Towards Green Data Centers

Safae Bourhnane[1(✉)], Mohamed Riduan Abid[2], Rachid Lghoul[2], Khalid Zine-Dine[3], Najib Elkamoun[1], and Driss Benhaddou[4]

[1] Faculty of Sciences, LAROSERI Laboratory, Chouaib Doukkali University, El Jadida, Morocco
s.bourhnane@aui.ma, elkamoun.n@ucd.ac.ma
[2] School of Science and Engineering, Al Akhawayn University, Ifrane, Morocco
{r.abid,r.lghoul}@aui.ma
[3] School of Sciences, Mohamed V University, Rabat, Morocco
zinedinekhalid@fsr.ac.ma
[4] College of Technology, University of Houston, Houston, USA
dbenhaddou@uh.edu

Abstract. Green Computing has been the trend among computer scientists for its eco-friendliness. It serves as a great solution to be integrated with Smart Grids (SG). Data stemming from SGs falls under the realm of Big Data as it is voluminous, various, and has a great velocity. Hence, these data need processing and storage. For this, High-Performance Computing, through clustering a set of computers, proves necessary. Nowadays, with the hardware advances that the world is witnessing, the Raspberry Pi (RP) creates a number of opportunities to deploy cost-effective and energy-efficient clusters, which respect the concepts of Green Computing. In this paper, we are presenting the work done within a USAID sponsored project which aims at developing a SG testbed at Al Akhawayn University in Ifrane, Morocco. We are presenting the deployment of a 5-node cluster based on RPs. The cluster has Hadoop installed and runs the TestDFSIO and Terasort benchmarks for the performance analysis in addition to an energy efficiency analysis.

Keywords: Big Data · Hadoop · Raspberry Pi · HPC · Smart Grids · Green Computing · Cost-effectiveness · Energy-efficiency

1 Introduction

Nowadays, Green Computing constitutes a fashion among IT practitioners and companies. The main motivation behind this shift is the realization that energy consumption is significantly increasing which has a direct impact on the environment. Furthermore, a significant amount of the energy consumed worldwide goes to the manufacturing, storage, operation, and cooling of data centers. This is mainly due to the increasing compute power required by different applications in different fields, and by diverse companies and institutions [1].

J. L. Afonso et al. (Eds.): SESC 2019, LNICST 315, pp. 291–307, 2020.
https://doi.org/10.1007/978-3-030-45694-8_23

Green Computing is a perfect solution for Smart Grids (SG) as it respects the energy efficiency as one of its main building blocks.

SG have been introduced to accommodate for the increasing demand of energy all over the world. The "Grid" refers to the traditional electrical system that is responsible for bringing energy from power plants to end users. The term "Smart" refers to a set of features added to the traditional grid that makes is intelligent. Actually, the smartness of the SG resides in the two-way communication system that allows electricity to flow in both directions: from the power plant to the end-user and the other way around.

The SG has a set of meters, instruments, and equipment that are connected to each other and to the grid. The communication between all the components is done via a specific and well-chosen protocol [2].

Since all the components are connected, we can infer that SGs generate huge amount of data that has all the aspects of Big Data. It goes without saying that Big Data requires two major operations: processing and storage. Bringing up processing, High-Performance Computing (HPC) is the first thing that comes to one's mind. HPC is usually provided through two main venues: supercomputers, and clusters of commodity computers. The solution of supercomputers is no longer adopted because of the high cost of purchase, maintenance, and staff. Companies nowadays are opting for solutions that involve clustering computers to achieve the high performance.

HPC is mostly used in solving advanced and more complicated problems in addition to performing research activities [3]. Now, with the advent of hardware, the Raspberry Pi provides new opportunities to deploy energy-efficient and cost-effective clusters.

Clusters of Raspberry Pi have been trending in the last decade and have been the center of interest of many researchers and practitioners in the field. However, the community did not provide a closer look at the performance of these clusters, nor at their energy efficiency.

In this paper, we are testing the performance of a 5-node Raspberry Pi cluster running Hadoop. For this, we used two main benchmarks: Hadoop TestDFSIO and Terasort. The results obtained have been compared to the ones of a study that was previously carried on using the same benchmarks on a cluster of commodity computers. In addition to that, we are measuring the energy consumption of the cluster when running the Terasort on the biggest dataset of our experiment.

The rest of the paper is organized as follows: Sect. 2 presents the work previously done in using Raspberry Pis to run HPC jobs. In Sect. 3, we present the background of the work done in this paper. Then, in Sect. 4, we describe the experimental setup in addition to the technologies used. The next section presents the results and analysis of the experiments. The last section discusses the conclusions and the future work.

2 Related Work

Raspberry Pi clusters have been the trend of cluster computing for the last years. A significant amount of work has been carried out using Raspberry Pis as a cost-effective and energy-efficient alternative. This section presents different attempts to deploy Raspberry Pi clusters.

Helmer et al. in [4] are describing their work done in deploying a Raspberry Pi cluster that consists of 300 nodes. They are presenting the first steps consisting of setting up and configuring the hardware along with the system software, in addition to the maintenance and the monitoring of the system. Furthermore, they discuss some of the limitations that would hinder the deployment of their cluster. These reside in the low processor speed that does not go beyond 700 MHz, and the card performance that is relatively slow which is explained by the actual design of the flash memory. However, they did not present any benchmarks to test the performance of their cluster.

Iridis-pi is another low-cost cluster that is meant for demonstration [5]. The cluster consists of 64 Raspberry Pis Model B, each having a 700 MHz ARM1176JZF-S RISC processor, and 256 MB of RAM. The cluster is hosted on a Lego chassis. The interconnection between the nodes is done via commodity Ethernet cables. The system has a total of 16 GB of RAM and 1 TB of flash storage capacity. For the numerical compute power assessment, the well-known LINPACK benchmark was used for the single-node performance, and High-Performance LINPACK (HPL) benchmark was used to measure the throughput of the entire cluster. Concerning the benchmarking results, the single-node execution showed a computational performance peak of around 65000 kflops. Also, large problem sizes in HPL benchmark showed a good scalability when increasing the number of nodes, however, the scalability was not significant when it came to small problems because of the network overhead.

A very famous use of Raspberry Pis in data processing is what is known as the Glasgow Raspberry Pi Cluster [6]. In their paper, the authors present the "PiCloud" as a set of clusters of Raspberry Pi devices that emulate the entire stack of the cloud. The cluster contains 56 Model B Raspberry Pi devices that are interconnected using a multi-root tree topology. Each one of the Raspberry Pis uses 16 GB SD card which could support up to 3 co-located concurrent virtualized hosts realized through Linux Containers. Therefore, the virtualization component of the Cloud is provided through the containers and not through virtual machines. The deployment of this cloud is still not mature according to the authors, they are still investigating the adaptation of the *libvirt* framework that allows for an easier and more secure management of the virtual resources. Plus, they have a plan of implementing sophisticated live migration with their PiCloud.

Nick Schot has presented a study about the feasibility of Raspberry Pi based data centers for Big Data applications in [6]. The paper is taking a closer look at the benefits and potentials of using Raspberry Pi in a micro data center with Big Data applications as its main purpose. The author presented an analysis of the performance, the scalability, energy consumption, and ease of management. For this, Hadoop framework was used. As results, the cluster showed a moderate performance with the bottleneck being the SD card and more specifically the random write speed which turned out to be extremely low (1.26 MB/s). Regarding the power consumption, the experimentations revealed that Raspberry Pi requires very little power even when operating under load. One Raspberry Pi consumes about 2 W and remains relatively cool at about 55 °C.

This paper extends the work previously done by assessing the performance using the TestDFSIO and the Terasort benchmarks. We measure the running time for both benchmarks and the energy consumed while running the Terasort benchmark.

3 Background

3.1 Raspberry Pi

Raspberry Pi was first introduced to allow for easier access to computing education in underdeveloped countries. It was first launched in 2012 within the open source ecosystem. The very first board had a single-core, 700 MHz CPU, and no more than 256 MB RAM, while the latest model has a quad-core, 1.4 GHz CPU, and 1 GB of RAM [7].

The very popular use of Raspberry Pi over the world reside in learning programming and building hardware projects, home automation, and industrial applications.

Raspberry Pi is a computer that runs Raspbian Linux operating system, which is open source with a set of open source software running on top of it. From hardware perspectives, it provides a set of General-Purpose Input/output (GPIO) that allow the interaction with the external world through sensors for example. The GPIOs also allow to control electric components through actuators and hence explore the Internet of Things (IoT).

3.2 Hadoop

Understanding Hadoop requires the understanding of the Big Data and its issues related to the traditional processing system.

The Relational Database Management Systems used to focus on structured, semi-structured, and unstructured data. This did not solve the following problems:

- Storage of the colossal amount of data.
- Storage of heterogeneous data.
- Access and processing speed.

Hadoop emerged to solve these problems through bringing a framework that allows for a distributed storage of data for later parallel processing.

Hadoop has two main components: The first one being HDFS (Hadoop Distributed File System), is the component responsible for the distributed storage of the data under different formats. The second component is YARN (Yet Another Resource Negotiator). It takes care of the resource management in Hadoop and allows for jobs allocation.

HDFS can be seen as an abstraction. It is represented as a single unit that is meant to store Big Data stemming from different sources. However, the storage is actually done across multiple nodes in a distributed manner that follows the master-slave architecture.

YARN takes care of all the processing activities by allocating resources and ensuring the scheduling of the tasks. It has two major components: ResourceManager and NodeManager. The ResourceManager is the master node that receives processing requests and then passes parts of them to the NodeManagers. Every Datanode has a NodeManager that is installed and that does the actual processing [8].

3.3 Green Computing

Green Computing refers to the practice of efficient and eco-friendly computing. Many companies have realized that going green would help a lot in maintaining good public

relations and significantly reducing the cost. Hence, Green Computing has been the trend among companies and industries. However, going Green is not straightforward.

According to [9], ICT industry was responsible for 3% of the world energy consumption in 2012. This is supposed to increase by 20% a year.

Green Computing has five core green technologies: Green Data Centers, Virtualization, Cloud Computing, Power Optimization, and Grid Computing.

In addition to that, it has the following benefits:

- Reducing energy consumption of computing resources while performing heavy operations.
- Saving energy in idle states.
- Reducing computing wastes.

3.4 MiGrid Research Project

This section of the paper presents the work done in implementing a SG at Al Akhawayn University in Ifrane as a testbed within a USAID sponsored project. This project aims at developing a holistic testbed platform that integrates smart buildings, renewable energy production, and storage.

The general architecture of the SG is depicted in the figure below (Fig. 1) [10].

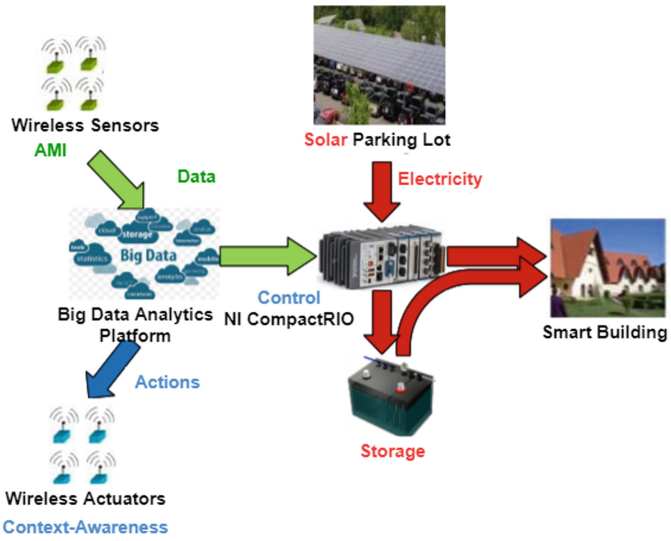

Fig. 1. AUI Smart Grid testbed

The architecture consists of the following elements:

1. Wireless Sensor Network: It is supposed to sense data in the environment where it is deployed. We based our technology choices and architecture on the thesis in [11].

2. Big Data Analytics Platform: It is meant to store and process the big data coming from the wireless sensor network [12]. This platform consists of a cluster of Raspberry Pis, which is implemented and tested performance wise.
3. Wireless Actuator Network: A set of wirelessly connected actuators that take care of translating the signals received into actions.
4. NI CompactRIO Controller: It is the main controlling unit of the system that decides on whether to inject the energy produced to the grid or to store it in batteries.
5. Solar Parking Lot: It is the main renewable energy source.
6. Storage Device: It consists of Hydrogen batteries that are meant to store the excess of energy produced by the solar station for later usage.

In this paper, we are tackling the Big Data Analytics Platform part through building an HPC cluster using Raspberry Pis and assessing its performance. This cluster is deployed in our testbed and hence needs to respect two main constraints: cost-effectiveness and energy-efficiency.

4 Experimental Setup

In order to test the performance of our Raspberry Pi cluster, we ran a set of experiments using two main Hadoop benchmarks: Terasort and TestDFSIO.

Our cluster contains five nodes: one master, and four workers. For each dataset size, we ran the benchmark three times, starting with two nodes up to 4 by adding one node at a time.

The next sections of the paper present the hardware and software requirements and architectures.

4.1 Hardware and Software Requirements

For the sake of this experiment, we made use of the following hardware.

- 5 x Raspberry Pi 3 Model B+
- 5 x HDD 1 TB
- 5 x SD Card 8 GB
- An 8-port switch
- Ethernet cables
- Monitor
- Keyboard
- Mouse

The specifications of the Raspberry Pi used are described in Table 1 below:

Table 1. Raspberry Pi specifications

Spec	Raspberry Pi 3 B+
CPU type/speed	ARM Cortex-A53 1.4 GHz
RAM size	1 GB SRAM
Integrated Wife	2.4 GHz and 5 GHz
Ethernet speed	300 Mbps
PoE	Yes
Bluetooth	4.2
Cores	4

Initially, we used SD cards of 8 GB. We assumed that more space will be required to store all the data stemming from the SG. Thus, we extended the storage to 1 TB using external HDDs.

Regarding the software requirements, the main piece of software used in this experiment is Hadoop version 2.7.

The hardware architecture opted for is depicted in Fig. 2.

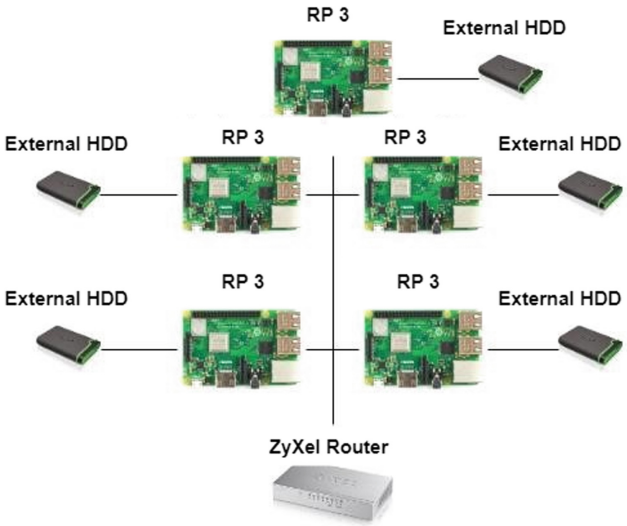

Fig. 2. Hardware architecture

The choice of RPs stems from the nature of the research project and aligns with the energy efficiency concept introduced by the SG. Thus, we are targeting a green

processing unit that costs less than the traditional one, and eventually consumes less energy all without being less performant.

Each one of the physical nodes shown above is running Hadoop according to the following software architecture and based on whether the node is a master or a slave. The software architecture is shown in Fig. 3.

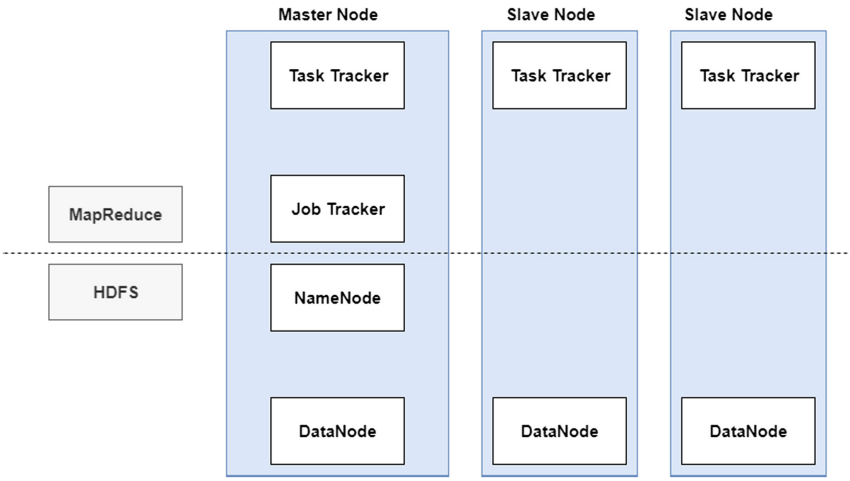

Fig. 3. Software architecture

5 Results and Analysis

5.1 Benchmarks Used

To assess the performance of our Hadoop cluster, we used two main benchmarks: Terasort and TestDFSIO. The Terasort benchmark was used because we sorting is one of the main operations done on SG's data. Also, it is important to test the cluster I/O speed wise as this feature is widely used by SG applications.

The Terasort Benchmark is used to test both Hadoop components: HDFS et MapReduce. It does so by sorting different amounts of data to measure the capabilities of distributing and "mapreducing" files and jobs in a cluster. The benchmark has three main components.

– Teragen: It generates the random data to be sorted.
– Terasort: Sorts the generated data using MapReduce.
– TeraValidate: It is used to validate the output of the Terasort component.

The second benchmark used is TestDFSIO. It is basically a "read" and "write" test for HDFS. It is mainly used as stress test for HDFS to discover the bottlenecks and have an idea about how fast the cluster is in terms of I/O.

The dataset used for the Terasort benchmark and that is given by the Teragen function has the following format: (10 bytes key) (10 bytes row_id) (78 bytes filler). The key consists of random characters (e.g. '~'), the row_id identifies the row by an integer, and the filler consists of characters from A to Z.

5.2 Results

Performance Results

The performance results were obtained through running the Terasort benchmark on the cluster using four different dataset sizes: 100 Mb, 1 GB, 10 GB, and 30 GB.

We tested the cluster using 2, 3, and 4 nodes and we compare the results with the ones obtained in [13]. Each test was performed three times and then the mean time was calculated. The results of running the Terasort are shown in Fig. 4 below.

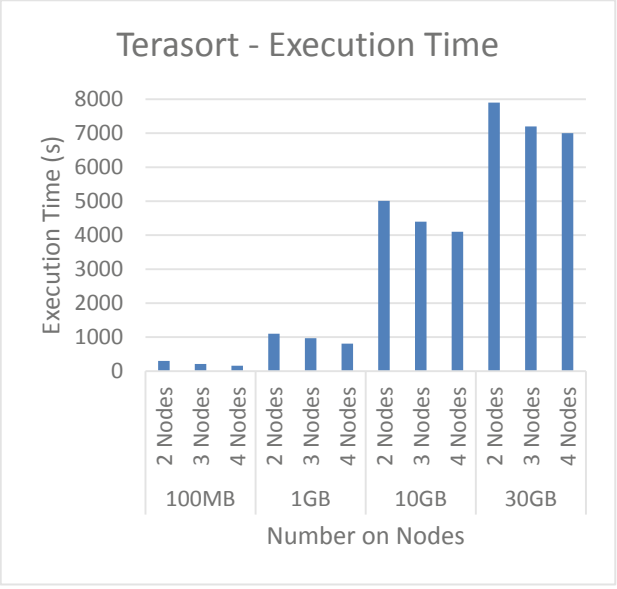

Fig. 4. Terasort execution time

As we can notice from the graph above, scaling up in the cluster helped gaining performance wise as the execution time goes from 299 s to 159 s to sort 100 MB of data, and from 7905 s to 7001 s when sorting 30 GB of data.

To be able to assess the results obtained, we compared them with the results of another cluster of commodity hardware and that runs the same benchmark. This cluster is using the Dell OptiPlex 755 computer with the specification shown in the Table 2 below.

Table 2. Dell OptiPlex 755 features

Characteristic	Value
RAM total memory	975 MB
Disk space	160 GB
Number of processors	2
Processor model	Intel® Core™2 Duo CPU E4500 @ 2.20 GHz
CPU architecture	I386/i686
CPU op-mode(s)	32-bit, 64-bit
Linux Kernel	Distributor ID: Ubuntu Description: Ubuntu 12.04.3 LTS Release: 12.04 Codename: precise

The RPi cluster does not introduce any gain in terms of performance when sorting 30 GB of data using clusters of 3 and 4 nodes. The loss, compared to the normal cluster, was found to be around 20% when working with 3 nodes and we lost up to 75% with a cluster of 4 nodes.

The difference in the execution time of both clusters is shown in the graph of Fig. 5 below.

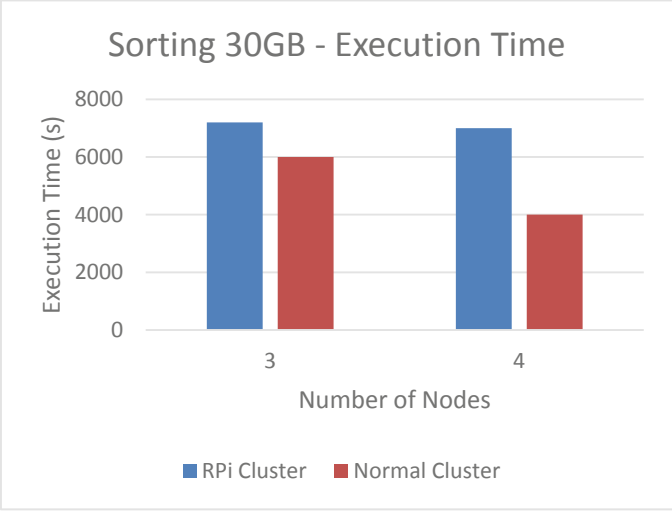

Fig. 5. Terasort execution time of RPi cluster and normal cluster

Stress Testing Results

For the stress testing of the cluster, we used the TestDFSIO benchmark. We start by writing different sizes of files, mainly 100 MB, 1 GB, 10 GB, and 100 GB. Then we read the same files. At each read/write operation we measure the execution time. The results of the write operation are shown in Fig. 6 below.

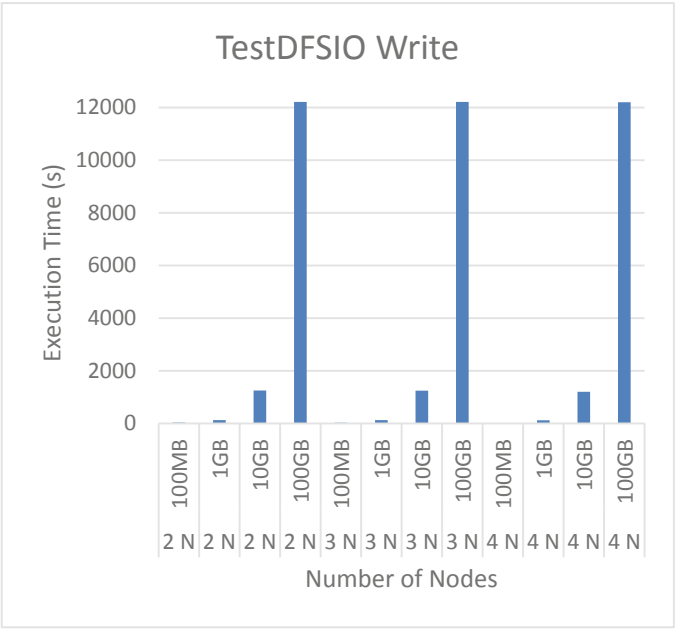

Fig. 6. TestDFSIO write execution time

As shown in the graph above, there is a slight gain in the performance as the execution time decreases when adding nodes to the cluster. The increase of the performance is noticed when dealing with bigger file sizes. However, we cannot really notice the execution time of writing 100 MB as it is very low. It is shown separately in Fig. 7.

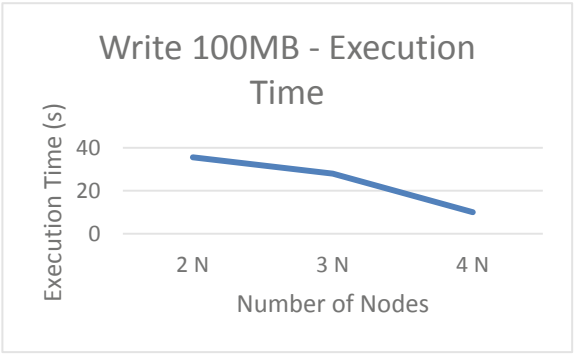

Fig. 7. Write 100 MB - execution time

We are comparing the results obtained with the ones of the commodity cluster described previously. The results of the comparison are shown in Fig. 8.

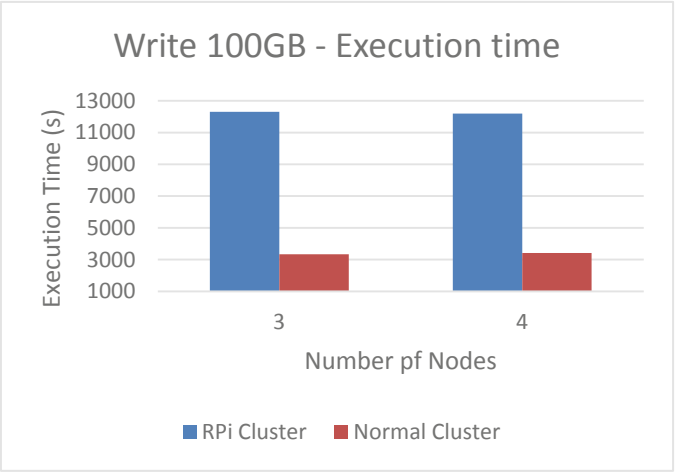

Fig. 8. Write 100 GB execution time of RPi cluster and normal cluster

As we can infer from the graph, there is a drop in the performance of the RP cluster compared to the normal one. The execution time got almost tripled up when writing 100 GB for both clusters.

Next, we are looking at the Read execution time of the same file size. The results obtained are presented in the graph of Fig. 9 below.

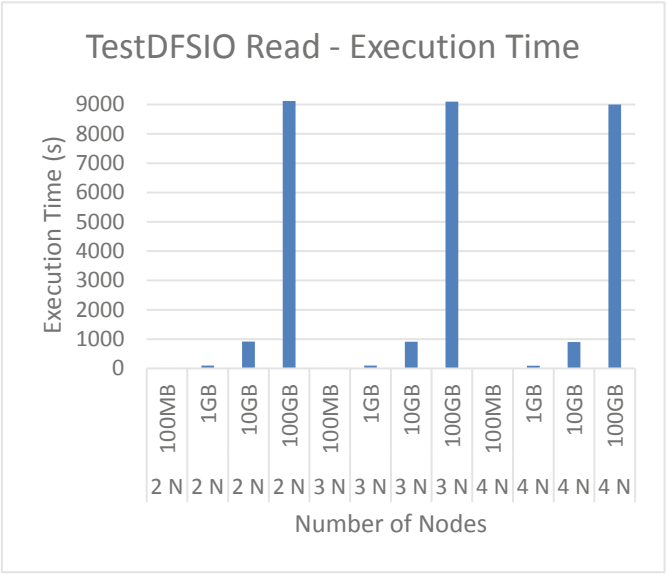

Fig. 9. TestDFSIO read - execution time

The execution time of reading 100 MB is very low compared to the other datasets. We are presenting it separately in the graph of Fig. 10 below.

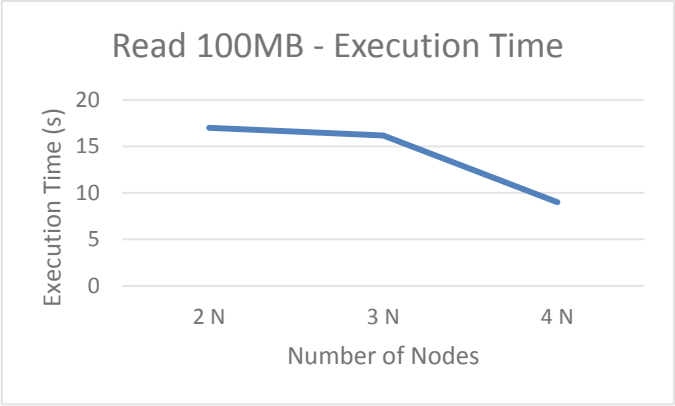

Fig. 10. Read 100 MB - execution time

The comparison between the two clusters is described in Fig. 11 below. As we can notice, we lost a lot in terms of performance as the reading on 100 GB of data using the Raspberry Pi cluster takes almost triple the time compared to the normal cluster. Also, we can notice that scaling up in the cluster does not help gaining performance wise.

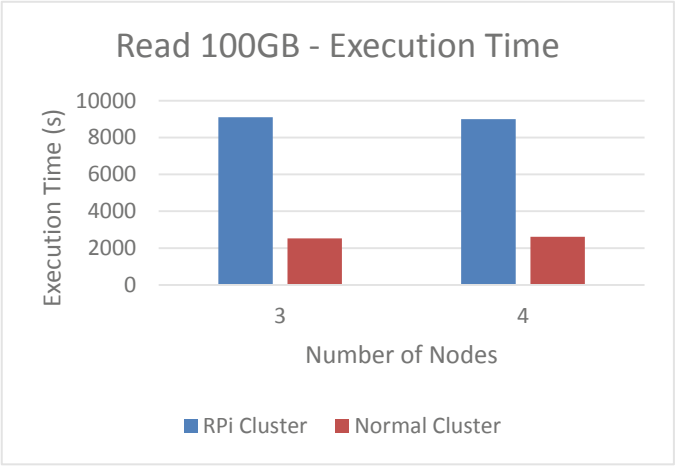

Fig. 11. Read 100 GB execution time of RPi cluster and normal cluster

Energy Consumption

For the energy consumption analysis, we measured the current needed by one Raspberry Pi while performing the Terasort benchmark with 100 MB of data in a 5-node cluster.

We noticed that the current varies between a maximum value of 0.10 A when the Raspberry Pi is performing jobs, and a minimum of 0.04 A when it is in an idle state. The current was measured using a multimeter each second for 159 s (which is the time taken by the cluster to sort 100 MB of data).

In order to calculate the power consumed, we had to multiply the current obtained by the voltage needed for the Raspberry Pi to function which is 5 V, according to the following formula:

Pc = I * V, with I being the current and V the voltage.

The minimum power consumed was found to be 1 W, and the highest was 10 W. The power consumed during the entire working time of the Terasort benchmark with 100 MB is shown in Fig. 12 below.

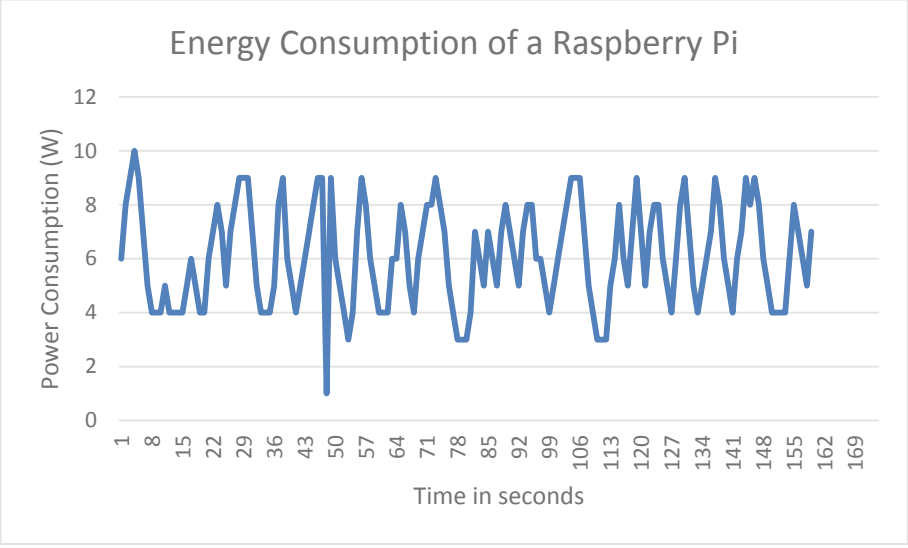

Fig. 12. Power consumption of a Raspberry Pi

5.3 Analysis

From the results obtained, the RPi cluster did not perform well with the TestDFSIO benchmark compared to how a normal cluster would perform. However, and in both read and write operations, the Raspberry Pi cluster performed slightly better when we scaled up in the cluster by adding more nodes.

In the next figures (Figs. 13 and 14), we are looking at the throughput of both read and write operations.

Based on the results of the commodity hardware (CH) cluster described previously in this paper regarding the TestDFSIO benchmark, we can notice that the Raspberry Pi is relatively slow compared to a normal computer. The comparison of the throughput of both cluster (with 3 nodes) and with the same dataset sizes is shown in Fig. 15.

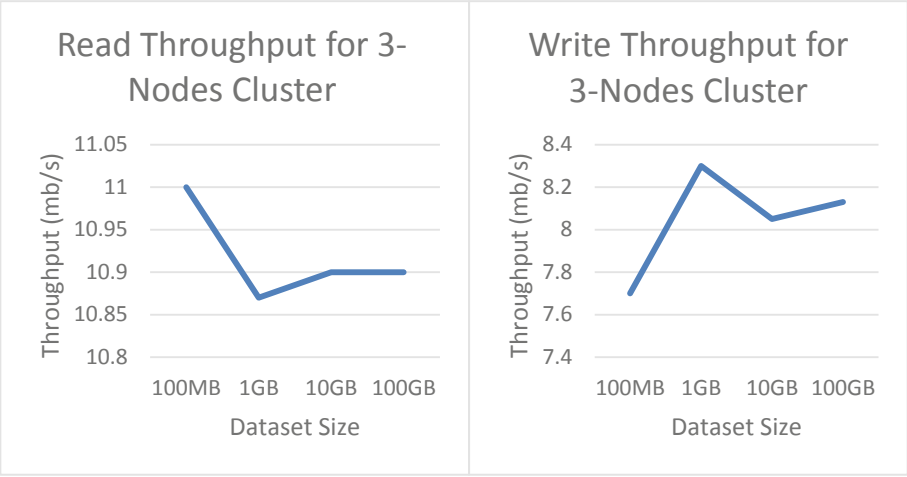

Fig. 13. TestDFSIO throughput

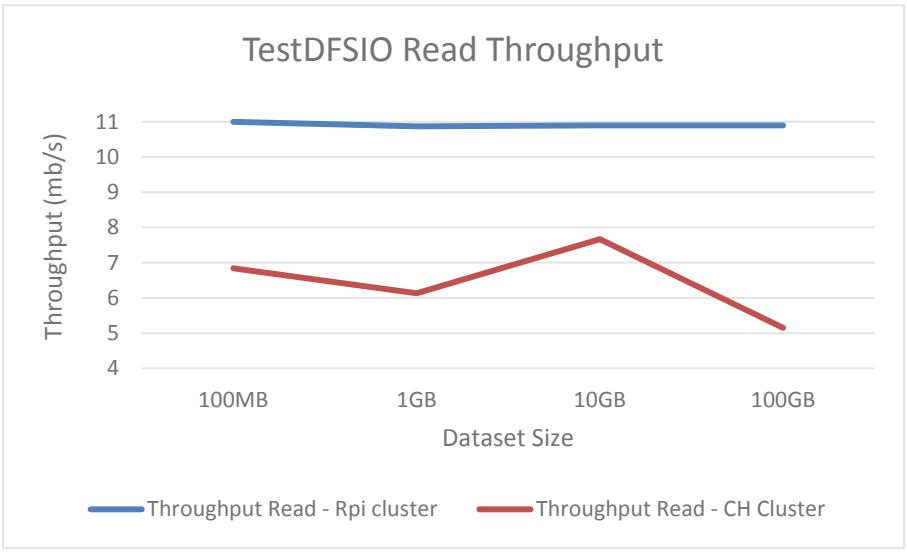

Fig. 14. TestDFSIO read throughput for 3 nodes RPi cluster and CH cluster

Concerning the Terasort benchmark, the Raspberry Pi cluster did not perform better than the traditional cluster, however, the drop in the performance is not as significant as the one with the TestDFSIO benchmark.

The Terasort took longer time to be completed. This is mainly due to the low computing power of the Raspberry Pis. The relatively low performance of the Terasort may be overcome by adding more nodes to the cluster.

By looking at the power consumption of the Raspberry Pi in a working mode, we can notice that the consumption is low which leaves room to add more nodes to the cluster

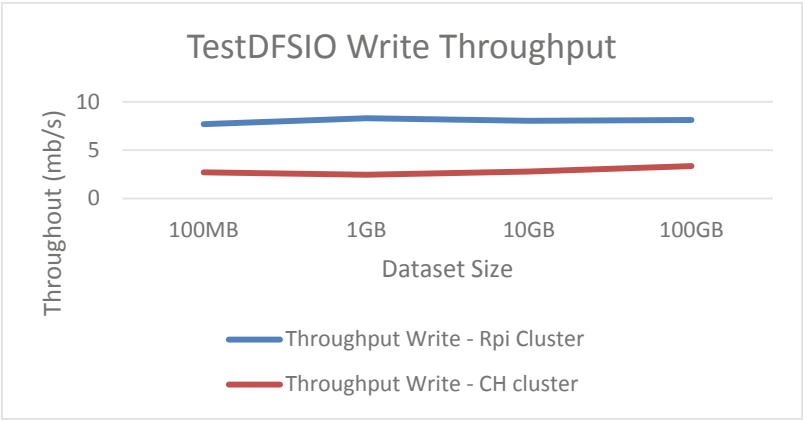

Fig. 15. TestDFSIO write throughput for 3 nodes RPi cluster and CH cluster

without being worried about the energy consumption. Hence, this respects the concept of Green Computing and make our cluster a green one.

We consider this solution to be very suitable for our project since the platform in question is not supposed to perform real-time operations. This does not make the performance an issue to worry about.

6 Conclusion and Future Work

In this paper, we introduced our approach to deploy an energy-efficient and cost-effective Big Data Analytics Platform for Smart Grids. We presented the work done in comparing two different clusters: one based on commodity hardware and the other one based on Raspberry Pis. To do the assessment, we used two different Hadoop benchmarks: Terasort and TestDFSIO.

The results have shown that there was a significant drop in the performance with the TestDFSIO benchmark compared to the traditional cluster. However, we noticed that the Terasort benchmark delivered a little less performance that can be easily overcome by adding more nodes to the cluster.

In addition to the performance assessment, we measured the power consumed by one Raspberry Pi in a working mode within a cluster of 5 nodes. The measurements showed that a Raspberry Pi does not consume a significant amount of power. This means that the cluster can be scaled up without worrying about the power consumed.

Based on the results of this experiment, both performance and energy consumption wise, we can say that our cluster is a green one. Hence, the big data analytics platform conceived for the SG testbed is well suited for the general concept.

As future work, we intend to measure the energy consumption of one machine in a traditional cluster, so that we can compare the energy consumed under the same workload. In addition to that, we will be testing the Raspberry Pi cluster with real data that we will be gathering from our real-world testbed.

Acknowledgment. This work is sponsored by US-NAS/USAID under the PEER Cycle 5 project grant# 5-398, entitled "Towards Smart Microgrid: Renewable Energy Integration into Smart Buildings".

References

1. Ray, I.: Green Computing. Chandigarh Science Congress (CHASCON) (2012). https://doi.org/10.13140/2.1.1546.0164
2. Yacout, D.: An Introduction to Smart Grid. Institutes of Graduates Studies and Research, Alexandria (2013)
3. Techopedia: High-Performance Computing (HPC). Technopedia. https://www.techopedia.com/definition/4595/high-performance-computing-hpc. Accessed 31 May 2019
4. Abrahamsson, P., et al.: Affordable and energy-efficient cloud computing clusters: the Bolzano Raspberry Pi cloud cluster experiment. In: IEEE International Conference on Cloud Computing Technology and Science, pp. 170–175 (2013). https://doi.org/10.1109/cloudcom.2013.121
5. Cox, S.J., Cox, J.T., Boardman, R.P., Johnston, S.J., Scott, M., O'Brien, N.S.: Iridis-pi: a low-cost, compact demonstration cluster. Cluster Comput. **17**(2), 349–358 (2013). https://doi.org/10.1007/s10586-013-0282-7
6. Tso, P., et al.: The Glasgow Raspberry Pi cloud: a scale model for cloud computing infrastructures. In: Distributed Computing Systems Workshops (ICDCSW). IEEE (2013). https://doi.org/10.1109/icdcsw.2013.25
7. OpenSource: What is Raspberry Pi?. https://opensource.com/resources/raspberry-pi. Accessed 29 Mar 2019
8. Sinha, S.: What is Hadoop? Introduction to Big Data & Hadoop, 22 May 2019. https://www.edureka.co/blog/what-is-hadoop/. Accessed 01 June 2019
9. Jindal, G., Gupta, M.: Green computing "Future of Computers". Int. J. Emerg. Res. Manag. Technol., 14–18 (2012)
10. Abid, M.R., Lghoul, R., Benhaddou, D.: ICT for renewable energy integration into smart buildings: IoT and big data approach. In: IEEE AFRICON (2017). https://doi.org/10.1109/afrcon.2017.8095594
11. Abid, M.R.: Link Quality Characterization in IEEE 802.11s Wireless Mesh Networks. Auburn University, Auburn (2010)
12. Achahbar, O., et al.: Approaches for high-performance big data processing: applications and challenges. In: Big Data: Algorithms, Analytics, and Applications (2015). https://doi.org/10.1201/b18050
13. Achahbar, O., Abid, M.R.: The impact of virtualization on high performance computing clustering in the cloud. Int. J. Distrib. Syst. Technol. **6**, 65–81 (2015)