Personal Big Data Computing Platform for Deep Learning: Implementation and Performance Benchmark

Chien-Heng Wu/Assistant Researcher [1], Yu-Feng Chung/Assistant Researcher [2], Wen-Yi Chang/Research Fellow [3], and Whey-Fone Tsai/Senior Research Fellow [4]
[1234] Application and Technology Division, National Center for High-Performance, Hsinchu City, Taiwan

Abstract - *The present study proposes a Personal Big Data Computing Platform which integrates Hadoop, Spark and HBase in a virtual machine environment as well as in a physical multi-node cluster to provide users for education uses or practical engineering applications. To illustrate the capability and applicability of the proposed platform, an engineering application example for the offshore wind field prediction in Taiwan is given. In this example, the heavy training work of the deep learning model is executed on the proposed platform on a daily basis. To demonstrate the computing capacity of the Personal Big Data Computing Platform, a benchmark is given by using the distributed TensorFlow approach. The results shows that based on the same small batch size, the VM platform is only 2.4 times slower than the physical cluster. Thus, the VM platform is very sufficient for educational uses. In contrast, with large batch size used, the physical cluster can perform 30 times faster than the VM platform. Hence, the physical multi-node cluster could be a good choice for personal use even for small and medium enterprise. In general, the proposed Personal Big Data Platform could be a very helpful platform for learning, development and testing in Hadoop, Spark, and HBase systems, especially for beginners.*

Keywords: Big Data Development Platform, Spark, Hadoop, MapReduce, HBase, Deep Learning Application

1 Introduction

With explosively growing data volume in many application fields (such as social media, dynamic messages, instant communication, pictures, video, huge set of images etc.), the demand for large-scale computation and storage capacities has been greatly boosted in recent years. Big Data issues are urging science and engineering users run their applications in a much more efficient way. Under such circumstances, the well-known Hadoop system [1-2] has been developed and adopted by many enterprises. Generally, building a Hadoop Cluster at least requires two physical machines for NameNode and DataNode, that enables functionalities of data partition, data replication and data distribution. Hence, for Big Data computing, a Hadoop middleware with the basis of master-slave architecture is highly recommended. On top of that, Apache Spark is also a representative open-source, distributed data processing system [3-4], which boasts the ability of in-memory computing and has been proved to be potentially 10~100 times faster than Hadoop[5]. However, building a Hadoop or Spark system within the Linux operation system may be difficult for entry-level beginners without strong IT background. Lack of Big Data platform has been a critical issue for users stepping into Big Data world.

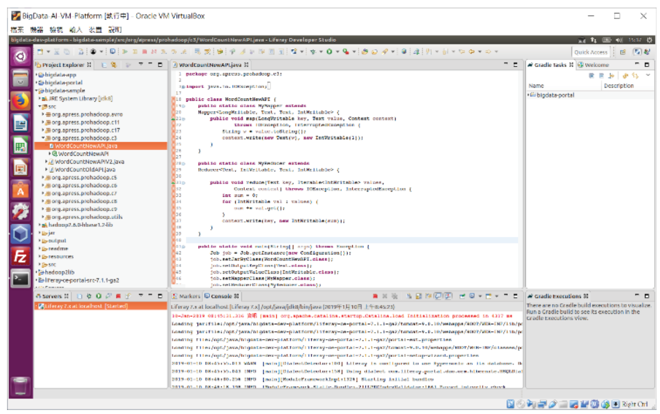

Figure 1: Big Data Development System

For the considerations described above, this study utilizes one NameNode and three DataNode to construct Personal Big Data Computing Platform, which can run Spark and Hadoop system with HBase[6] in the physical (1+3) multi-node cluster. Moreover, a duplicate Personal Big Data Computing Platform is constructed on the virtual machines for education purposes. In this platform, it contains Big Data Development System (Figure 1) for users to develop and test their Java codes. In addition, we also have overcome the problems of unstable HBase system and the superfluous data. The core components of Big Data Computing Platform including Spark, Hadoop, HBase, Zookeeper and Big Data Development System are all integrated into this platform. In addition, the Big Data Development System is bundled with Liferay Portal [7] server and sample source codes for developers to integrate the backend computing by implementing the specific portlets such as Job Submission, Job Status and other application portlets for their own purposes. For the purposes of portal development and Map/Reduce programming, this system provides developers a powerful tool for reducing application complexity, reducing development time, and improving application performance [8].

Spark is built on the top of Hadoop system. Both of them are managed by YARN for the resource management. MapReduce[9] is a computing model for data parallel-processing in high-performance cluster. Based on the divide-and-conquer method, it can significantly reduce the computing time for data-intensive applications. Developers can design and implement their Hadoop Map/Reduce programming by using Big Data Development System. The next step is to run their executable source code directly in physical Spark and Hadoop systems. This would save a lot of time because of without transferring executable JAR file and large set of data to a remote site. Therefore, the Personal Big Data Platform could be a very helpful platform for learning, development and testing in Spark, Hadoop and HBase systems, especially for beginners.

2 System Architecture

2.1 Software Architecture

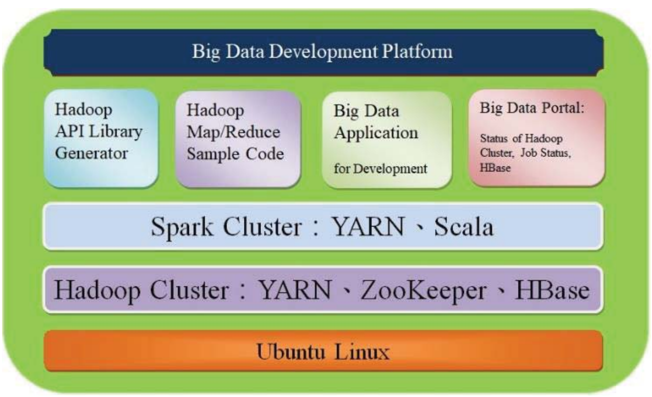

Figure 2: Architecture of Big Data Computing Platform

As shown in Figure 2, this platform integrates Spark, Hadoop and HBase systems in a (1+3) multi-node cluster, in which it includes the development tier, the middleware tier and the system tier. The software adopted in each tier is listed in Table 1.

BigData Development System	● Liferay IDE (Eclipse) 3.4.0 ● Liferay Portal 7.1.1 GA2 ● Hadoop Library 2.6.0 ● Sample Code of Hadoop 2.x Map/Reduce
Spark and Scala	● spark-1.6.0-bin-hadoop2.6.tgz ● scala-2.10.0.gz
Hadoop 2.6.0 (CDH 5.15.0)	● hadoop-2.6.0-cdh5.15.0.tar.gz ● zookeeper-3.4.5-cdh5.15.0.tar.gz ● hbase-1.2.0-cdh5.15.0.tar.gz
Java Machine	JDK 8.0 (jdk-8u162-linux-x64.tar.gz)
Operating System	Ubuntu 16.04.5 LTS

Table 1: Software of Big Data Computing Platform

In this architecture, the development tier contains four modules, which are Hadoop Map/Reduce sample codes, Big Data application development, Hadoop Library API Generator and Liferay Portal development. The middleware tier contains the multi-node Hadoop Cluster for developers to run their Map/Reduce applications during the development period. The developers can also implement Scala or Java to access Spark cluster within this tier. The system tier contains Java virtual machine within the Linux operating system, Ubuntu Linux. With this platform, users don't have to worry about the complexity of building the Big Data systems, Spark and Hadoop. Instead, they can just concentrate on the big data analytics for their specific applications.

2.2 Computing Capacity

The computing capacity of Personal Big Data Computing platform is shown in Table 2. The GPU power for deep learning computation is around 34 TFLOPS, in which it contains 10,752 GPU cores by using (GTX-1080Ti Graphic Card * 3). The in-memory computing capacity of Spark system is around 192GB. The storage capacity of Hadoop system is around 18TB. The computing capacity is suitable not only for personal use but also for university labs.

Personal Big Data Platform	(1 NameNode + 3 DataNode)
CPU	12 Cores / 24 Threads
GPU Power (Deep learning)	34 TFLOPS
Spark Cluster	192GB
Hadoop Cluster	18TB

Table 2: Computing Capacity of Personal Big Data Computing Platform

2.3 Demonstration

Figure 3: Personal Big Data Computing Platform

Personal Big Data Computing Platform contains the features such as small size, low-heating, ease of movement.

For university labs or small business, it is also a better choice for data security and time-saving without sending their high-value data and source code to other remote sites.

2.4 Deep Learning Application

2.4.1 Distributed TensorFlow

This case study utilizes distributed TensorFlow to test the computational performance in multi-node VM platform and multi-node physical cluster. TensorFlow[10] is an open source software library for machine learning and deep learning. It provides API for users to create a cluster of TensorFlow servers implementing deep learning distributed computation in multi-node cluster. As shown in Figure 4, a TensorFlow cluster comprises two components, which are the “parameter server” and “worker”. The parameter server hosts nodes to store and update variables, while the worker hosts nodes to perform compute-intensive training tasks. The tasks of a job are typically run on different machines. Hence, a TensorFlow server is created to communicate with any other server in the cluster to perform distributed computation. When the computation process starts, the workers train a batch of dataset and return the corrected weight of model to the parameter server. Then the parameter server updates and returns the weight of model to the workers. The architecture of distributed computation is show as Figure 4.

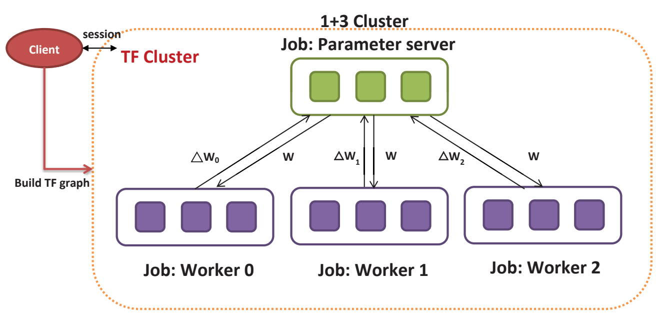

Figure 4: Architecture of distributed TensorFlow cluster

2.4.2 Application example: RNN-LSTM model for offshore wind field prediction

- This study adopted the deep learning RNN-LSTM model for offshore wind field prediction developed by our research team[11] as the application example and evaluated the computing performance of Personal Big Data Computing Platform in this case.

- The regional wind field data with a sampling period of 10 minutes, length of six months, and variables containing wind speed, wind direction, pressure, relative humidity, temperature are used as training dataset for RNN-LSTM model. Then, the wind field data are further organized as the time sequence dataset with the shape of (56953, 36, 225). It means there are 56953 samples, and each sample has 36 time steps and 225 features in it.

The RNN-LSTM model for offshore wind field prediction is deployed through the distributed TensorFlow approach to test different computing environments including multi-node virtual machine(VM) platform and multi-node physical cluster system. Different batch sizes are used according to its system capacity, in which the virtual machine can afford 64 while the physical cluster system can afford more than 1024. The Mean Square Error is used as the loss function of RNN-LSTM model. In these tests, the computing time is recorded when the loss value converges less than the threshold. The screenshot of the distributed Tensorflow computation in the multi-node physical cluster is show as Figure 5.

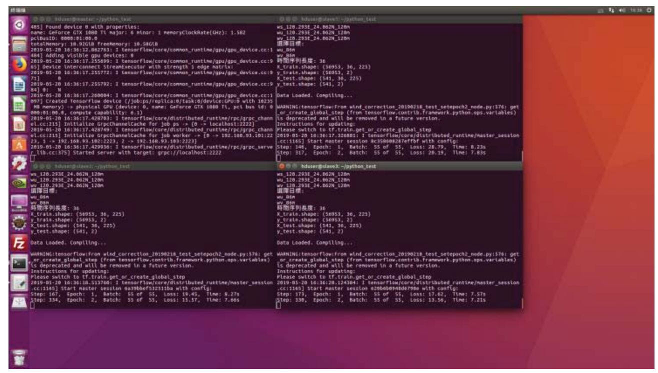

Figure 5: Distributed Tensorflow computation in the multi-node physical cluster

As show in Table 3. In Case 1, the model is trained in a 2-node virtual machine platform. In Case 2, the model is trained in a 4-node virtual machine platform. In Case 3, the model is trained in a 2-node physical cluster system. In Case 4, the model is trained in a 4-node physical cluster system. In Case 5, the model is trained in a 2-node physical cluster system. In Case 6, the model is trained in a 4-node physical cluster system.

Case	Platform	Nodes	Batch Size	Time
1	2-node virtual machine	1 name node+ 1 data node	64	15hrs 33mins 3.61sec
2	4-node virtual machine	1 name node+ 3 data node	64	7hrs 30mins 0.44sec
3	2-node physical cluster	1 name node+ 1 data node	64	6hrs 14mins 0.09sec
4	4-node physical cluster	1 name node+ 3 data node	64	2hrs 39mins 21.91sec
5	2-node physical cluster	1 name node+ 1 data node	1024	34mins 59.16sec
6	4-node physical cluster	1 name node+ 3 data node	1024	12mins 44.16sec

Table 3: Conditions and results of computation cases

The computational results are also shown in Table 3. By comparing Case 1 ~ Case 4, the computational time by the 2-node physical cluster is 2.5 times faster than the 2-node virtual machine, and the 4-node physical cluster is 2.4 times faster than the 2-node physical small-cluster. It means that the computing performance gets better as the node number of the physical cluster increases. By comparing Case 4 with Case 6, when the batch size of model increases from 64 to 1024, the computational time is reduced by 12.5 times in the 4-node physical cluster cases.

2.5 Concluding Remark

The present study is focused on the development of Personal Big Data Computing Platform, and the benchmark performance of distributed TensorFlow is proposed. The conclusion and suggestion are provided below:

A. This study utilizes one NameNode and three DataNode to construct Personal Big Data Computing Platform, which integrates Big Data Development System with Spark, Hadoop and HBase in the physical (1+3) multi-node cluster, in which it includes the development tier, the middleware tier and the system tier. Users do not have to worry about the complexity of building the Big Data systems. Instead, they can just concentrate on the big data analytics for their specific applications.

B. This case study utilizes distributed TensorFlow to test the performance of distributed computation in multi-node VM platform and multi-node physical cluster. Based on the same small batch size, the VM platform is only 2.4 times slower than the physical cluster. Thus, the VM platform is very sufficient for educational uses. In contrast, with large batch size used, the physical cluster can perform 30 times faster than the VM platform.

C. This study adopted the deep learning RNN-LSTM model for offshore wind field prediction as an application example to deomonstrate the capability and applicability of Personal Big Data Computing Platform. We believe more applications can be integrated into the proposed platform in the future.

2.6 Future Work

Due to the use of most Hadoop systems is still in the command-line interface, it may be inconvenient for beginners to use. To improve the working efficiency for users in Spark and Hadoop management and applications, the user friendly interface such as a Big Data portal would be the better way for future work. The Big Data portal will be developed by using Liferay Portal, which will provide friendly interface for users to access Spark and Hadoop cluster in Personal Big Data Computing Platform. In addition, Spark possesses the ability of in-memory computing, and is shown to be potentially 10~100 times faster than Hadoop. The future work of research would be focused on Spark applications in Personal Big Data Computing Platform. This would allow users for fast data processing of their big data development and applications.

2.7 Acknowledgement

Financial support from the Ministry of Science and Technology, Taiwan, under grants MOST 107-2218-E-492-004 is highly appreciated. We are also grateful to the National Center for High-Performance Computing for computer time and facilities.

3 References

[1] Dean, J and Ghemawat, S., “MapReduce: Simplified Data Processing on Large Clusters” 6th Symposium on Operating Systems Design & Implementation (OSDI’ 04), December 6-8, San Franscisco, California, USA. (2004)

[2] Apache Hadoop, http://hadoop.apache.org/

[3] Apache Spark, http://spark.apache.org/

[4] M. Zaharia, M. Chowdhury, T. Das, A. Dave, J. Ma, M. McCauley,M. J. Franklin, S. Shenker, and I. Stoica. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In Pro-ceedings of the 9th USENIX Conference on Networked Systems Design and Implementation, NSDI’12, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.

[5] Yuan Yuan, Meisam Fathi Salmi, Yin Huai, Kaibo Wang, Rubao Lee, Xiaodong Zhang. Spark-GPU: An Accelerated In-Memory Data Processing Engine on Clusters. In Proceedings of the 2016 IEEE International Conference on Big Data (IEEE BigData 2016) , December 5~8, Washington D.C., USA., pp.273-283, 2016 (EI).

[6] Apache HBase, https://hbase.apache.org/

[7] Liferay Portal，https://www.liferay.com

[8] Chien-Heng Wu, Wen-Yi Chang, Whey-Fone Tsai, Franco Lin, Ching-Fang Lee, Chao-Tung Yang, “MULTI-NODE BIG DATA VM PLATFORM AND JOB SUBMISSION PORTLET” 2nd International Conference on Big Data, Cloud Computing, and Data Science (BCD 2017) , July 9~13, Hamamatsu, Japan, pp. 234–239, 2017 (EI)

[9] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. In Proc. of OSDI, 2004.

[10] Tensorflow, https://www.tensorflow.org

[11] Yu-Feng Chung, Chih-Yu Kuo, Wen-Yi Chang, Whey-Fone Tsai, “FORECAST OF WIND FIELD IN TAIWAN OFFSHORE WIND FARM BY MODEL SIMULATION AND DEEP LEARNING” 3rd International Conference on Offshore Renewable Energy (CORE 2018), Aug. 29-30 2018, Glasgow, UK.