

Energy-Saving Cloud Computing Platform Based On Micro-Embedded System

Wen-Hsu HSIEH^{*}, San-Peng KAO^{**}, Kuang-Hung TAN^{**}, Jiann-Liang CHEN^{**}

^{*} Department of Computer and Communication, De Lin Institute of Technology, New Taipei, Taiwan

^{**} Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

wnhsieh742@gmail.com, D10007502@mail.ntust.edu.tw, M9907953@mail.ntust.edu.tw, Lchen@mail.ntust.edu.tw

Abstract— Energy consumption and computing performance are two essential considerations when service providers establish new data centres. The energy-saving cloud computing platform proposed in this study as potential applications in internet network information centres because of its excellent energy efficiency when manage large datasets. Increased data nodes in distributed computing systems greatly enhance data processing capacity. Compared to a standard platform, the proposed energy-saving cloud computing platform achieves the goals of energy-saving and high-performance computing which reduce power consumption by 45.5% and reduce computation time by 22.6%.

Keywords— Energy saving, Hadoop, MapReduce, Cloud computing, Distributed computing, Power consumption.

I. INTRODUCTION

The large amounts of popular applications require heavy computing workloads as well as storage and server demands. Large data centres currently in operation have considerable energy consumption. They also require numerous cooling fans, air conditioners and other cooling mechanisms to reduce the heat generated by processors, which further increases their energy consumption. Therefore, effectively reducing energy consumption for data centres is a critical issue.

Intel introduced the "micro server" concept, in which an inexpensive, energy-saving dual- or quad-core chip of the kind that might normally be used to power a laptop is squeezed onto a small system board to obtain a blade system, smaller than the conventional blade but still powerful enough for data processing.

Another excellent choice is the RISC-based processor ARM (Advanced RISC Machine). Due to the performance requirements of smart handheld devices and consumer products, uses of the 2.5 GHz ARM processor Cortex™ A15 Core have evolved into applications for multiprocessor architectures to provide high computing capability. However, although the computing power of ARM-based processors have substantially improved, most studies of mobile devices have

focused on the access and use of grid resources rather than on using of mobile devices themselves as grid computing nodes [1].

The Apache™ Hadoop™ project [2, 3] develops open-source software for reliable, scalable and data-intensive distributed applications written in the Java programming language. The software was designed to run applications on large clusters using commodity hardware, and a growing number of companies and academic institutions have begun using Hadoop [4-7], which is an open-source version of the Google MapReduce framework for data-intensive computing. The data-intensive Hadoop computing framework is built on a large-scale, highly resilient object-based cluster storage managed by Hadoop Distributed File System (HDFS) [8].

The efficient and energy-saving Hadoop cloud computing platform proposed in this study initially distributes a large data set to multiple nodes. Compared to a standard platform, the proposed energy-saving cloud computing platform achieves the goals of energy-saving and high-performance computing by reducing power consumption by 45.5% and by reducing computation time by 22.6%. The potential application is data-intensive computing in non-severe requirements, such computing for data centres, community websites, etc. Another contribution of this study is to setup Hadoop system on an embedded platform, thus the existed Hadoop service based on x86 platform, could be re-used without re-implement or re-compile.

The remainder of the paper is structured as follows. Section II provides background information about Hadoop MapReduce and HDFS. Section III gives an overview of the system architecture and how the energy-saving cloud computing environment was built. Section IV describes the experimental setting and the results confirming the effectiveness of the system, and section V concludes the paper and suggests future research directions.

II. RELATED WORK

This section presents the key research findings and introduces the Hadoop technology. The Hadoop open source

framework implements the MapReduce parallel programming model and a user-level distributed file system for managing storage resources across the cluster for analysing large datasets. The MapReduce framework effectively and automatically manages distributed computing resources by increasing the number of data nodes, which increases speed when processing large datasets.

Figure 1 shows the component stack of Hadoop. At the bottom is the hardware environment composed of a group of server clusters. Comes up is an HDFS file system for managing distributed file resources. The next MapReduce framework is responsible for the allocation of data nodes and reply collecting results to the user. The top-level services could be composed of cloud applications which are implemented of MapReduce model.

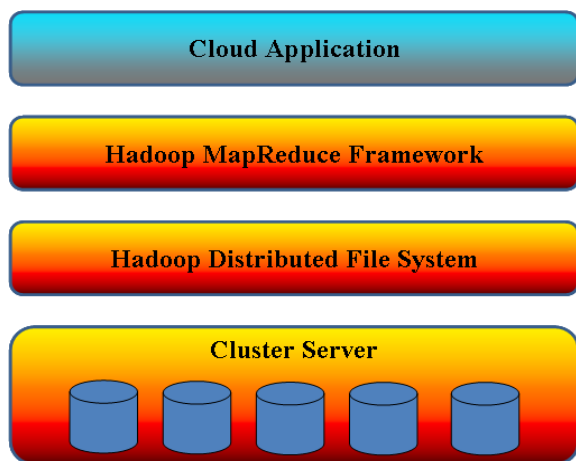


Figure 1. The component stack of Hadoop

A. MapReduce Framework

Hadoop MapReduce was inspired by Google's MapReduce as a mechanism for processing large amounts of raw data [9-11]. A MapReduce task is usually completed in three steps: map, copy and reduce. The JobTracker coordinates the parallel processing of data using Map and Reduce. TaskTrackers nodes with available slots at or near the data have chosen to do Map job to process a set of key/value pairs then produce a set of intermediate key/value pairs. The JobTracker sorts these temporary values then dispatch to proper reducers according to different keys. All values with the same key will be placed in a container, so the reducer could get all values quickly by the values.next() method. When completed, the Client machine can read the result file from HDFS, and the job is considered complete.

B. Hadoop Distributed File System (HDFS)

To manage storage resources across the cluster, Hadoop uses a distributed user-level file system named HDFS, which is written in Java and designed for portability across heterogeneous hardware and software platforms [12]. Hadoop is designed to be highly fault-tolerant and to have sufficiently

high throughput to handle large data sets and run on commodity hardware.

The HDFS cluster is a node group with a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode which keeps the directory tree of all files in the file system, executes file system operations like opening, closing, renaming files and directories and tracks where across the cluster the file data is kept. The DataNodes execute read and write requests from Hadoop clients. The DataNodes also perform block creation, deletion, and replication as instructed by NameNode.

III. THE PROPOSED ENERGY-SAVING CLOUD COMPUTING PLATFORM

This section describes the actual use of Hadoop for data-intensive computing on a energy-saving cloud computing platform.

A. System Architecture

The goal of this study was to exploit the features of a low power ARM process in a distributed computing environment to build a energy-saving cloud computing platform. Use of the Hadoop framework for managing all distributed nodes for distributed computing increases the energy efficiency, fault tolerance, reliability, and scalability of a computing platform. Figure 2 is a diagram of the system concept.

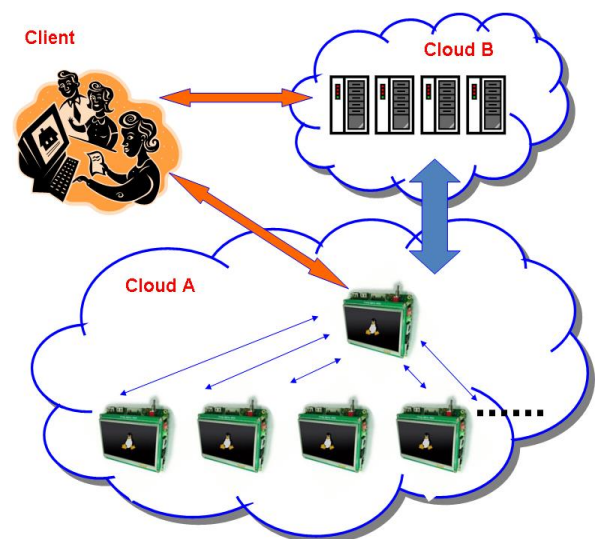


Figure 2. The system concept

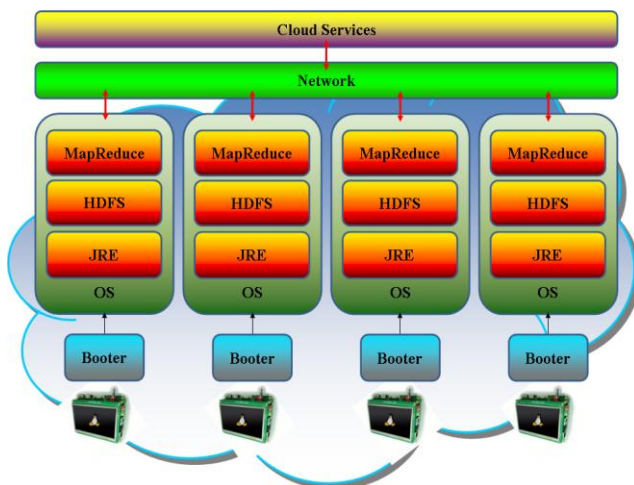
An Intel ® Atom™ N270 processor was used as a control group to simulate the x86-based micro server. DevKit8000 develop kit was used as the experiment group to simulate a energy-saving cloud computing host.

Table 1 shows that, in terms of hardware, the HP MINI 2140 with Intel ® Atom™ N270 processor is much better than DevKit8000 regardless of memory size and processing power.

TABLE 1. HARDWARE FEATURES OF THE HP MINI2140 AND DevKit8000

Hardware Spec.	DevKit8000	HP MINI 2140
Core Processor	OMAP-3530 (ARM Cortex™-A8)	Intel® Atom™ N270
Manufacturing Process	65nm	45nm
Processor Clock	720MHz	1600MHz
L2 Cache	256KB	512KB
Memory	256MB DDR	1G DDRII
Storage	KINGMAX 2GB SD Card	KINGMAX 2GB SD Card
Operating System	Ubuntu 9.10 Embedded	Ubuntu 10.04
JRE Environment	1.6.0_30 for Embedded	1.6.0_30
Hadoop	0.20.2	0.20.2

The Hadoop was originally developed for an x86 based platform, so the main task of the study was porting it to an ARM-based platform. Figure 3 shows the software and system architecture of the proposed energy-saving cloud computing environment. The lowermost hardware layer is DevKit8000. The boot loader layer drives the hardware device and loads the boot program. Ubuntu 9.10 is embedded in the next layer, which is the operating system layer. The application layer then installs the Java virtual machine and builds up HDFS and Hadoop service to provide distributed computing capability. The top layer is the Service layer, in which could provide cloud services based on Hive™, HBase™ or Hadoop MapReduce framework to develop more attractive services.

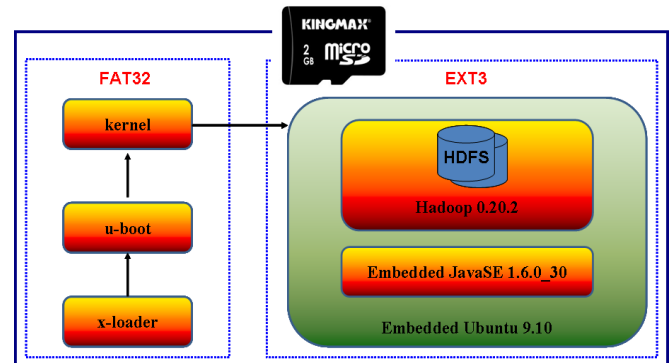
**Figure 3.** The proposed energy-saving cloud computing environment

B. Implementation

This section shows the setup for the energy-saving cloud computing environment. Since DevKit8000 only has 256MB of built-in Nand Flash, it does not meet the space requirements of the system to be installed. To maintain a similar environment, a Kingston 2G SD card was used for system storage in both the HP MINI2140 and DevKit8000.

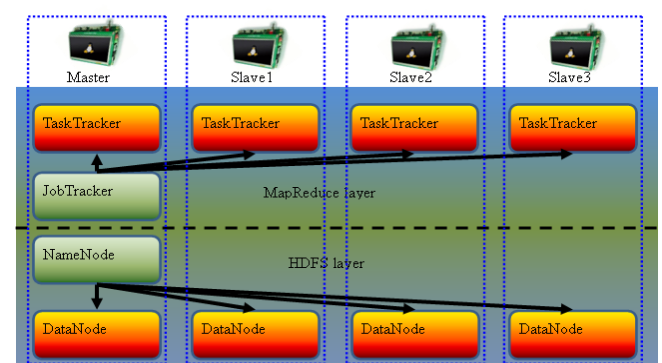
The bootable Kingston 2G SD card has two partitions, one is FAT32 format to store booting sequence program such as x-loader, u-Boot and kernel. Another EXT3 partition is installed

with embedded Ubuntu 9.10 operating system, JavaSE 6 for embedded version and Hadoop 0.20.2. After the installation, JavaSE 6 for embedded could run on DevKit8000 platform and shows the java version is “1.6.0_30”. Figure 4 shows that the system partition includes a boot loader and file system (Operating system, JavaSE 6 for embedded and Hadoop).

**Figure 4.** The system partition shows on Kingston 2G SD card

There are two types of Hadoop cluster, single-node cluster and multi-node cluster. To monitor the performance degradation, we setup a single-node Hadoop cluster and a set of multi-node Hadoop cluster for comparison. In the single-node cluster, the master node plays the role of TaskTracker, JobTracker, NameNode and DataNode. In single-node cluster the replication value of Hadoop was setup to 1. After the setup, you could find one node in Hadoop Map / Reduce Administration page.

As multi-node cluster is an extension of single-node cluster, the master node plays the same role as in single-node cluster. Three slave nodes were added and played as TaskTracker and DataNode show in Fig. 5. In multi-node cluster, the replication value cannot exceed the number of nodes, so the replication of Hadoop was setup to 4 in multi-node cluster.

**Figure 5.** Multi-node Hadoop cluster

Due to the hardware limitations of the DevKit8000 platform, a single machine could only use 256 MB of RAM to run Hadoop MapReduce framework, including NameNode, JobTacker, DataNode and TaskTracker. Therefore, the heap

size of the JAVA environment was modified to avoid the Java heap space problem. The same setting was also applied on HP MINI2140.

IV. PERFORMANCE ANALYSIS

A. Prerequisite

After the energy-saving cloud computing environment was set up as described in Section III, system performance was measured in terms of computing speed and total energy consumption. The size of data could also affect the number of executions of MapReduce task, so it is also the observation object of this study. We based on the data-intensive applications, word count for different file sizes, 64MB, 128MB, 192MB and 256MB to calculate number of words in each file to assess system performance.

The default block size of HDFS is 64MB. Total execution time and total energy consumption were collected for 5 runs of each cycle on different platforms to calculate the average value of process time and energy consumption. During testing, the backlight of the HP MINI2140 was turned off to minimize power consumption. As noted in section III above, the HP MINI2140 and the DevKit8000 used a Kingston 2G SD card for system storage.

B. HP MINI 2140 Test Result

Table 2 shows the average process time and the corresponding energy consumption, J (Second * Watt) recorded for file sizes of 64MB, 128MB, 192MB and 256MB.

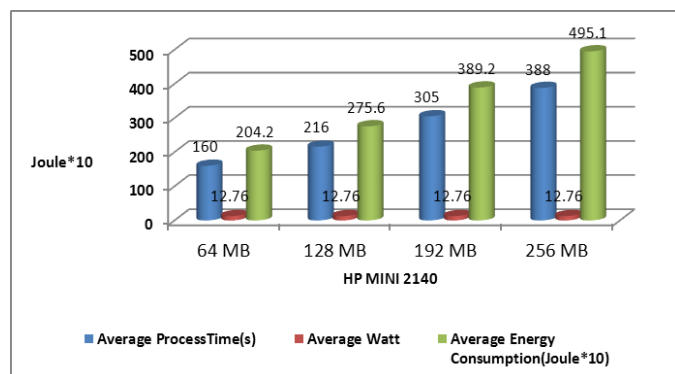


Figure 6. Average energy consumption of the HP MINI2140

C. DevKit8000 Test Result

A single-node Hadoop cluster on one DevKit8000 had a longer process time on 256MB data due to the limited hardware specifications, but had much better energy consumption compared to HP MINI2140.

In DevKit8000, multi-node Hadoop cluster mode showed that two data nodes could process data simultaneously. According to the default HDFS block size is 64MB, so even we got two data nodes at data size is 64MB, only one node was assigned the job. But when data size is 128MB, both of nodes process the data at the same time, that's why we got the

same process time on 64MB and 128MB. Table 3 compares average energy consumption between a single-node Hadoop cluster and multi-node Hadoop cluster with 2, 3 and 4 DevKit8000. When using 4 data nodes simultaneously, all data sizes were completed in the first round of testing.

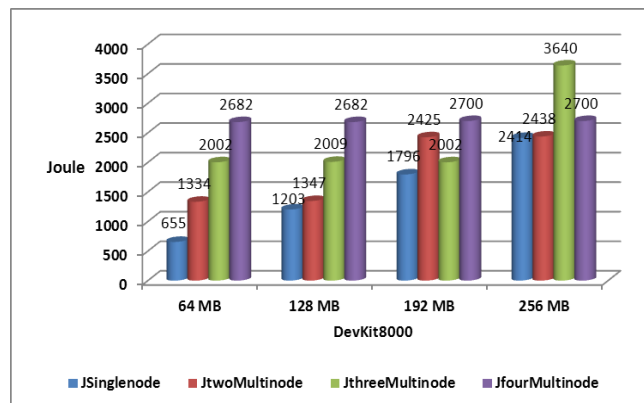


Figure 7. Average energy consumption of the DevKit8000

D. Performance comparison between our energy-saving cloud computing platform and HP MINI2140

Figure 6 shows the average processing time of 256MB for a multi-node Hadoop cluster of four DevKit8000 was 300s, which was 22.6% faster than the 388s processing time obtained for the HP MINI2140. In terms of energy consumption, the 256MB on the multi-node Hadoop cluster of four DevKit8000s consumed 2700 joules, which was 44.5% lower than the 4951 joules consumed by the HP MINI2140. The experiment confirmed the flexibility of the proposed energy-saving cloud computing environment based on Hadoop and the better processing time and energy efficiency when performing the same task.

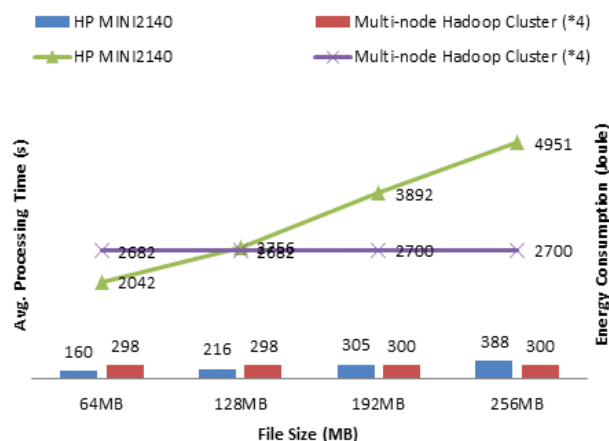


Figure 8. Performance comparison of Hadoop cluster and the HP MINI2140

V. CONCLUSION AND FUTURE WORK

The energy-saving cloud computing platform installed on an ARM-based DevKit8000 embedded with embedded Ubuntu, JavaSE 6 for embedded and ported with Hadoop MapReduce framework achieved a high processing speed with

low energy consumption. By using Hadoop, the platform provides highly scalable distributed computing capability by concatenating multiple DevKit8000 platforms, and the test results show that the multi-node Hadoop cluster reduces average processing time for a large dataset by 22.6% and reduces energy consumption by 44.5% joule compared to the HP MINI2140 in a similar archiving task.

Because of its low energy consumption, the Hadoop cluster is suitable for application in social networking sites, data centres and other non-severe computing server environments that require large amounts of data processing in a high-density cloud computing environment. Therefore, the proposed energy-saving cloud computing platform is suitable for building a high-density server cluster for a green data centres.

Future research could focus on the performance improve for Hadoop framework, and designing a dynamic scheduling mechanism for data intensive applications.

ACKNOWLEDGMENT

The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially/partially supporting this research.

REFERENCES

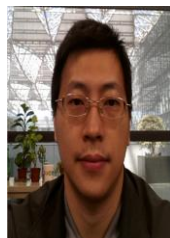
- [1] M. Black and W. Edgar, "Exploring Mobile Devices as Grid Resources: Using an x86 Virtual Machine to Run BOINC on an iPhone," *Proceedings of the IEEE/ACM International Conference on Grid Computing*, pp. 9-16, 2009.
- [2] Hadoop - Apache Software Foundation project home page [<http://hadoop.apache.org/>].
- [3] T. White, *Hadoop: The Definitive Guide*, 1st edition, O'Reilly Media, June 2009, ISBN 9780596521974.
- [4] M. Husain, "Heuristics-Based Query Processing for Large RDF Graphs Using Cloud Computing," *IEEE Transactions on Knowledge and Data Engineering*, vol.23, pp. 1312- 1327, Sep. 2011.
- [5] W. Fang, "Mars: Accelerating MapReduce with Graphics Processors," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 608-620, Apr. 2011.
- [6] R.C Taylor, "An Overview of the Hadoop / MapReduce / HBase Framework and Its Current Applications in Bioinformatics," *Proceedings of the 11th Annual Bioinformatics Open Source Conference (BOSC) 2010*, Boston, MA, USA. July 2010.
- [7] J. Cohen, "Graph Twiddling in a MapReduce World," *Computing in Science & Engineering*, vol. 11, pp. 29-41, 2009.
- [8] S. Konstantin, H. Kuang, S. Radia, and R. Chansler., "The Hadoop Distributed File System," *Proceedings of the Symposium on Massive Storage Systems and Technologies*, 2010.
- [9] J. Dean and S. Ghemawat, "Mapreduce: a Flexible Data Processing Tool," *Commun. ACM*, vol. 53, no. 1, pp.72-77, 2010.

- [10] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, vol. 51, pp. 107-113, 2008.
- [11] J. Dean and S. Ghemawat, "Mapreduce: Simplified Data Processing on Large Clusters," *Proceedings of the OSDI'04*, 2004.
- [12] HDFS™, [<http://wiki.apache.org/hadoop/DFS>].



Wen-Hsu Hsieh was born at Taipei, Taiwan R.O.C. February 9th. 1963. He received the master degree in Computer Science from the University of Oklahoma City, U.S.A. in May 1994.

From August 1986 to May 1990, he worked in the computer center of University of Aletheia as an Engineer. From May 1990 to 1994 May, he persuaded his bachelor and master degree at Oklahoma City University, U.S.A. He was an instructor of the Department of Computer Center, De Lin Institute of Technology from August 1994 to July 1997. From August 1997 to July 2007, he was the instructor of the General Education Center. He was the instructor of the Computer and Communication Engineering Department from August 2008 until now. His research interests include Computer Network, the application of cloud computing, mobile communication and SDN. Currently, Professor Hsieh also is the PhD student of the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C.



San-Peng Kao was received a B.S. degree in Department of Applied Mathematics from National Chung-Hsing University (NCHU), in 1997, and a M.S. degree in Department of Computer Science & Information Engineering from National Dong Hwa University (NDHU), Taipei, Taiwan, in 2001. He had been worked for ODM Company for seven years. He is currently a Ph.D. student in Department of Electrical Engineering of National Taiwan University of Science and Technology (NTUST). His major interests are in Advanced Telecommunication technologies, Internet of Things and Automation Control



Kuang-Hung Tan was received a M.S. degree in Department of Electrical Engineering of National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, in 2012. He had been worked for Telecommunication Company for five years. His major interests are in Advanced Telecommunication technologies, Internet of Things and Distribution Computing.



Jiann-Liang Chen was born in Taiwan on December 15, 1963. He received the Ph.D. degree in Electrical Engineering from National Taiwan University, Taipei, Taiwan in 1989. Since August 2008, he has been with the Department of Electrical Engineering of National Taiwan University of Science and Technology, where he is a professor now. His current research interests are directed at cellular mobility management and personal communication systems.