Review of The Paper "Managing the Academic Data Lifecycle: A Case Study of HPCC"

Liang Dong

1 Summary

The paper did a case study of High Performance Computing Cluster(HPCC) in academic data processing. One author of the paper is from Lexisnexis which invented HPCC. And three others are from Big Data Systems Lab in Clemson University. It is assumed that they have much experience in HPCC. In the paper, the academic data comes from NSF(National Science Foundation), NCSES(National Center for Educational Statistics), National Research Council(NRC), Integrated Postsecondary Education Data System(IPEDS) and other organizations. So the data is in very different formats and unstructured.

The paper can be divided into four parts except the introduction. In the first part, the paper gives a general introduction to HPCC platform. The second part is the most important part of the paper, the application of HPCC is discussed in three steps: ingesting, cleaning and linking data. The paper firstly introduces multiple datasets from disparate sources in this section. The data is classified into two main categories: Tabular Data and XML Data. Then the paper discusses how to clean and link unstructured data. In this section, Six Enterprise Control Language (ECL) programs are given to show dataset layout, text cleaning and data transforming function. In the third part, the paper introduces test queries and experiment platform: two physical clusters consists of reconfigurable, homogeneous set of nodes. The execution graph for one sample query is given. The performance results are listed in tables, showing that HPCC runs faster for each of the queries as the number of nodes is increased. But the speedup for two of three sample queries for a number of nodes beyond two is small. In the last part, the paper talks about the development of academic data processing. Also, the paper compares HPCC and Hadoop platform in hardware, data structure, database capability and other factors.

2 Discussion

2.1 Advantages

The paper is a good tutorial for HPCC learners because it gives detailed description of a HPCC application. Take the first part for example, it talks a lot about ECL language, parallel batch data processing (Thor) and high-performance online query applications using indexed data files (Roxie). For HPCC learners, it is good to introduce ECL types(Defintions and Actions) and formats like a textbook. Moreover, the second part discusses the whole process of HPCC application. It even explains reserved keyword EXPORT in ECL language, which is boring for experts but necessary for learners.

The paper is convincing at giving sample programs in the second part: Application of HPCC to scholarly data. The sample programs cover data ingestion, transforming, cleaning and linking. In data ingestion, ECL language is used to support CSV, XML, raw text and Excel spreadsheet. It strongly validates the strengths of HPCC to support unstructured academic data.

The paper is valuable for discussing the progress in processing academic data. In the last part, the paper introduces the trend of academic data storage selection from SQL to No-SQL. The need to employ data-intensive computing is brought out. Also,The paper discusses the similarities and differences between the Hadoop and HPCC platforms.

2.2 Disadvantages

The paper suffers from two main drawbacks although it is good tutorial. The first issue is that it lacks strong evidence that HPCC is suitable for academic data processing. Traditional scholarly data sources listed in the paper are only in the order of several GBs. Why should we use HPCC instead of personal computer? The datasets are small and can be stored in 16GB of RAM. Also, the computation is not difficult in single node.For example, the paper says that the HPCC standard library(Str library and machine learning library) makes data cleaning easier. But there are other libraries like python re library and Javascript RegExp to do the same work. Maybe the problem is not a good example of data intensive computing. The paper should give strong evidence of benefits of HPCC in solving this problem. The second issue is that the experiment seems to be bad designed. The performance on two clusters are not comparable because the memory of each node in different cluster is varying. Readers may wonder why the paper does not replicate the same queries on the same cluster.

Except the above two issues, there are some weaknesses in the writing and organization.

Confusing Graph The graphs in the paper are not clear. For example, graph 1(Thor Cluster) is from official document but miss some explanation. Graph 9 is confusing because of bad layout and unclear numbers on the line. The meaning of the numbers is not explained.

Unrelated Discusssion The paper lacks analysis of the performance of HPCC under different nodes. Performance results and discussion are not closely related. It is confusing to compare HPCC and Hadoop in the discussion part. The paper does a case study of HPCC and the experiment and performance evaluation is done in HPCC framework. The result doesn't include Hadoop performance. So the comparison is not based on experiment results and unreliable.

Improper Abbreviations Abbreviations like NCSES are confusing because the paper doesn't give the full name when they appeared first in the paper.

Bad Grammar There are some grammar mistakes in the paper. For example, the first sentence of the third part, "Two physical clusters are used test the performance performance of HPCC" should be changed to "Two physical clusters are used to test the performance of HPCC".

3 Evaluation

It is a useful investigation in application of HPCC in academic data, a good start for processing large academic datasets in the future. Firstly, it brings out an interesting problem: processing and integrating a large amount of academic data under various formats from various sources without any consistency or a predefined data structure. A comprehensive platform to support the entire academic data lifecycle is needed in the future. Secondly, it gives a solution in HPCC framework to solve the computing problem. The details of HPCC framework and ECL implementation are

included in the paper. It is a good tutorial for HPCC learners who are interested in solving similar problem in HPCC framework. It also supports basic standard for academic data processing in distributed system.

However, the paper can not have big impact in data intensive computing because the paper is not so convincing in meeting "big data" challenge. Firstly, the datasets are too small to be considered as "big data". In the experiment, query performance result on one node is hundreds of seconds. It is not good for real time query, but it is fast enough to generate a report for government or policy makers mentioned in the abstract. So the paper should better claim why we should use HPCC instead of other architectures to compute academic data. Is the academic data expanding very large in the future? Is the requirement of real time query is essential? By answering these questions, the paper can make the solution of HPCC framework more valuable. Secondly, the paper doesn't lead a new method in HPCC framework . There are similar case study papers which did better in this field . For example, Liu [**?**] proposed an approach to optimize I/O performance of small files on HDFS. Papadimitriou [**?**] proposed new framework to process extremely large datasets. Compared to those papers, the paper is not so innovative.

4 Conclusion

In summary, the paper makes a brief introduction to HPCC and does a case study of HPCC in academic data processing. It is a good tutorial for case study of HPCC. It may benefit future potential application of HPCC in the area of scholarly data research. But it is not a excellent research paper. There are three main reasons to name. There is no breakthrough methodology in the paper. The experiment and the discussion are not closely related. The grammar mistakes undermine the value of the paper. After all, it is not a classic paper.

References

[1] Liu, X., Han, J., Zhong, Y., Han, C., and He, X. Implementing WebGIS on Hadoop: A case study of improving small file I/O performance on HDFS. In Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on (pp. 1-8). IEEE.

[2] Papadimitriou, Spiros, and Jimeng Sun. Disco: Distributed co-clustering with map-reduce: A case study towards petabyte-scale end-to-end mining. Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on