Yes!! Not the Spark but the project setup and running time. We did not expect to spends 5 days to run the dataset and keep on getting disconnection and error. The issue was solved using sampled data. In overall working with Spark was fun specially with Jupiter interface that makes it very user friendly.

1.4 Q3:

Were there any confusing or surprising aspects of working with Spark? Did you come across some functionality that made your life easier or the computations run faster?

[Rozita]: I personally enjoyed working with machine learning and Pandas library. The implemented API makes the machine learning concept very easy and understandable. Its data representation was extremely streamlined forms of data representation. It helps for less writing code and more work is done.!

[Hiep]: While writing in Spark, I found someway to make it faster, at first cache only the data I need because memory in orion is limited :). The way Spark allow to write UDF(User-Define Functions) make it easier to custom aggregate function. Long code of MapReduce is replaced by shorter code in Spark

[Alper]: I'm surprised when I checked sampled data. Sampled data was evenly distributed when I compare with original data. For instance, before sampling let say I checked for subreddit=relationships in 2013, comment count was 988268. After %1 sampling, I checked for subreddit=relationships in 2013 again, and comments count was 983200. Sampling was really impressive and made my life much more easier at the end of the project. Furthermore, I agree with Rozita that using Panda dataframe and ML libraries made life easier.

1.5 Q4:

1. Give a rough estimate of how long you spent completing this assignment. What part of the assignment took the most time?

Dear Mathew, we learn that your projects always took a long time to be completed. We have started from November 25, 2019 and it is about to be completed on December 8, 2019. Of course, we did not work full time on this projects but it was time consuming. The most time taken was running time to get result specially before sampled data provided.

2. What went well?

As long as, we learn how to work with spark and learn about reddit data structure, the rest went well. Just need time to complete it and make it as the report.

3. What didn't go well?

Use original data and sampling data was the toughest part and waste a lot of time. I believe, our resources are not enough for this volume of data with the current number of users. or in other words, Big Data course resources are not enough for Big Data course.

1.6 Team Questions

If you worked in a team, answer the following: 1. How did you decide to divide up the workload and coordinate with your team?

Each team member took the responsibility of one question at the time. Warm up questions have been done twice since Rozita joined in midst of the project. We had code review session that after any pull request or raising flag by the team member, the rest of member conduct code review or some discussion on WhatsApp group for better understanding. In overall, we have spent the following hours per person on the project:

Big Data Project1 : 54 hours per person Big Data Project2 : 30 hours per person Big Data Project3 : 52 hours per person

2. Describe the questions or deliverables completed by each team member:

Question	Persion In Charge	Status
W1	Hiep & Rozita	DONE
W2	Alper & Rozita	DONE
W3	Hiep & Rozita	DONE
W4	Alper & Rozita	DONE
A1	Hiep & Rozita	DONE
A2	Alper	DONE
A3	Hiep	DONE
A4	Alper	DONE
A5	Hiep	DONE
A6	Alper	DONE
A7-1	Rozita	DONE
A7-2	Rozita	DONE
A7-3	Rozita	DONE
Retrospective	All Team	DONE

We had code review session that after completing by one member, others act as code reviewer in a circle loop.