บันทึกข้อความ

ส่วนราชการ ภาควิชาวิศวกรรมคอมพิวเตอร์ศิริราชพยาบาล คณะวิศวกรรมศาสตร์ ทิศ 3330
ที่ พบ 0523.8.6/ 01101 วันที่ 30 พฤศจิกายน 2559
เรื่อง ขออนุมัติพิจารณาสนับสนุนการนำเสนอผลงานวิจัย/ผลงานทางวิชาการ แบบ Oral Presentation
ณ ที่จัดประกวด ปีการศึกษา 2560 ที่ 21
เรียน รองศาสตราจารย์ และอาจารย์ผู้ช่วย

ตัวอย่างเรื่อง นางสาวธนิตา พลวงษ์ชัย ผู้ช่วยเจ้าหน้าที่ ทรงคุณวุฒิ ภาควิชาวิศวกรรมคอมพิวเตอร์ศิริราชพยาบาล นำเสนอผลงานวิจัยเรื่อง "A comparison of apache hadoop distributions using hibench" ในการประชุมวิชาการระดับนานาชาติ "ARTIFICIAL LIFE AND ROBOTICS (AROB 22nd 2017)
ระหว่างวันที่ 19 – 21 มกราคม 2560 ณ ประเทศญี่ปุ่น
เพื่อส่งเสริมการพัฒนาความรู้ ประสบการณ์ทางวิชาการและมีความรู้ที่ดีขึ้นในทางการศึกษาและการวิจัยด้านวิทยาการคอมพิวเตอร์ต่อไป

ดังนั้น จึงขอร้องผู้อุทิศประจำสัญญาสนับสนุนการนำเสนอผลงานดังกล่าวจำนวนเงินรวมทั้งสิ้น 12,520 บาท (หนึ่งแสนสองพันห้าร้อย-twenty thousand) เพื่อใช้ในการประชุมวิชาการระดับนานาชาติ 1 ชุด
ตามตัวอย่างดังนี้ 1. จัดทำใบตอบรับการเข้าร่วมประชุม ARTIFICIAL LIFE AND ROBOTICS (AROB 22nd 2017)
2. ระยะชั่วโมงในการเข้าร่วมประชุม 3. บทความที่จะนำไปเสนอต่อที่ประชุม

ขอเรียนมาเพื่อโปรดพิจารณาอนุมัติ

(นางสาวธนิตา พลวงษ์ชัย)
อาจารย์ผู้ช่วย ภาควิชาวิศวกรรมคอมพิวเตอร์ศิริราชพยาบาล

เรียน รองศาสตราจารย์ และอาจารย์ผู้ช่วยเจ้าหน้าที่

(ผู้ช่วยศาสตราจารย์ ดร.วัชรินทร์ วงษ์สุทธิสิน)

หัวหน้าภาควิชาวิศวกรรมคอมพิวเตอร์ศิริราชพยาบาล

(ลงชื่อ)

ชัยอร ปษิศกุล

(ลงลายมือชื่อ)

(ลงชื่อ)

(ลงลายมือชื่อ)

ปวันที่ 14 ธันวาคม 2559
คำวิจารณ์ในการตีพิมพ์เผยแพร่ผลงานวิจัย
เรื่อง “A comparison of apache hadoop distributions using hibench”
ในการประชุมวิชาการระดับนานาชาติ “ARTIFICIAL LIFE AND ROBOTICS (AROB 22nd 2017)”
ระหว่างวันที่ 19 – 21 มกราคม 2560 ณ ประเทศญี่ปุ่น

<table>
<thead>
<tr>
<th>ลำดับที่</th>
<th>รายการ</th>
<th>มหาวิทยาลัย</th>
<th>ค่าสวัสดิการระบบสารสนเทศ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ค่าเครื่องเปิดลิสต์สารสาหร่าย อุบลฯ – กรุงเทพฯ – อุบลฯ</td>
<td>2,600</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>ค่าเครื่องเปิดลิสต์สารสาหร่าย กรุงเทพฯ – ชลบุรี – กรุงเทพฯ</td>
<td>12,120</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>ค่าที่พัก 4 คืน x 6,000 บาท/คืน</td>
<td>24,000</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ค่าเบี้ยประกันสุทธิ 4 วัน x 2,100 บาท</td>
<td>8,400</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>ค่าธรรมเนียมจัดพิมพ์และจ่ายประทาน</td>
<td>1,280</td>
<td>4,120</td>
</tr>
<tr>
<td></td>
<td>รวมทั้งสิ้น</td>
<td>40,000</td>
<td>12,520</td>
</tr>
</tbody>
</table>
LETTER OF ACCEPTANCE

Dear Authors:

Paper Title:
A comparison of apache hadoop distributions using hibench

Author(s):
Aranya Pothisom Florence, Thanisa Nunmorda

Presentation method: Oral Presentation

As general chair of AROB, I am pleased to inform you that your paper has been accepted for presentation at the symposium. If you have any questions, please don't hesitate to contact us.

We are looking forward to meeting you at the conference.

Sincerely yours,

Hiroshi Teraoka
General Chair of AROB,
Tokyo Medical and Dental University

AROB Secretariat
International Society of Artificial Life and Robotics
A-101, 8-7 Hatakonaka, Oka, 870-0856 Japan,
Tel: +81-97-594-0131
Fax: +81-97-547-9242
E-mail: arobsoc@isarob.org URL: http://isarob.org/symposium/
Accepted Papers will be published in the Proceedings of the AROB. Extended version of the selected papers will be published in the international journal: *artificial life and robotics*, Springer Japan.
Registration No.: 2230043

Participant: Prof. Thanisa Numnonda
Affiliation: King Mongkut's Institute of Technology Ladkrabang, Thailand
Email Address: thanisa@it.kmitl.ac.th

Academic Area:
Others: Big Data

Submission Number(s): 2210089 and 2210146
Paper Title(s): Real-time recommendation engine using lambda architecture and A comparison of apache hadoop distributions using hibench

Registration Fee
Symposium Fee (Regular): 35,000 JPY
Additional Paper: 30,000 JPY * 1

Total Amount: 65,000 JPY

Method of Payment:
Credit Card
Call for Papers

The Twenty-Second International Symposium on Artificial Life and Robotics will be held in Beppu, Oita, Japan, January 19-21, 2017. This symposium will bring together researchers to discuss development of new technologies concerning artificial life and robotics based on computer simulations and hardware designs of state-of-the-art technologies, and to share findings on how advancements in artificial life and robotics technologies that relate to artificial intelligence, virtual reality, and computer science are creating the basis for exciting new research and applications in various fields listed in the following topics:

Topics of interest include, but are not limited to:

- Agent-based modeling
- Artificial brain
- Artificial intelligence
- Artificial life
- Artificial living
- Artificial mind
- Bio-robot
- Bio-complexity
- Bio-informatics
- Biological evolution
- Biomedical imaging
- Brain science
- Chaos
- Cognitive science
- Complexity
- Control techniques
- Data mining
- DNA computing
- Evolutionary computations (Genetic algorithm)
- Genome/Genics medicine
- Human-machine interaction and collaboration
- Identification and estimation
- Intelligent control
- Learning
- Manipulator
- Medical informatics
- Mobile robots
- Molecular biology
- Molecular network
- Motion planning and navigation
- Multi-agent systems
- Nano-biology
- Nano-robot
- Neural networks
- Neurocomputing technologies and its application for hardware
- Quantum computing
- Regenerative medicine
- Robot vision and image processing
- Robotic Mechanism
- Sensor and multi-sensor data fusion
- Swarm Intelligence
- Swarm robot
- Tele-operation
- Walking robot

Important Dates:

- October 12, 2016 -> October 31, 2016: Deadline for Abstract Submissions
- October 5, 2016 -> October 24, 2016: Deadline for OS Proposal
- October 26, 2016 -> November 7, 2016: Notification of Acceptance
- December 1, 2016: Final Camera-Ready Papers Due
- December 1, 2016: Deadline for Early Registration

Publication:

Accepted papers will be published in the Proceedings of the AROB. Extended versions of the selected papers will be published in the international journal: ARTIFICIAL LIFE AND ROBOTICS, Springer Japan.
Beppu Onsen Guide

Beppu is one of the oldest and most famous hot spring areas in Japan. The city is divided into eight major hot spring areas, known as Onsen (温泉).

- Beppu Onsen:
  Close to Beppu Station, the center of Beppu, Beppu Onsen is the most popular area in Beppu.

- Koma Kato Onsen:
  10-minute walk from Beppu Station.

- Takeawara Onsen:
  Originally built in 1879, Takeawara Onsen is a short seven-minute walk from Beppu Station. A bathhouse with an elaborate facade is a symbol of Beppu Onsen.

- Myochin Onsen:
  Located on the outskirts of Beppu. Visitors enjoy several types of hot springs, such as mud, sulfur, and hot air. It is accompanied by a picturesque riverside that is popular for both natural and cultural attractions.
Timetable: Please note that you have 10 minutes for presentation and 5 minutes for discussion.
<table>
<thead>
<tr>
<th>Time</th>
<th>Room A</th>
<th>Room B</th>
<th>Room C</th>
<th>Room D</th>
<th>Room E</th>
</tr>
</thead>
<tbody>
<tr>
<td>9:00-10:00</td>
<td>C-51</td>
<td>C-60</td>
<td>C-70</td>
<td>C-80</td>
<td>C-90</td>
</tr>
<tr>
<td>10:00-10:30</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
</tr>
<tr>
<td>10:30-11:00</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
</tr>
<tr>
<td>11:00-11:30</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
</tr>
<tr>
<td>11:30-12:00</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
<td>Coffee Break</td>
</tr>
<tr>
<td>12:00-14:00</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
</tr>
<tr>
<td>14:00-15:30</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
</tr>
<tr>
<td>15:30-16:00</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
<td>Lunch / Poster Session</td>
</tr>
</tbody>
</table>

**Saturday, January 20, 2017**

**10:00-10:30**
Text:** Coffee Break

**10:30-11:00**
**Session A**
**Title:** Brain-Machine Interfaces with Functional Neurological Symmetry
**Speakers:** Kentaro Haraguchi, M.D., Ph.D., Osaka University, Japan

**11:00-11:30**
**Lunch / Poster Session**

**11:30-12:00**
**Session A (Continued)**
**Title:** Neurorehabilitation
**Speakers:** K. Hishida, K. Watanabe

**12:00-14:00**
**Lunch / Poster Session**

**14:00-15:30**
**Session A (Continued)**
**Title:** Artificial Life and Robotics
**Speakers:** K. Watanabe, K. Hishida

**15:30-16:00**
**Session A (Continued)**
**Title:** Artificial Intelligences
**Speakers:** K. Watanabe, K. Hishida

**16:00-17:00**
**Session A (Continued)**
**Title:** Neurorehabilitation
**Speakers:** K. Hishida, K. Watanabe

---

**http://www.organicconference.org/index.php?main_page=timetable**
Room E

G99 Data mining II

Chair:

G99-1 Selecting important features for detecting unreliable pages on Facebook
   Poulida Sengura, Chidawave Jirempon

G99-2 A comparison of Apache Hadoop distributions using H2O benchmark
   Araya Paninakorn, Florence, Martha Mwanasuya

G99-3 Proposal of network visualization of customer expectation by using Web
   Promoter Sage

G99-4 Identifying symptoms of elderly people by using Sequential Rules
   Sulitja Woro beloved, Poulida Sengura

G99-5 The analysis of association of category of interesting pages on Facebook using FP-Growth Algorithm
   Recknam pm Chongjana, Poulida Sengura

G99-6 Mining Bestseller Products from Product Models Repositories [Withdrawal]
   Jutrina Dejpiriy
A comparison of apache Hadoop distributions using HiBench

Araya Pattharamon Florencia1 and Thanisa Numnumenda2
1UBC Ratchaburi University, Thailand
2Faculty of Information Technology, King Mongkut’s Institute of Technology Ladkrabang, Thailand
(Tel: 66-45-353-329, Fax: 66-45-353-300)
1araya.f@ubu.ac.th, 2thanisa@kt.kmitl.ac.th

Abstract: Nowadays, Big Data technology plays a vital role as one of the emerging technologies that can be used in various applications. Every second, data generated from billions of devices are sent up into the cloud to be analyzed and used for prediction and decision making in various applications. The Big Data platform can be seen as a foundation of Big Data implementation. This article reports a comparison of three different platforms: Apache Hadoop, Cloudera (Express), and Hortonworks in the respects of installation, cluster management, cost, and stability using HiBench as a testing benchmark. In all three distributions, Spark was highly shown great potential to be chosen as a processing platform for Big Data and Cloudera (Express) was chosen to be the easiest in the respects of installation and cluster management.

Keywords: Apache Hadoop, Big Data, HiBench, Spark

I INTRODUCTION

In the digital era, data and information are generated, collected, and analysed intensely. The amount of data created in a day for each and every device is enormous. These data are produced in high velocity and they can be structured or unstructured coming from various sources. The three Vs defined volume, velocity, and variety referring to characteristics of Big Data. Analysing this data can be useful for any organizations to better target clients. To be able to analyse this kind of data, Big Data platform should be implemented as an infrastructure to serve Big Data requirements. The Big Data platform is used for gathering, preparing, and analysing this kind of data for further analysis, including using machine learning for both prediction and decision making. Choosing the platform is selecting a foundation for the system and define the ecosystem of Big Data analysis, including tools for data analysis such as MapReduce, Hive, Mahout, Impala, or Spark.

In this research, three distributions were set up in similar requirements using Amazon Web Services cloud platforms. Each cluster consisted of 5 nodes, which are one manager, one namenode, and 3 datanodes. The namenode was m3.2xlarge while the other nodes of the cluster were m3.large EC2s. The benchmark test was executed using HiBench [1], a realistic and comprehensive benchmark suite for Hadoop originally developed by IBM. HiBench offers three sets of Hadoop programs, including Micro-Benchmarks, Web Search, and Machine Learning. Three categories of test conducted in this research were MapReduce, Hive, and Spark. For Micro-Benchmarks groups, this research used HadoopJoin (Hive), JavaSparkJoin, ScalaSparkJoin, and PythonSparkJoin programs. Then, for Web Search groups, HadooPageRank (MapReduce), JavaSparkPageRank, ScalaSparkPageRank, and PythonSparkPageRank were used. Finally, for Machine Learning groups, HadoopKMeans (Mahout), JavaSparkKMeans, ScalaSparkKMeans, and PythonSparkKMeans were used. All of these programs were used for evaluating the system performance in terms of processing speed, throughput, resource utilization, and data size.

2 RELATED THEORIES

Hadoop [2] is one of Apache’s open source projects for store and manage Big Data. Hadoop is written in Java and has the ability to support fault tolerance for storing data in more than one place. Hadoop system is a core foundation for many commodity servers in horizontal scale nodes. Hadoop project was originally started by Doug Cutting and Mike Cafarella [3] when they were working at Yahoo. Later on, many companies started to use Hadoop including eBay, Facebook, and Amazon. At the time of conducting this research, Hadoop 2.5 was used, however, the latest version of Hadoop at this stage is 3.x alpha which was recently released in September this year (2016). The current stable version is 2.7.2 [4]. Normally Hadoop technology is run on a group of servers consisted of at least one Master node and many Worker nodes which will be used for processing and storing data. These worker nodes can be used as Data Node or node management (Node Manager). There are a few popular Hadoop distributions in the market such as Cloudera, MapR, and Hortonworks. This article reports a comparison of three different Big Data platforms: Apache Hadoop, Cloudera (Express), and Hortonworks in the respects of installation, cluster management, cost, and stability.

In any distribution, Hadoop ecosystem mainly comprises of HDFS as a file system, MapReduce as a software framework and YARN as a resource manager.
These are not enough for further data analysis which required SQL queries, random access read/write data and data management including transferring data to and from RDBMS or retrieving streaming data. Additional tools such as Hive, Spark, Pig, and Impala. In this research, we focused on HDFS, YARN, Hive, and Spark.

HDFS [3], Hadoop Distributed File System is the most popular technology for unstructured data storage because Hadoop stores data in commodity server's storage in distributed fashion. Being distributed means that its data in small blocks are duplicated at least 3 copies and are stored in different nodes which is important for fault tolerance.

MapReduce [6] is the main processing framework for older Hadoop version which processes data in batch mode. Java programs need to be developed to use MapReduce for processing data in HDFS. In a newer version of Hadoop, Spark was introduced and predicted to replace MapReduce as data processing nowadays can be interactive, real-time processing and even machine learning, not only batch processing anymore.

YARN, Yet Another Resource Negotiator is a resource manager on Master node in Hadoop system, the main duty is distributing jobs to worker nodes via Node Manager. The processing mode can be MapReduce in batch, Tez in interactive mode or Spark in real-time.

Hive is a simple SQL like language tool for query data in HDFS to bypass Java program development for using MapReduce. However, Hive actually translates SQL like queries to be MapReduce which is then processed in batch mode.

Makout is a predictive analytic tool for data scientist using Java offering various predictive algorithms such as recommender, classification, and clustering.

Spark [7] is a Big Data real-time data processing technology using in-Memory processing mode. Spark is the most prominent framework in Big Data processing tools as its processing time is a lot faster than MapReduce. Spark offers importing data from various sources including HDFS, Cloud storage, and NoSQL in different languages such as Java, Scala, R, and Python.

Pig is very similar to Hive, Pig Latin script can be used to query data on HDFS without Java program to use MapReduce. However, Pig is suitable for ETL for dealing with JSON.

Impala is an SQL like language which similar to Hive but work faster than Hive as Impala works in in-memory mode using its Statistics at Master node then distributing jobs to Worker nodes through Impala (executor Impala). Impala itself is developed by Cloudera and only available on Cloudera distribution.

One important point of working with Big Data is the ability of processing data from other sources that is not HDFS. Processing data on HDFS can be classified as four types those are interactive analysis, batch analysis, real-time analysis and machine learning. When choosing Big Data platform, the main concern should be both using Hadoop with data on HDFS and processing data from other sources using Spark.

2.1 Processing data with Hadoop

When analyzing data on Hadoop, MapReduce is used and completed in batch processing fashion which required Java programming or some other languages. Recently, the popularity of MapReduce has been declined as being replaced by Spark. However, some Hadoop distributions offer analytic tools with SQL like language/script, these tools are Hive, Pig, and Impala.

2.2 Processing data with Spark

Spark can process data on HDFS ten times faster than MapReduce. It can be used as a tool to import data from other type of sources such as cloud storage, NoSQL, RDBMS. Spark has ability to be operated as distributed or distributed mode over Hadoop clusters on YARN as interactive mode. Four main components of Spark are Spark Core, Spark Streaming, Spark SQL, and MLlib. Spark Core provides processing platform API for various languages including Java, Scala, Python, and R. Spark Streaming offered real-time processing which is suitable for importing real-time feeds of data. Spark SQL is an SQL like processing platform and MLlib is a machine learning processing tools.

3 METHODOLOGY

Three main computation aspects in research methodology being discussed here are cluster installation, benchmark test using HiBench and cluster management.

3.1 Cluster Installation

All three distributions being investigated in this research were installed in AWS cloud service using four EC2s [8], each distribution consisted of one as a master node with other three as worker nodes. Each EC2 was Ubuntu server 14.04 1.78, the master was m3.large (3 vCPU, 30 GB RAM, 160 GB SSD, 500 GB Storage) and the worker nodes were m3.large (4 vCPU, 15 GB RAM, 80 GB SSD, 500 GB Storage).

3.2 HiBench Tests

A simple test of importing data into HDFS and export out was tested using both command line and over HIUI. Data processing was tested on MapReduce, Hive, Pig, and Spark. HiBench was then used as a benchmark tool, HiBench consists of three groups, those are:

- Micro Benchmark, SQL query was tested using Join for Hive and Spark
- Web Search, PageRank algorithm was tested on both MapReduce and Spark
- Machine Learning, K-Nearest algorithm was tested on Mahout and Spark

3.3 Cluster Management

In term of cluster management, we tested how difficult for each distribution to add or delete a node to and from a cluster. Another EC2 was created and added into the cluster then one node was removed from the cluster. After adding a new node to the cluster, configurations for services on the new node were observed. Before removing any node,
services have to be stopped completely for that node to be able to be removed. Adding and removing nodes can be challenging in the real situation when a node needs a replacement.

4 RESULTS

According to the comparison aspects discussed in methodology section, results can be divided into three parts below.

4.1 Cluster Installation

Hadoop distribution was the most difficult as in all three distributions tested, it required a knowledgeable staff to conduct a cluster installation using pure command line. The hard work of installing each node was node and services configuration. The installer needed to know all related services available for the version of installation. While the installation and configuration on Cloudera (Express) and Hortonworks were quite easy with a well designed graphic user interface (GUI), Cloudera or CDH offered a set of services called parcels to the installer can choose from Cloudera Manager to install any related services using parcels without having to know which chaining required services. Cloudera Manager interface is shown in Fig.1. However, Hortonworks installation was different from the other two, the installation was conducted using CloudBreak and at least 9 nodes using EC2s on AWS. Hortonworks graphic user interface manager called Ambari is shown in Fig.2.

4.2 HiBench Testing Results

According to the installation of Hortonworks that required 9 nodes in total and some difficulty of YARN configuration, therefore, the results from Hortonworks distribution cannot be used in comparison here. The results of Hadoop 2.6 is shown in Table 1 and CDH in Table 2.

<table>
<thead>
<tr>
<th>Application</th>
<th>Input Data Size (Bytes)</th>
<th>Time (seconds)</th>
<th>Throughput (bytes/second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HadoopParquet (Eval)</td>
<td>193,704,741</td>
<td>163.242</td>
<td>9,690</td>
</tr>
<tr>
<td>JavaSparkJoin</td>
<td>193,704,741</td>
<td>10.734</td>
<td>17,315,006</td>
</tr>
<tr>
<td>ScalaSparkJoin</td>
<td>193,704,741</td>
<td>10.706</td>
<td>18,031,492</td>
</tr>
<tr>
<td>PySparkPyGraf</td>
<td>193,704,741</td>
<td>12.735</td>
<td>15,115,004</td>
</tr>
<tr>
<td>HadoopPageRank (MapReduce)</td>
<td>433,459</td>
<td>433,459</td>
<td>896,008</td>
</tr>
<tr>
<td>JavaSparkPageRank</td>
<td>259,928,113</td>
<td>1.714</td>
<td>151,473,072</td>
</tr>
<tr>
<td>ScalaSparkPageRank</td>
<td>259,928,113</td>
<td>1.735</td>
<td>149,458,275</td>
</tr>
<tr>
<td>PySparkPageRank</td>
<td>259,928,113</td>
<td>3.9</td>
<td>66,448,234</td>
</tr>
<tr>
<td>HadoopKmeans (MapReduce)</td>
<td>4,016,371,688</td>
<td>714,959</td>
<td>5,517,310</td>
</tr>
<tr>
<td>JavaSparkKmeans</td>
<td>4,016,371,688</td>
<td>1.756</td>
<td>3,457,227,612</td>
</tr>
<tr>
<td>ScalaSparkKmeans</td>
<td>4,016,371,688</td>
<td>1.687</td>
<td>2,565,728,195</td>
</tr>
<tr>
<td>PythonSparkKmeans</td>
<td>4,016,371,688</td>
<td>3.727</td>
<td>1,077,543,217</td>
</tr>
</tbody>
</table>

Fig. 1. Services on Cloudera Manager

Fig. 2. Services on Ambari
### Table 2: Data processing on CDH

<table>
<thead>
<tr>
<th>Application</th>
<th>Input Data Size (bytes)</th>
<th>Time (seconds)</th>
<th>Throughput (bytes/second)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hadoop (Hive)</td>
<td>193,204,734</td>
<td>67.16</td>
<td>2,905</td>
</tr>
<tr>
<td>JavaSpark</td>
<td>60,044</td>
<td>44.044</td>
<td>4,287,136</td>
</tr>
<tr>
<td>ScalaSpark</td>
<td>45,701</td>
<td>4,245,842</td>
<td></td>
</tr>
<tr>
<td>PythonSpark</td>
<td>44,357</td>
<td>4,351,429</td>
<td></td>
</tr>
<tr>
<td>HadoopPageRank (mapreduce)</td>
<td>264.409</td>
<td>943,810</td>
<td></td>
</tr>
<tr>
<td>JavaSpark PageRank</td>
<td>79.573</td>
<td>2,607,185</td>
<td></td>
</tr>
<tr>
<td>ScalaSpark PageRank</td>
<td>102.21</td>
<td>2,542,581</td>
<td></td>
</tr>
<tr>
<td>PythonSpark PageRank</td>
<td>251.405</td>
<td>1,835,901</td>
<td></td>
</tr>
<tr>
<td>HadoopKinesis (Mahout)</td>
<td>421.077</td>
<td>9,076,377</td>
<td></td>
</tr>
<tr>
<td>JavaSpark Kmeans</td>
<td>227.927</td>
<td>17,621,167</td>
<td></td>
</tr>
<tr>
<td>BodeSpark Kmeans</td>
<td>47.986</td>
<td>50,655,452</td>
<td></td>
</tr>
<tr>
<td>PythonSpark Kmeans</td>
<td>209.842</td>
<td>133,909,594</td>
<td></td>
</tr>
</tbody>
</table>

A comparison of processing time between Hadoop 2.6 and CDH is shown in Fig. 3. Throughput comparison is shown in Fig. 4. Processing time of each testing program using CDH, when processed by Spark, Hadoop is faster. However, both CDH and Hadoop, Spark confirmed a better speed over traditional tools offering by Hadoop technology. Similarly shown in throughput, Hadoop offered a better throughput in each run result.

### 4.3 Cluster Management

Adding node to an existing cluster for Hadoop was difficult to stop services and the cluster including preconfigure the new node before restarting the cluster. In CDH and Hortonworks were easier done over GUI using Cloudera Manager for CDH and Ambari on Hortonworks. New nodes can be RC2 and pre-installed operating system before adding to the cluster. Fig. 5 shows a node added into existing cluster can be completed using node template on Cloudera Manager. Deleting nodes from the cluster should be conducted carefully under circumstances that the related services already stopped. CDH and Hortonworks are suggested for less-skilled administrators while Hadoop required a well-versed knowledgeable staff.

---

**Fig. 3.** Processing time using Hadoop versus CDH

**Fig. 4.** Processing Throughput using Hadoop versus CDH

**Fig. 5.** Using template for adding new node to a cluster
CONCLUSION

The results showed that Cloudera Express was the easiest in installation and cluster management aspects. However, Hortonworks also provided with graphical user interface for cluster maintenance using open source called Ambari. Being light and flexible, Apache Hadoop was recommended with some trade offs such as complexity of command line uses. In all three distributions, Spark was highly shown great potential to be chosen as a processing platform for Big data. The results of Micro-Benchmark tests, both speed and throughput of using Spark were 10 times better than Hive. The PageRank testing results confirmed Spark (any languages) were 100 times faster than MapReduce. Similar results for K-Means of Machine Learning testing group, Spark were exceptionally superior over Mahout, especially Scale Spark.

REFERENCES

[1] Huang S, Huang J, Dai J, Xia T, Huang B (2010), The HBFoods Benchmark Suite: Characterization of the MapReduce-Based Data Analysis, 2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW), pp. 41-51
[8] Holman J. O’Dee K (2015), How to Deploy Apache Hadoop Clusters Like a Boss, January 2015