UPPSALA
UNIVERSITET

Report for Large Datsets for Scientific Applications

Project

Seattle Library Checkout Records

Group 25

Nikhil Karthik Punnam, Vishnu Sharma, Yuvarani Masarapu

June 14, 2019

Abstract

Library is a place of main source of literary and artistic material. One could find everything knowledge-based from physical books, journals, newspapers, prints to records and tapes. With advancements in technology, it's application in a Library can greatly improve human productivity. Fast processing of purchases and searches would be possible.

Our main aim with this project is to apply our knowledge on two years' record data set obtained from Seattle Public Library [1] and work with it to make some validations to above mentioned facts about technology and library.

1 Background

Technology is growing at an incredibly high pace and it has become of an uttermost importance to keep up with it. One such place where technology is required to safeguard information and keep a track on the same is the Library.

Nowadays, we have a lot of different types of libraries accessible at different levels for people, be it the students, teachers, researchers or the general public. So, it makes sense for one to track library purchases and any other related information.

Technology can help improve the performance at the Library, making purchase tracking easier which in turn would help one get information like which purchases are the most common and how frequent are they. Such information is information is important when one needs to know how accessible a book is in a library and how frequently can it be available for purchase.

2 Data Format

The dataset is in the form of csv files distributed across the years. The dataset we handled is a record of all check outs made between April 2005 to September 2017 at the Seattle Public Library excluding renewals.

The dataset consists of the checkout records, data dictionary and library collection. The checkout record holds the raw data with only the columns that seemed most important to us and also helped in shrinking the data size. The data dictionary helps us decoding certain columns like the "ItemType" column from the checkout records. The library collection acts as a metadata about each file. This can be used to rebuild the whole dataset by merging it with the checkout records.

We aim to perform some basic analysis on the dataset and use it to measure

computational performance and scalability. Some of the questions we would like to answer with our analysis are:

- Finding the most frequently checked out item over the years.
- Finding the most frequently checked out author of sub-item books over the years.
- Finding frequency of items of a specific author each year.

3 Computational Experiments

We used Spark to analyse our dataset. [2]

3.1 Tools & Motivation

Apache provides Hadoop and spark to process large datasets efficiently. Spark can process the data in-memory while hadoop has to read from and write into the disk. Due to this, spark can be approximately 100 times faster than hadoop. However, hadoop performs better over very large datasets when compared with spark. Spark provides rdd and dataframe to perform operations over the data, which are very easy to use. These are immutable collection of records, created over data transformations such as map, filter, group etc. Sparks maintains a record of all these transformations, such that it can recreate the collection incase of any error or data loss; thereby providing fault tolerance. Since, our dataset is 7 Gbs in size, we have decided to use spark rather than hadoop for our project. Spark can perform better over this data, since its not too huge.

Our Spark cluster consisted of 1 master node and 3 worker nodes. Each of the node was made using the following configurations:

- Flavor: 'ssc.medium' of SNIC Science cloud, RAM of 4GB, VCPUs 2 VCPU and Disk of 40GB.
- Source: 'Ubuntu 18.04 LTS (Bionic Beaver) - latest' image
- Security Group:

```
ALLOW IPv4 22/tcp from 0.0.0.0/0
ALLOW IPv4 10000-65535/tcp from 76cbc145-e030
ALLOW IPv6 to ::/0
ALLOW IPv4 to 0.0.0.0/0
ALLOW IPv4 8080/tcp from 130.243.233.46/32
ALLOW IPv4 7337/tcp from 192.168.0.0/16
ALLOW IPv6 to ::/0
ALLOW IPv4 50020/tcp from 192.168.0.0/16
```

```
ALLOW IPv4 7077/tcp from 192.168.0.0/16
ALLOW IPv4 50010/tcp from 192.168.0.0/16
ALLOW IPv4 50070/tcp from 192.168.0.0/16
ALLOW IPv4 4040/tcp from 130.243.233.46/32
ALLOW IPv4 10000-65335/tcp from team25
ALLOW IPv4 50070/tcp from 130.238.0.0/16
ALLOW IPv4 4040/tcp from 192.168.0.0/32
ALLOW IPv4 7337/tcp from 192.168.1.37/32
ALLOW IP=v4 8020/tcp from 192.168.0.0/32
ALLOW IPv4 9000/tcp from 192.168.0.0/16
```

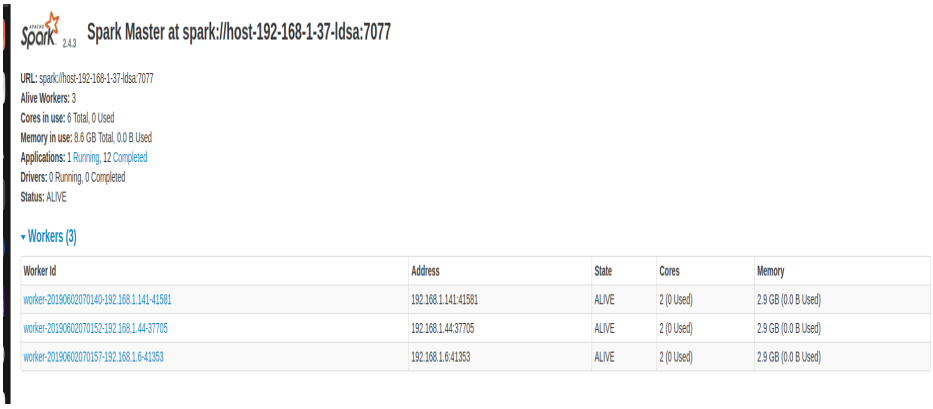

Figure 1:
fig:SparkUI

Figure **??** shows the worker nodes in the Spark UI.

3.2 Scalability

We tried changing the number of worker nodes and double the size of the data set to analyse the scalability of the cluster.

Worker/Size	7.1GB	14.2GB
1	975.54	1141.23
2	492.72	544.51
3	355.3	410.79

3.3 Plots

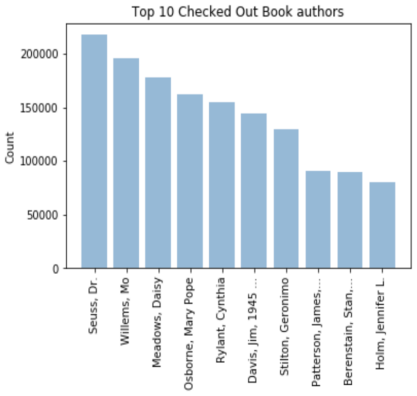

Figure 2:
Most checkout out authors of books

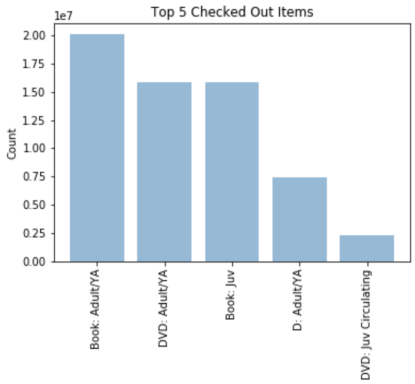

Figure 3:
Most checked out items across all items

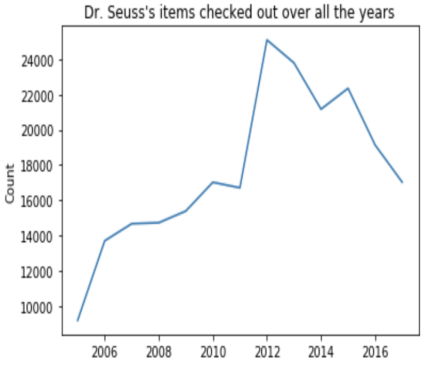

Figure 4:
Dr. Seuss's items frequency checkout per year

4 Discusion and Conclusion

We see that the time taken to process the data decrease with increase in number of nodes as expected with dealing with data computation of big data. This indicates that our cluster was successful in handling bigger data in this instance.

Our approach of using Spark via python was suitable in helping analyse and get the required information mined from the dataset. Since we were dealing with CSVs, using Pyspark's dataframes and RDD helped analyse the dataset better than traditional MapReduce. [3]

For more in-depth analysis of scalability, changing number of cores available, increasing number of CPUs, changing default memory size of RDD can help evaluate how these factors contribute in performance and scalability of distributed systems [4] .

References

[1] Kaggle. Seattle library checkout records, 2017. https://www.kaggle.com/seattle-public-library/seattle-library-checkout-records.

[2] Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

[3] Ben Blamey, Andreas Hellander, and Salman Zubair Toor. Apache spark streaming and harmonicio: A performance and architecture comparison. *CoRR*, abs/1807.07724, 2018.

[4] Taraneh Khazaei and Danny Luo. Spark performance tuning: A checklist, 2017. https://zerogravitylabs.ca/spark-performance-tuning-checklist/.