Plataforma de HPC portable de bajo consumo energético para aplicaciones de minería de datos

Portable low-power High Performance Computing (HPC) Platform for data mining applications

Baixo consumo de energia portátil plataforma HPC para aplicações de exploração de dados

DOI: http://dx.doi.org/10.23913/reci.v6i11.62

Armando Saúl Carranza Sánchez
Instituto Tecnológico de Colima, México
G1546005@itcolima.edu.mx

Jesús Alberto Verduzco Ramírez
Instituto Tecnológico de Colima, México
averduzco@itcolima.edu.mx

Nicandro Farías Mendoza
Instituto Tecnológico de Colima
nfarias@itcolima.edu.mx

Francisco Cervantes Zambrano
Instituto Tecnológico de Colima, México
francisco.cervantes@itcolima.edu.mx

Fernando Rodríguez Haro
Universidad de Colima, México
ferharo@ucol.mx

Resumen

La necesidad de plataformas computacionales que proporcionen soporte a las aplicaciones denominadas intensivas ha estado incrementándose en muchas organizaciones debido al surgimiento de una serie de aplicaciones que requieren de manejo intensivo de datos y tiempos de respuesta cortos. Un ejemplo pueden ser las aplicaciones que utilizan la minería de datos. El hardware ha evolucionado de manera acelerada para satisfacer estas necesidades del cómputo intensivo. Por otro lado, pequeñas computadoras montadas en tarjetas denominadas SoC (System on Chip) (R., G., y M., 2013) han aparecido a partir de la miniaturización de componentes y la popularización de los sistemas embebidos, El ejemplo más significativo es Raspberry Pi (raspberrypi, 2016), una alternativa de bajo costo y consumo energético reducido que permite realizar tareas de cómputo. Debido a su popularidad, los fabricantes se han dedicado a incrementar su desempeño generando modelos con mayores prestaciones, los cuales se han vuelto una opción viable para el procesamiento intensivo de datos. Aquí analizamos dicho aspecto.

Para ello se describe la construcción de una plataforma de HPC basada en tarjetas SoC, lo que permite desarrollar y ejecutar aplicaciones de minería de datos. Los resultados obtenidos en las diferentes pruebas de operación y tolerancia a fallas muestran que dicha plataforma ofrece el rendimiento y la robustez necesarios para ser utilizada en el desarrollo de aplicaciones de minería de datos orientadas a la educación y también para la enseñanza de la disciplina antes mencionada, con una inversión que es posible alcanzar en instituciones académicas y pequeñas organizaciones.

Palabras clave: clúster, sistemas embebidos, computación de alto rendimiento, Sistema en Chip (SoC), minería de datos.

Abstract

The need for computing platforms that provide support to the so-called intensive applications has been increasing in many organizations because of the emergence of a series of applications that require intensive management of data and short response times. An example may be applications that use data mining. The hardware has evolved at an accelerated pace to meet these needs of data-intensive computing. On the other hand, small computers mounted on cards called SoC (System on Chip) (R., G., and M., 2013) have appeared from the miniaturization of components and the popularization of embedded systems, the most significant example is Raspberry Pi (raspberrypi, 2016), a low-cost alternative and reduced energy consumption that allows computing tasks. Because of its popularity, manufacturers have dedicated themselves to increase their performance by generating models with further benefits, which have become a viable option for data-intensive processing. Here we analyze that aspect.

Describes the construction of a High Performance Computing (HPC) platform based on SoC cards, allowing you to develop and run applications of data mining. The results obtained in the different operation and fault tolerance tests show that this platform offers the performance and robustness needed to be used in the development of applications of data mining oriented to education and also for the teaching of the above discipline, with an investment that is possible to achieve in academic institutions and small organizations.

Key words: cluster, embedded systems, High Performance Computing (HPC), System on Chip (SoC), Data mining.

Resumo

A necessidade de plataformas que fornecem suporte para aplicações intensivas chamada computação tem vindo a aumentar em muitas organizações, devido ao surgimento de uma série de aplicações que requerem gerenciamento de dados intensivos e tempos de resposta curtos. Um exemplo seria aplicações utilizando mineração de dados. O hardware tem evoluído a um ritmo acelerado para atender a essas necessidades de computação intensiva. Além disso, pequenos computadores montados em cartões de chamadas SoC (System on Chip) (R., G., M., 2013) têm aparecido desde a miniaturização de componentes e popularização de sistemas embarcados, o exemplo mais significativo framboesa Pi é (Pi framboesa, 2016), uma alternativa de baixo custo e de baixo consumo de energia que permite que as tarefas de computação. Devido à sua popularidade, os fabricantes têm dedicado-se a aumentar o seu desempenho gerando modelos de desempenho mais elevados, que se tornaram uma opção viável para o processamento de dados intensivos. Aqui analisamos este aspecto.

Para esta construção de uma plataforma HPC baseados cartões SoC descrito, permitindo desenvolver e executar aplicações de mineração de dados. Os resultados obtidos nos vários testes operacionais e tolerância a falhas mostram que esta plataforma fornece o desempenho ea robustez necessária para utilização no desenvolvimento de mineração aplicações orientada a dados educação e também para o ensino da disciplina acima com um investimento que pode ser conseguido em pequenas organizações e instituições acadêmicas.

Palavras-chave: clusters, sistemas embarcados, computação de alto desempenho, System on Chip (SoC), mineração de dados.

Fecha Recepción: Agosto 2016 **Fecha Aceptación:** Diciembre 2016

Introduction

High performance computing (HPC) is the use of parallel processing to run applications efficiently, reliably and quickly (Garcia Nocetti, 2014). HPC computing systems have been used as tools for the development and execution of intensive applications such as computational simulations and computation of complex operations whose solution would involve an excessive time in conventional computing equipment. The HPC is based on the use of computers equipped with redundancy of hardware; Examples of these teams are cluster, supercomputers, among others.

One of the major drawbacks of the HPC is the high cost, classified in the cost of acquisition and the cost of maintenance performed during the life of these equipment. These two factors make the HPC a technology reserved for organizations that have sufficient financial resources. Rajovic et al. (2014) describe that HPC systems are present in the processing of large amounts of data. These systems have, among other disadvantages, the high energy consumption required for their operation and also for operating the cooling system. As a result, organizations that manage small budgets, such as small universities, are left out of having such facilities, which puts them at a distinct disadvantage.

The Mont Blanc project (Valero et al., 2013) proposes an alternative focused on the use of low energy consumption technologies for HPC. Basically, the idea is to use SoCs, which offer advantages over traditional HPC equipment by mitigating the mentioned aspects (Rajovic et al., 2014).

The need for the services provided by the HPC continues to increase in all types of organizations. An example is data mining, which requires high-performance equipment for its operation due to the magnitude of the data and processes it generates. Currently the term HPDA (High Performance Data Analysis) is handled to refer to the computation of high performance applied to the data mining.

This project seeks to create an economic HPC platform with a low energy consumption that is an accessible alternative for organizations with a reduced budget, interested in the use of data mining to improve their processes. The document is developed according to the following parts: proposal of solution, description of the architecture, design, implementation, results and conclusions.

Proposed Solution

In order to implement our idea, the following technical characteristics have been considered for the design and implementation of this platform:

- **Low cost.** One of the main objectives of this project is that the cost of this platform is reduced and thus a cost-effective option for academic institutions or small organizations that normally handle budgets that prevent to buy expensive supercomputer equipment.
- **Reduced energy consumption.** The platform should be geared towards energy savings by providing high performance services, but with a minimum energy consumption. To this end, the components selected for this platform require 5 volts and 2.5 amperes to operate, consuming 12.5 watts per hour. This implies a low power consumption, compared to a personal computer that oscillates between 300 and 600 watts per hour.
- **Facilities and space.** Our solution must occupy for its physical installation a reduced space, compared to the supercomputers that require large installations for its operation. The platform will be composed of elements with dimensions of 11 x 8 x 1.4 cm, which allow to comply with this requirement established at the beginning of the project.
- **Processing power.** The built platform must provide sufficient computing capacity to process educational applications and support training in data mining techniques. In order to obtain necessary processing power, a cluster was integrated with 24 cards that together provide a theoretical yield of 768 Gflops and an estimate in Linpack (A. Petitet, 2016) of 614.40 Gflops and a consumption of 300 W / H.

Architecture Design

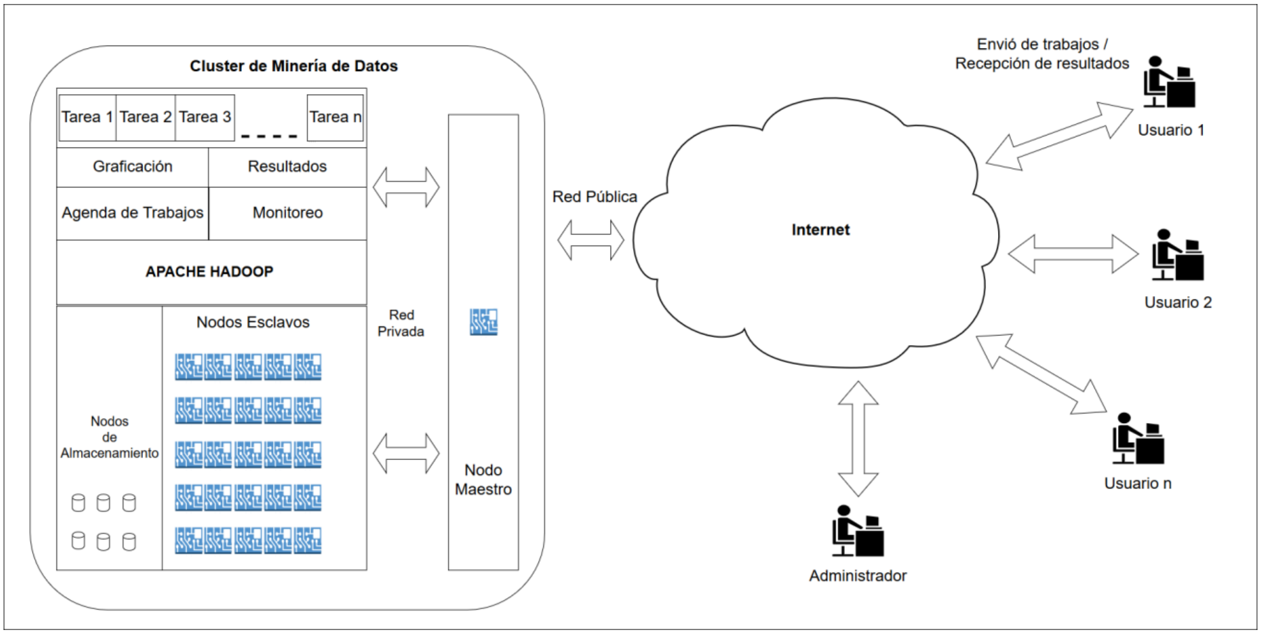
The platform is aimed at providing functional high-performance computing infrastructure services with the particular aim of being used in data mining applications. Figure 1 shows the conceptual model of the platform.

The architecture consists of a set of functional modules, which are described below:

Hardware Modules

- **Master node.** This node is the entry point for cluster users when they want to process tasks. Also, this node assigns tasks to the rest of the nodes, controlling the data input and output, and also provides support for the monitoring function of the cluster nodes.
- **Slave nodes.** They perform the processing of tasks designated by the master node.
- **Storage nodes.** Set of cluster nodes that have a hard disk to store the tasks and the result of their processing.
- **Private interconnection network.** It is the data network that interconnects the master, slave and storage nodes.
- **Public interconnection network.** This connection allows the access and use of the architecture from any computer connected to the internet.

Figure 1. Platform Conceptual Model.



Source: Verduzco et al., 2017

Software Modules

- **Results.** In this module the data obtained from a task performed will be accessible to the user; These datasets can be consulted when the user requires it.
- **Graphing.** In this module the results obtained are shown in a format accompanied by graphs that facilitate their analysis and interpretation.
- **Work agenda.** This module will schedule the tasks submitted by the users.
- **Monitoring.** This module will allow monitoring the operational state of the platform and the verification of the operation of each node.
- **Apache Hadoop.** It is the main module, which establishes an environment that allows to manage the processing of tasks and the distributed storage between the nodes.

Implementation of the architecture

The technical details of the architecture and the software used are described below.

Configuring the Cluster

For the creation of the platform we used 24 Cubieboard nodes of the models A80 and Truck Plus (table 1). The operating systems supported by the cards are: Android (Google, 2016) and Lubuntu (Lubuntu, 2016). For the project the Lubuntu operating system was installed. The cluster was configured in the master-slave architecture, designating a single node as master and the remaining nodes as slaves.

Table 1. Types of SoC used in the project.

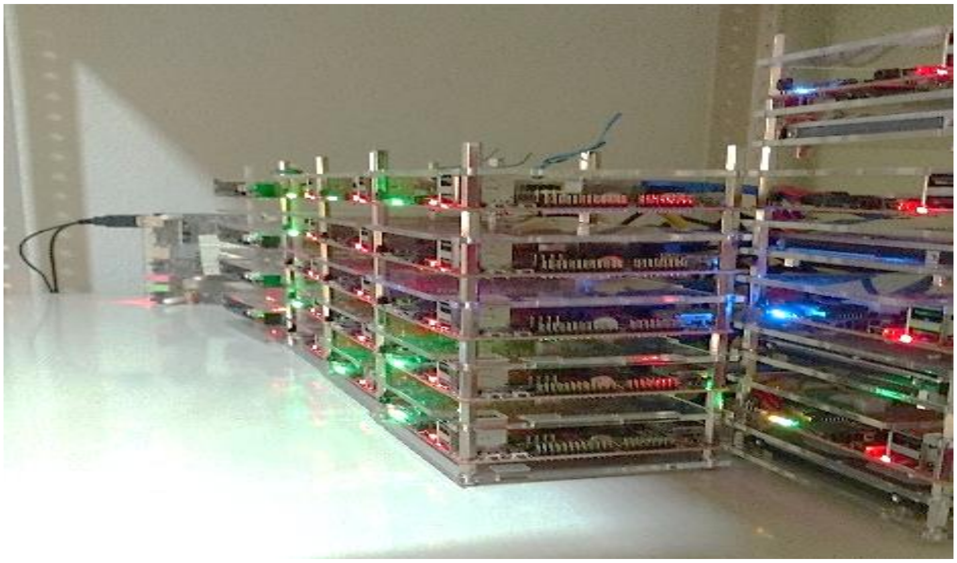
Tarjeta	Procesador	Almacenamiento	Red
Cubie board 4	Allwinner A80 Octa Core 4 x Cortex-A15 to 2016 MHz 4 x Cortex-A7 at 1320 MHz RAM: 2GB DDR3	8GB eMMC en memoria interna y 64 GB en SD	10M / 100M / 1G Gigabit Ethernet Wi-Fi with external antenna connection Bluetooth 4.0
Cubie Truck Plus	SoC A83T/H8 @ 2Ghz DRAM 2GiB DDR3 @ 672MHz (SK hynix H5TQ4G83AFR * 2)	NAND 8GB eMMC en memoria interna y en HDD hasta 2TB	10M / 100M / 1G Gigabit Ethernet

Source: Verduzco et al., 2017

Installing Apache Hadoop

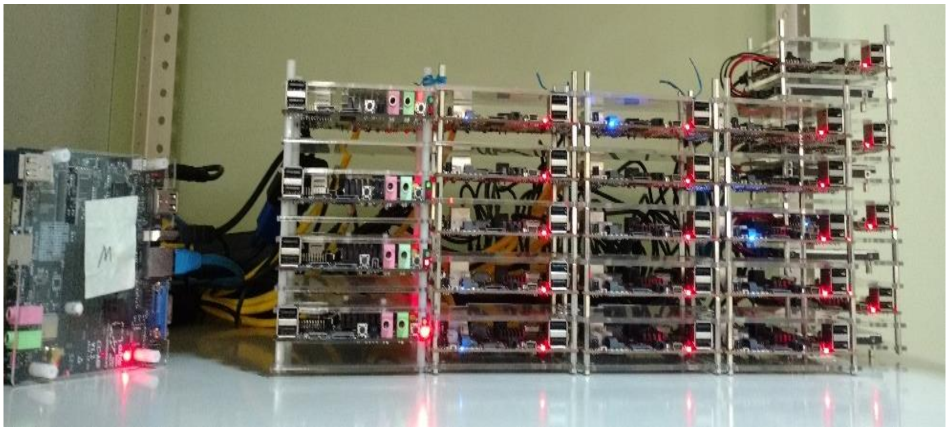
It was decided to install the Hadoop platform (Apache Hadoop, 2016), mainly because it is widely used both in academic institutions and companies to perform data analysis, in addition to the variety of frameworks it supports. The installation method consisted in configuring the Apache Hadoop and Apache Pig programs (Apache Pig, 2017) on a single node, to later replicate the remaining nodes. Figures 2 and 3 show the cluster in operation.

Figure 2. Image of the mining data cluster in operation.



Source: Verduzco et al., 2017

Figure 3. Side view of the data mining cluster in operation.

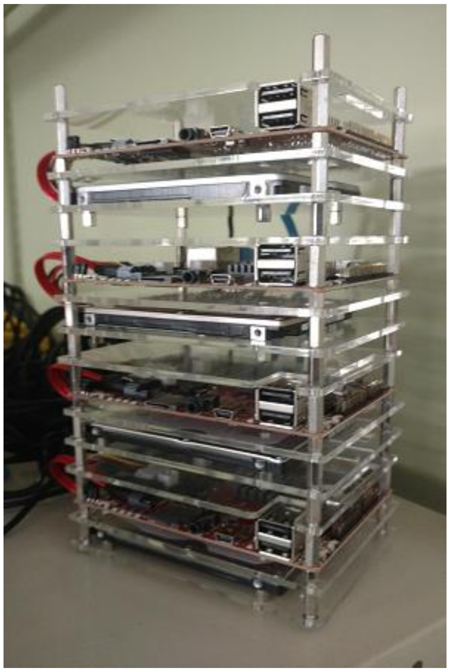


Source: Verduzco et al., 2017

Storage Cluster

The cluster requires dedicated storage to accommodate the tasks and generated data. For this purpose, the storage cluster consisting of four nodes was configured. Each node was fitted with a 1TB hard disk.

Figure 4. Storage Cluster Image.



Source: Verduzco et al., 2017

Results

In order to determine the functionality of the platform, it was decided to carry out different tests classified in: operation, fault tolerance and stress.

Testing environment

The environment in which the tests were run is as follows:

- One 28-port Gigabit Ethernet LINKSYS switch.
- Five cuebieboard a80 octa-core cards at 2.0 Ghz, 2 GB RAM, 8 GB in NAND memory.
- A node of the five was established as a master node, which will assign processing tasks to the slave nodes.
- 19 cards cubie truck plus octa-core at 2.0 Ghz, 2 GB RAM, 8 GB in NAND memory.
- Four of these cards added a hard drive to handle the storage of input and output files.
- Operating system Lubuntu version 14.04 Kernel 3.4.39
- Apache Hadoop versión 2.7.2
- Software Apache Pig versión 0.16

Operation test

To carry out this test, a set of data resulting from an air quality study carried out in different provinces of Spain was selected. Each registration takes place on the first day of each month during the period from 1997 to 2013, totaling 153 121 records. Next, the set of instructions for processing this task is shown in Figure 5 and the results obtained in Table 2.

Figure 5. Algorithm used in the execution of the test of operation.

```
measure = load '/pruebas/calidad_del_aire_cyl_1997_2013.csv' using PigStorage(';') AS
(date:chararray, co:float, no:float, no2:float, o3:float, pm10:float, sh2:float, pm25:float,
pst:float, so2:float, province:chararray, station:chararray);
dump measure;
filter_measure = filter measure by date != 'dia';
measure_by_province = group filter_measure by province;
num_measures_by_province = foreach measure_by_province generate group,
AVG(filter_measure.co) as measure;
DUMP num_measures_by_province.
```

Source: Ramos, 2014

Table 2. Operation test results.

Provincia	Media de Carbón en el Aire de 1997 – 2013
León	0.98
Soria	0.18
Burgos	0.86
Zamora	0.84
Ávila	0.96
Segovia	1.01
Palencia	1.17
Salamanca	1.38
Valladolid	0.68

Source: Verduzco et al., 2017

Fault tolerance test

The purpose of this test is to measure the impact of node failure on the cluster when running an algorithm. For this, the functionality offered by the file system with which Hadoop works, which allows the establishment of the number of replications of data segments distributed in the nodes of the cluster, was taken into account. For this test, the following variables were considered: number of nodes deactivated and index of replication, in order to determine the degree of success that the task will be completed and the memory consumption caused by the selected replication index. The selected task was processed ten times in order to obtain the fault tolerance indicators mentioned above. Table 3 shows the results obtained.

Table 3. Results of the fault tolerance test.

Índice de replicación	Nodos desactivados	% Éxito al completar tarea	Consumo redundante de almacenamiento
1	2	10.0 %	0.11
3	4	20.0 %	0.24
6	8	40.0 %	0.61
9	4	20.0 %	0.24
12	4	20.0 %	0.24
15	4	20.0 %	0.24
18	4	20.0 %	0.24
21	4	20.0 %	0.24
23	18	90.0 %	4.14

Source: Verduzco et al., 2017

Also, in table 3 it is visible that when configuring the platform with a replication index equal to one and with two nodes out of operation, it generates a success of 10% in the completion of the task, which indicates that the configuration of the index Replication is not optimal for the platform. With the replication configuration in six and eight nodes deactivated a success of 40% was obtained, which is a candidate to the optimal configuration. With the replication index in 23 and 18 nodes deactivated, a success rate of 90% is obtained, which guarantees that the task will be developed if at least six nodes work.

The drawback of replication index configurations is the redundant storage consumption caused by the multiple copies of data resident on the nodes.

Stress test

This test bench is intended to measure the processing time of the mining cluster. For this purpose, a data collection of 200 thousand, 400 thousand, 600 thousand, 800 thousand and 1 million records was taken and, in order to establish execution times based on the numerical trend of the data, the collections of Data previously mentioned in the following numerical distributions, Normal distribution, Bernoulli, Chi-square, Hypergeometric, Laplace, Lognormal, Poisson, Uniform (Solano and Alvarez, 2005). The algorithm that was applied to each group of records consists of the reading processing. The following are the instructions used:

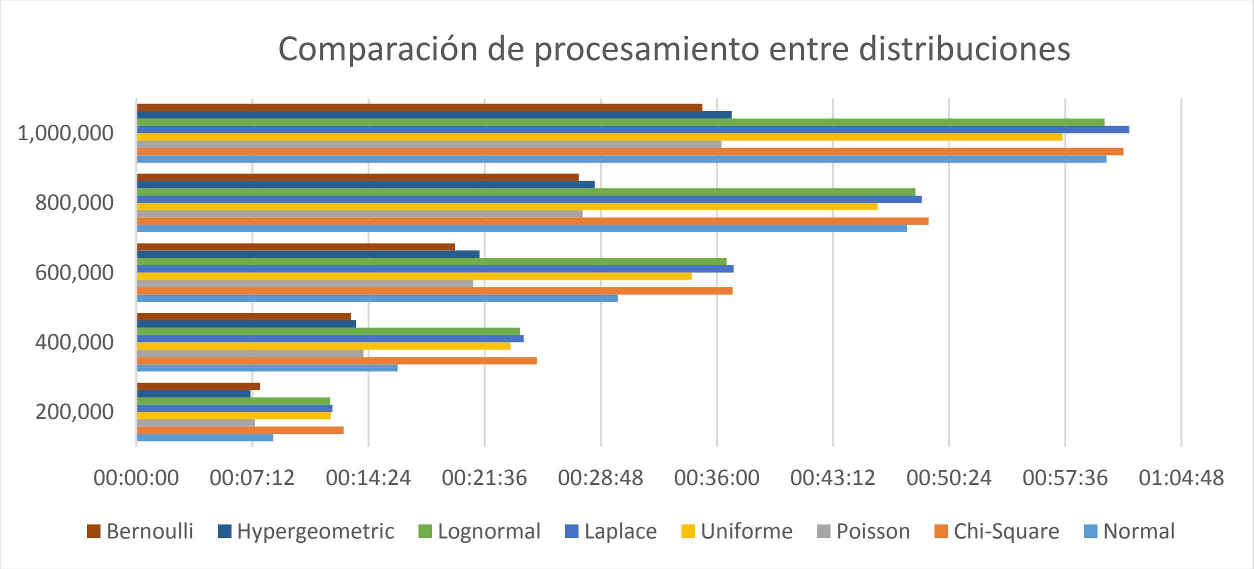
Figure 6. Instructions executed for the stress test.

```
numeros = load '/numeroschiq/200milchiq.CSV' using PigStorage(',') AS (col1:float, col2:float, col3:float, col4:float, col5:float, col6:float,col7:float, col8:float, col11:float, col10:float);
STORE numeros INTO 'pig_output_numeros1millonbernoulli' USING PigStorage('\t');
```

Source: Verduzco et al., 2017

Each data collection was processed three times and the average time required to complete the operation was obtained. Figure 6 shows the results obtained.

Figure 7. Comparison of processing between distributions.



Source: Verduzco et al., 2017

As can be seen in Figure 7, the execution times vary depending on the numerical distribution of the data. The distributions with the longest execution time are Chi-square and Laplace.

CONCLUSIONS

This document shows the implementation of a platform oriented to the development and execution of data mining applications. What is remarkable about this platform is that it is made up of SoC technologies, which implies a reduced cost. Other aspects to emphasize are the reduced space occupied by its facilities and the low energy consumption associated with its operation. The different tests made allow us to affirm that this platform provides the necessary functionality to execute algorithms oriented to university education, research and training. All of the aforementioned aspects make this platform an interesting alternative for low budget institutions that want HPC systems.

Future work will be oriented to improve the usability of the platform by setting up different frameworks specialized in data mining.

Bibliography

A. Petitet, R. C. Whaley, J. Dongarra, A. Cleary (15 de Diciembre de 2016). *HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers*. Obtenido de HPL - A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers: http://www.netlib.org/benchmark/hpl/

Apache. (05 de Octubre de 2016). *Apache Hadoop*. Obtenido de Apache Hadoop: http://hadoop.apache.org/

Apache. (15 de 01 de 2017). *Apache Pig*. Obtenido de https://pig.apache.org/

Barr, M., & Massa, A. (2006). *Programming Embedded Systems: With C and GNU Development Tools*. O'REILLY.

Conaway, S. (03 de 07 de 2014). High Performance Data Analysis: Big Data Meets HPC. *High Performance Data Analysis: Big Data Meets HPC*. Recuperado el 14 de 02 de 2017, de http://www.scientificcomputing.com/blog/2014/03/high-performance-data-analysis-big-data-meets-hpc

Cubieboard. (9 de Septiembre de 2016). Obtenido de cubieboard: http://cubieboard.org/

Díaz, G. (31 de Mayo de 2016). Modelos de Programación Paralela. Merida, Venezuela.

F. Cloutier, M., Paradis, C., & M. Weaver, V. (2014). Design and Analysis of a 32-bit Embedded High-Performance Cluster Optimized for Energy and Performance. *Hardware-Software Co-Design for High Performance Computing (Co-HPC), 2014*. doi:10.1109/Co-HPC.2014.7

Garcia Nocetti, F. (Junio de 2014). *Cómputo de Alto Rendimiento (HPC) & Big Data.* Obtenido de Cómputo de Alto Rendimiento (HPC) & Big Data: http://www.inegi.org.mx/eventos/2014/big-data/doc/P-DemetrioGarcia.pdf

Google. (15 de 10 de 2016). *https://www.android.com/*. Obtenido de https://www.android.com/

HADOOP. (2014). *HADOOP big data analysis framework.* tutorialspoint.

J. Greaves, D. (2011). System on Chip Design and Modelling. *System on Chip Design and Modelling*. Cambridge, Inglaterra.

Lubuntu. (05 de Octubre de 2016). Obtenido de Lubuntu: http://lubuntu.net/

Pérez López, C. (2008). *Minería de Datos Técnicas y Herramientas.* Madrid: Thomson.

R, R., G, M., & M, A. P. (2013). System on Chip (SoC) for Telecommand System Design. *International Journal of Advanced Research in Computer and Communication Engineering*, 1580-1585.

Rajovic, N. R.-J.-F. (2016). The Mont-Blanc prototype: An Alternative Approach for HPC Systems.

Rajovic, N., Rico, A., Puzovic, N., Adeniyi Jones, C., & Ramirez, A. (2014). Making the Case for an ARM-Based HPC System. *ELSEVIER*, 322-334.

Ramos, J. A. (23 de Abril de 2014). *https://www.adictosaltrabajo.com*. Recuperado el 15 de Enero de 2017, de https://www.adictosaltrabajo.com: https://www.adictosaltrabajo.com/tutoriales/pig-first-steps/

Raspberrypi. (9 de Septiembre de 2016). Obtenido de raspberrypi: https://www.raspberrypi.org/

Solano, H. L., & Álvarez, C. R. (2005). *Estadística descriptiva y distribuciones de probabilidad.* Barranquilla: Ediciones Uninorte.

Srisuruk, W., & Kaewkasi, C. (s.f.). Low-Power Big Data Cluster. *Low-Power Big Data Cluster*. Suranaree, Tailandia. Obtenido de https://indico.cern.ch/event/311156/contributions/1684547/attachments/595776/819978/aiyara_cluster.pdf

Valero, M., Rajovic, N., M. Carpenter, P., Gelado, I., Puzovic, N., & Ramirez, A. (2013, Noviembre 17-22). Supercomputing with Commodity CPUs: Are Mobile SoCs Ready for HPC? *2013 SC - International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 1-12. Denver,Co: IEEE. doi:10.1145/2503210.2503281

Wu, X., Zhu, X., Wu, G.-Q., & Ding, W. (2013). Data mining with big data. *IEEE Transactions on Knowledge and Data Engineering*, 97 - 107. doi:10.1109/TKDE.2013.109