Big Data Platforms

Final Project

Small files in MapReduce jobs

Submission due to 09.02.2022

Background: MapReduce model was initially designed to process large amounts of data, where data usually persisted in large files. Initially MapReduce and HDFS were collocated. Data persisted in HDFS where each file is chunked into blocks, usually of 64MB size. Data partitioner of MapReduce framework will partition input dataset and decide on number of the map tasks to be launched across MapReduce cluster. When MapReduce job read a data from HDFS, then usually a single map task assigned to process a single HDFS block. In case data persisted in object storage, then MapReduce frameworks use "virtual" chunks and assign each map to read a particular byte range from the data. As example, when virtual block size is 64MB and data object is 80MB, then the first map will read 0-64MB offset from the file, while second map task will read the remaining offset, which is 64MB-80MB chunk. When the file size is below the chunk size, a single map will process entire file as is.

Problem statement: While the approach above is very efficient when the input files are sufficiently large, it has a major drawback when the input files are small, less below the chunk size. Both HDFS and MapReduce execution framework will suffer from different issues affecting their performance and effectiveness. There are various research papers on the topic of running MapReduce on small files. As example, the bibliography section contains two papers describing some of the issues in HDFS and in MapReduce. Both papers try to come with a solution to resolve small files issues in MapReduce.

Project definition: In this project students will explore challenges of executing MapReduce over small files persisted in the object storage. Students will define the problem and come with prototype solution.

Project scope

Project will be submitted as a pdf paper accompanied with a prototype code
The paper should contain the following sections

Template for the final report

Big Data Platforms

Small files and MapReduce

(Names)

(Submission date)

Abstract

Short description of the problem you are solving

Motivation and background

Describe what is MapReduce, object storage, how different object storage from HDFS and explain differences of running MapReduce over data in HDFS vs data in the object storage. (Use at least 2 different sources)

Small files problem

Define the problem when MapReduce executed over small files persisted in the object storage. Describe the relevant background of the issues running MapReduce over small files persisted in HDFS. Summary state of the art, background, what are issues in HDFS, what are issues in MapReduce and what kind of possible approaches suggested to overcome the problem. Use at least 4 different sources (2 from the bibliography and find 2 additional ones. Provide references to the relevant papers)

Our approach

Suggest two possible solutions to enable effective MapReduce analysis over small objects persisted in the object storage. Describe each of the solutions and emphasize strong and weak points in the proposed solutions. Compare both solutions to each other and conclude what should be preferable.

Prototype

Write Python code with a prototype of one of the proposed solutions. You can extend MapReduceEngine library from the home assignment 2 or code any other prototype

Next steps

Suggest next steps to the solutions you proposed

Conclusion

Short conclusion of the work you did

Bibliography

List of all sources you were using in the project. At least 6 different sources

Possible bibliography

1. https://www.cs.fsu.edu/~yuw/pubs/2015-NAS-Yu.pdf
2. https://mjeer.journals.ekb.eg/article_62728_c818f3f951476c6005647f9ba7364efd.pdf

Points on grading:

1. Extra points will be given to students who will create new GitHub project and upload their prototype there.
2. Extra points will be given to students who present deep solutions that address broad scope
3. The proposed solutions should address various aspects we learned during the course, like consistency, availability, fault tolerance, etc.
4. All the text students write should be their original and avoid as possible to copy text from other papers. In cases, when text need to be copied from another papers, make sure to reference what text is copied from other source and what is the source. Grade will be affected, If text is copied without proper reference.
5. Make sure you describe strong and weak points of your proposed solutions. Grade will be affected if there are additional weak or strong points than you managed to describe.