CS267 HW0 SP 19
High-performance data processing library PySpark

Nuochen Lyu
nlyu2@berkeley.edu

1. Bio

My personal website: www.lyulyulyu.com

I am a student in EECS Meng program in the division of data science and system. Currently, I am doing research in BAIR lab under Ph.D. Mitar and professor Dawn Song. We are building an AutoML system called Aika that could automatically find pipeline to solve existing dataset problem.

I received a B.S in computer engineering at the University of Illinois at Urbana. In 2017, I used to intern in Yahoo big data product team working on Spark source code. Since then I became very interested in the distributed system and big data processing system. It is very exciting for me to entered Berkeley, where the Spark is been invented. Now the Moore's is reaching its limitation and large scale multi-machine computing becomes the direction in the industry. In CS267, I would like to learn the key knowledge of building a parallel system, the bottleneck, hardware configuration, and its architectural details.

2. PySpark, application base on spark that surpasses Pandas

PySpark is a library for Spark, a powerful engine for big data processing created by UC Berkeley in 2010[1]. In the current day, most data analysis were conducted in Python using libraries such as Pands[2]. However, it is ususally limited locally in a single machine with a small dataset. If a scientist had made, for example, a Pandas based software. But now he would tackle a larger dataset. He/she could transfer the code by PySpark so it could take advantage of the distributed system in the cloud. With the power of cloud computing, the data workload could be divided and computed in parallel. Also because PySpark supports all standard Python library and C extension. could deploy their code in Spark with little modification. That is saying, move everything to the cloud! The program could be scalable to big data.

PySpark has achieved great speed through the Spark engine. It expands the data size that a single computer cannot process from gigabyte to more than petabyte level. Spark is built by Scala which is a functional programming language. Spark separate the job to the distributed system by its architecture on the top of RDD(Resilient distributed dataset). RDD is the basic data structure for Spark. RDD is the abstraction of the data which is actually divided and split into different machines. Operations such as map, count, and filter can be performed to RDD object. Computation is done under the Map-Reduce paradigm.[3] Intermediated results are performed and stored in RAMs. Compared to the Original Pandas implementation, Pyspark expanded and boost the programming significantly[4].

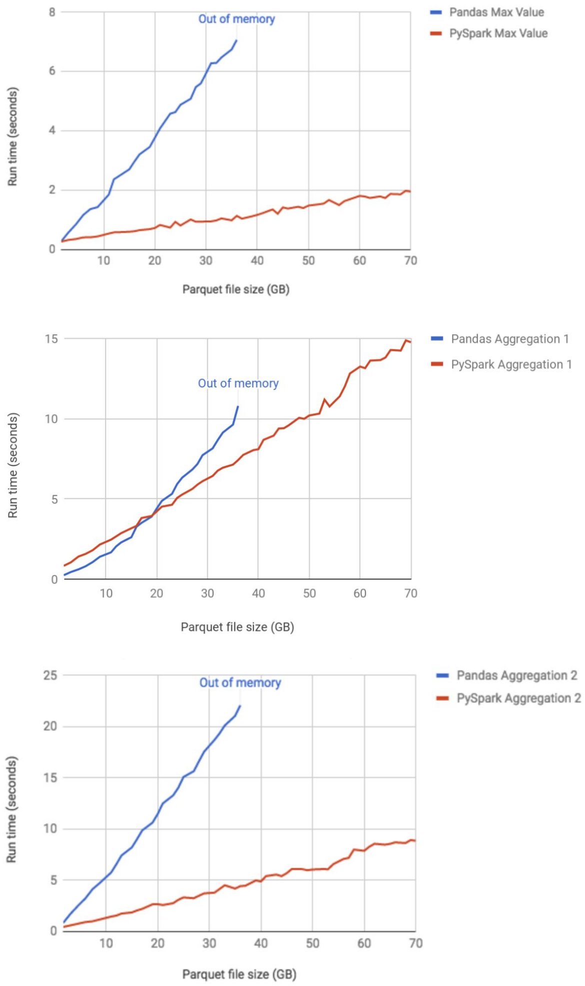
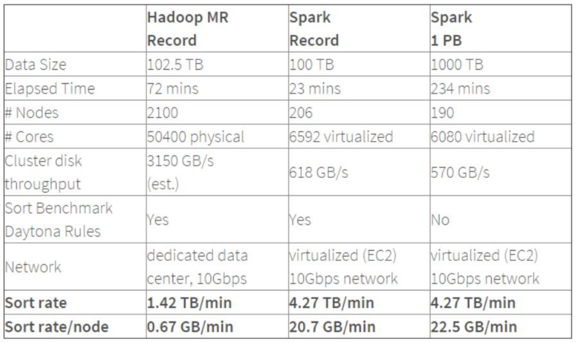


Fig. Pandas vs PySpark benchmark[4]

Among the big data parallel engine, Hadoop and Spark were two major currently used in the industry. Spark performed better than Hadoop in many aspects. Firstly, with its in-memory feature, Spark needs less disk space, therefore less expensive than Hadoop generally. Secondly, Spark uses in-memory data sharing, which contributed greatly to its high speed[5]. Since the bottleneck and speed between the disk and RAM differ significantly.



	Hadoop MR Record	**Spark Record**	**Spark 1 PB**
Data Size	102.5 TB	100 TB	1000 TB
Elapsed Time	72 mins	23 mins	234 mins
# Nodes	2100	206	190
# Cores	50400 physical	6592 virtualized	6080 virtualized
Cluster disk throughput	3150 GB/s (est.)	618 GB/s	570 GB/s
Sort Benchmark Daytona Rules	Yes	Yes	No
Network	dedicated data center, 10Gbps	virtualized (EC2) 10Gbps network	virtualized (EC2) 10Gbps network
Sort rate	**1.42 TB/min**	**4.27 TB/min**	**4.27 TB/min**
Sort rate/node	**0.67 GB/min**	**20.7 GB/min**	**22.5 GB/min**

Fig. Spark VS Hadoop benchmark[5]

Common bottlenecks for Spark appears usually in the cluster. The critical step is its shuffle stage which needs data to be stored in the disk. So disk IO is very important. Another factor is the network bandwidth since the machine needs to communicate and exchange data with each other. Finally, since Spark use RAM for the data storage. Data access becomes easy. In such scenario, CPU becomes the bottleneck.

PySpark is a very powerful tool for the data scientists to scale their project to enormous size. It transfers the existing python code to the Spark platform. In my past CS200 class, I found it very hard and slow to load or filter the data on my own computer using Pandas. There are millions of rows in the dataframe which deplete my computer. But in the real world, companies are facing Peta, Mega levels of data. PySpark saves the code by applying the distribution system that generates exponentially bigger computing power than a single machine.

3. Reference

[1] M. Zaharia, M. Chowdhury, M. Franklin, S. Shenker, I. Stoica, “Spark: Cluster Computing with Working Sets”, University of California Berkeley, CA, 2010

[2] IBM Corporation, “PySpark: High-performance data processing without learning Scala”, December 2016

[3] J. Dean, Sanjay. Ghemawat, “MapReduce: Simplified Data Processing on Large Clusters”, Google, lnc., 2004

[4] G. Wang, R. Xin, J. Damji, “Benchmarking Apache Spark on a Single Node Machine”, May 3rd 2018, URL: https://databricks.com/blog/2018/05/03/benchmarking-apache-spark-on-a-single-node-machine.html

[5] S. Nan, Z. Su, “Spark vs Hadoop MapReduce”, University of Rochester, 2017