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

Low Power Consumption Portable 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 support so-called intensive applications is increasing in many organizations, because a series of applications that require the intensive management of data and short response times have arisen, an example of such applications are those which use data mining. The hardware has evolved in an accelerated way to meet these needs of intensive computing. On the other hand, derived from the miniaturization of components and the introduction of embedded systems, there has arisen a new generation of small computers mounted on boards called SoC (System on Chip) (R., G., & M., 2013), the most significant example being Raspberry Pi (raspberrypi, 2016) that became an alternative of low cost and reduced energy consumption, to accomplish computing tasks. Due their popularity, manufacturers have put effort increasing their performance by building models with more features and overall performance, which makes it an alternative to consider for intensive data processing.

This document describes the construction of an HPC platform based on SoC cards, which allows the development and execution of data mining applications. The results obtained in the different tests of operation and fault tolerance show that this platform offers the performance and robustness necessary to be used in the development of applications of data mining oriented to the education and also for the teaching of this discipline, all of this with an investment that is possible to achieve in academic institutions and small organizations.

Key words: cluster, super-computing, parallel computing, 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

Introdução

Computação de alto desempenho (HPC) é o uso de processamento paralelo para executar aplicativos de forma eficiente, confiável e rápida (Garcia Nocetti, 2014). Os sistemas HPC de computação têm sido usados como ferramentas para o desenvolvimento e implementação de intensivo, tais como simulações em computador e cálculo de aplicações complexas operações, a solução envolveria tempo excessivo em equipamentos de computação convencional. O HPC é baseado no uso de equipamentos de computador equipado com redundância de hardware; Exemplos dessas equipes são supercomputadores de cluster, entre outros.

Um dos principais inconvenientes da HPC é o alto custo, classificadas em custo de aquisição e o custo de manutenção realizada durante o tempo de vida do equipamento. Esses dois fatores tornam o HPC em uma tecnologia reservada para organizações com fundo financeiro suficiente. Rajovic et al. (2014) divulgam que os sistemas de HPC estão presentes no processamento de grandes quantidades de dados. Estes sistemas têm, entre outras desvantagens, o elevado consumo de energia necessária para a operação e também para operar o sistema de arrefecimento. Consequentemente, as organizações que lidam com orçamentos menores, tais como pequenas universidades, possuindo queda fora de tais instalações, o que os coloca em desvantagem.

Mont Blanc (Valero et al., 2013) propõe um projeto voltado para o uso de tecnologias de eficiência energética para a alternativa HPC. Basicamente, a idéia é usar SoCs, que oferecem vantagens sobre equipamentos HPC tradicional para mitigar esses aspectos (Rajovic et al., 2014).

A necessidade de serviços prestados pela HPC continua a aumentar em todos os tipos de organizações. Um exemplo é a mineração de dados, o que requer equipes de alto desempenho para a operação devido à magnitude dos dados e processos que ele gera. o termo HPDA (Análise de Desempenho alta de Dados) está atualmente manipulados para se referir a computação de alto desempenho aplicado a mineração de dados.

Este projecto tem como objectivo criar uma plataforma de HPC econômico e redução do consumo de energia que é uma alternativa acessível para organizações com orçamento limitado, interessados em usar a mineração de dados para melhorar seus processos. O documento é desenvolvido de acordo com as seguintes peças: solução proposta, descrição de arquitetura, design, implementação, resultados e conclusões.

Solução proposta

A fim de implementar a nossa ideia, as seguintes características técnicas têm sido considerados para a concepção e implementação desta plataforma:

- **Baixo custo.** Um dos principais objectivos deste projecto é que o custo desta plataforma é reduzido e por isso é uma opção rentável para pequenas organizações ou instituições acadêmicas que normalmente lidam com orçamentos que impedem supercomputação compra de equipamentos caros.
- **Consumo de energia reduzido.** A plataforma deve ser dirigido para a poupança de energia, fornecendo serviços de alta performance, mas com consumo mínimo de energia. Para este fim, os componentes seleccionados para esta plataforma requerem 5 volts e 2,5 amps para executar, consumindo 12,5 watts por hora. Isto implica um baixo consumo de energia em comparação com um computador pessoal que varia de 300 a 600 watts por hora.
- **Instalações e espaço.** Nossa solução deve levar instalação física para um espaço pequeno, em comparação com supercomputadores que requerem grandes instalações para a operação. A plataforma será constituída por elementos com dimensões de 11 x 8 x 1,4 cm., Capaz de satisfazer este requisito estabelecido no início do projeto.
- **Poder de processamento.** A plataforma construída deve fornecer uma capacidade de computação suficiente para processar aplicações educacionais e fornecer suporte para treinamento em técnicas de mineração de dados. A fim de obter energia de processamento necessária, um aglomerado com 24 cartas que em conjunto proporcionam um rendimento teórico de 768 e um valor estimado de Gflops Linpack (Petitet A., 2016) de 614.40 Gflops e consumo de 300 W foi integrado / h.

Projeto de arquitetura

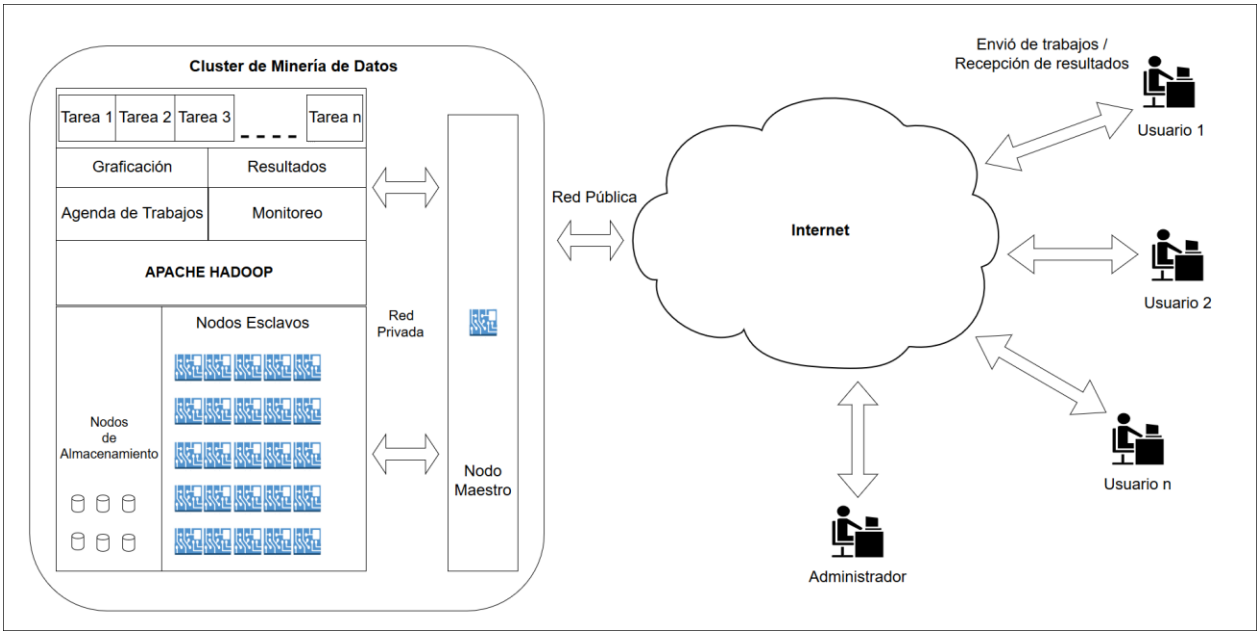
A plataforma é orientada para fornecer infra-estrutura de serviços funcional computação de alto desempenho com a finalidade específica de uso em mineração de dados aplicações. A Figura 1 mostra o modelo conceptual da plataforma mostrada.

A arquitectura é constituído por um conjunto de módulos funcionais, que são descritos abaixo:

Módulos de hardware

- **Mestre Node.** Este nó é o ponto de entrada para os usuários quando eles querem processar tarefas de cluster. Além disso, este nó atribui tarefas para outros nós, entrada de controlo e de saída de dados, e também fornece suporte para a função de monitorização de nós do cluster.
- **Nós escravos.** Executar tarefas de processamento designadas pelo nó mestre.
- Os nós de armazenamento. Conjunto de nós do cluster que têm um disco rígido para armazenar as tarefas e o resultado do processamento.
- **Privado interconexão de rede.** É a rede de dados que interliga os professores, escravos e nós de armazenamento.
- **Interconexão da rede pública.** Esta ligação permite o acesso ea utilização da arquitetura a partir de qualquer computador conectado à Internet.

Figura 1. Modelo Conceptual de la Plataforma.



Fuente propia: Verduzco et al., 2017

Módulos de software

- **Resultados.** Estes dados obtidos a partir do módulo de uma tarefa executada será acessível para o utilizador; esses conjuntos de dados podem ser vistos quando o usuário exige.
- **Gráficos.** Neste módulo os resultados são apresentados em formato gráfico que acompanhado facilitar a análise e interpretação.
- **Agenda de trabalho.** Este tarefas módulos apresentados pelos usuários calendarizarán.
- **Monitoramento.** Este módulo irá permitir monitorizar o estado operacional da plataforma e verificar o funcionamento de cada nó.
- **Apache Hadoop.** É o módulo principal, que fornece um ambiente que permite gerenciar o processamento e armazenamento de trabalhos distribuídos entre os nós.

Arquitetura Implementação

Em seguida, os detalhes técnicos da arquitetura e software utilizados são descritos.

Configuração de Cluster

Para a criação da plataforma 24 cubieboard de nós A80 e caminhão Plus (Tabela 1) foram utilizados modelos. Sistemas operacionais suportados para os cartões são: Android (Google, 2016) e Lubuntu (Lubuntu, 2016). Projeto Lubuntu para o sistema operacional foi instalado. O aglomerado foi configurado em arquitectura de mestre-escravo, que designa um único nó como mestre e os restantes nós como escravos.

Tabla 1. Tipos de SoC utilizadas en el proyecto.

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

Fuente propia: Verduzco et al., 2017

Instalando Apache Hadoop

Foi decidido instalar a plataforma Hadoop (Apache Hadoop, 2016), principalmente porque ele é amplamente utilizado tanto no meio acadêmico e as empresas para realizar a análise de dados, bem como a variedade de quadros que ele suporta. O método de instalação consistiu de um único nó configurado programas Apache Hadoop e Apache Pig (Apache porco, 2017), mais tarde para fazer repetições para os nós restantes. As Figuras 2 e 3 mostram a operação de agrupamento.

Figura 2. Imagen del clúster de minería de datos en operación.

Fuente propia: Verduzco et al., 2017

Figura 3. Vista lateral del clúster de minería de datos en operación.

Fuente propia: Verduzco et al., 2017

Cluster de armazenamento

O aglomerado necessita de armazenamento dedicado para acomodar tarefas e dados gerados. Para este fim, o conjunto de armazenamento constituído por quatro nós são configurados. Cada nó está instalado um disco rígido de 1 TB.

Figura 4. Imagen del Clúster de almacenamiento.

Fuente propia: Verduzco et al., 2017

Resultados

A fim de determinar a funcionalidade da plataforma, decidiu-se realizar testes diferentes classificados: operação, tolerância a falhas e stress.

Testes de condução ambiente

O ambiente no qual os testes foram realizados como se segue:

- Un switch LINKSYS de 28 puertos Gigabit Ethernet.
- Cinco tarjetas cuebieboard a80 octa-core a 2.0 Ghz, 2 GB RAM, 8 GB en memoria NAND.
- Un nodo de las cinco fue establecido como nodo maestro, el cual asignará tareas de procesamiento a los nodos esclavos.
- 19 tarjetas cubie truck plus octa-core a 2.0 Ghz, 2 GB RAM, 8 GB en memoria NAND.
- A cuatro de estas tarjetas se añadió una unidad de disco duro para manejar el almacenamiento de archivos de entrada y salida.
- Sistema operativo Lubuntu versión 14.04 Kernel 3.4.39

- Apache Hadoop versión 2.7.2
- Software Apache Pig versión 0.16

Teste de funcionamento

Para executar este teste foram seleccionados de um conjunto de dados resultante a partir de um estudo da qualidade do ar realizado em diferentes províncias de Espanha. Cada registro tem lugar no primeiro dia de cada mês durante o período de 1997 a 2013, totalizando 153,121 registros. Então, na Figura 5, o conjunto de instruções para o processamento desta tarefa e a Tabela 2 mostra os resultados obtidos.

Figura 5. Algoritmo utilizado en la ejecución de la prueba de operación

```
1. 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);
2. dump measure;
3. filter_measure = filter measure by date != 'dia';
4. measure_by_province = group filter_measure by province;
5. num_measures_by_province = foreach measure_by_province generate group,
   AVG(filter_measure.co) as measure;
6. DUMP num_measures_by_province.
```

Fuente: Ramos, 2014

Tabla 2. Resultados prueba de operación.

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

Fuente propia: Verduzco et al., 2017

Falha teste de tolerância

O objetivo deste ensaio é medir o impacto dos nós do cluster falha para ser a execução de um algoritmo. Para ele levou em conta a funcionalidade oferecida pelo sistema de arquivos que trabalha com Hadoop, que permite definir o número de repetições de segmentos distribuídos nós do cluster de dados. Para este teste, foram consideradas as variáveis: número de nós com deficiência e replicação do índice, a fim de determinar o grau de sucesso do consumo tarefa e memória causada pela replicação índice selecionado é concluída. A tarefa seleccionada foi executado dez vezes, a fim de obter a tolerância a falhas indicadores mencionado acima. A Tabela 3 mostra os resultados obtidos.

Tabla 3. Resultados de la prueba de toleracia a fallos.

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

Fuente propia: Verduzco et al., 2017

Além disso, na Tabela 3 é visível quando a configuração da plataforma com uma taxa de replicação igual a um e dois nós de fora de operação, gera uma taxa de sucesso de 10% após a conclusão da tarefa, o que indica que o índice de configuração replicação não é óptima para a plataforma. Com a configuração de replicação em seis de oito nodos sucesso fora 40%, o qual é um candidato para a configuração óptima foi obtida. Com a taxa de replicação de nodos 23 e 18 fora de sucesso de 90% é obtido, o qual garante que a tarefa é desenvolvido, se, pelo menos, seis nós de trabalho.

A desvantagem de configurações de replicação índice é o consumo de armazenamento redundante causada por várias cópias de dados residentes nos nós.

Stress Test

Este teste destina-se a medir o tempo de processamento de mineração de dados do agrupamento. Para este fim, levou uma coleção de dados de 200 mil, 400 mil, 600 mil, 800 mil e 1 milhão de registros, com o objectivo de estabelecer os tempos de execução com base em recolhas de dados tendência numéricos gerados dados acima mencionados nos seguintes distribuições numérica, normal, de Bernoulli, qui-quadrado, hipergeométrico Laplace, lognormal, Poisson, uniforme (Solano e Alvarez, 2005) de distribuição. O algoritmo foi aplicado a cada grupo de registos é o processamento de leitura. Em seguida, as instruções utilizadas são as seguintes:

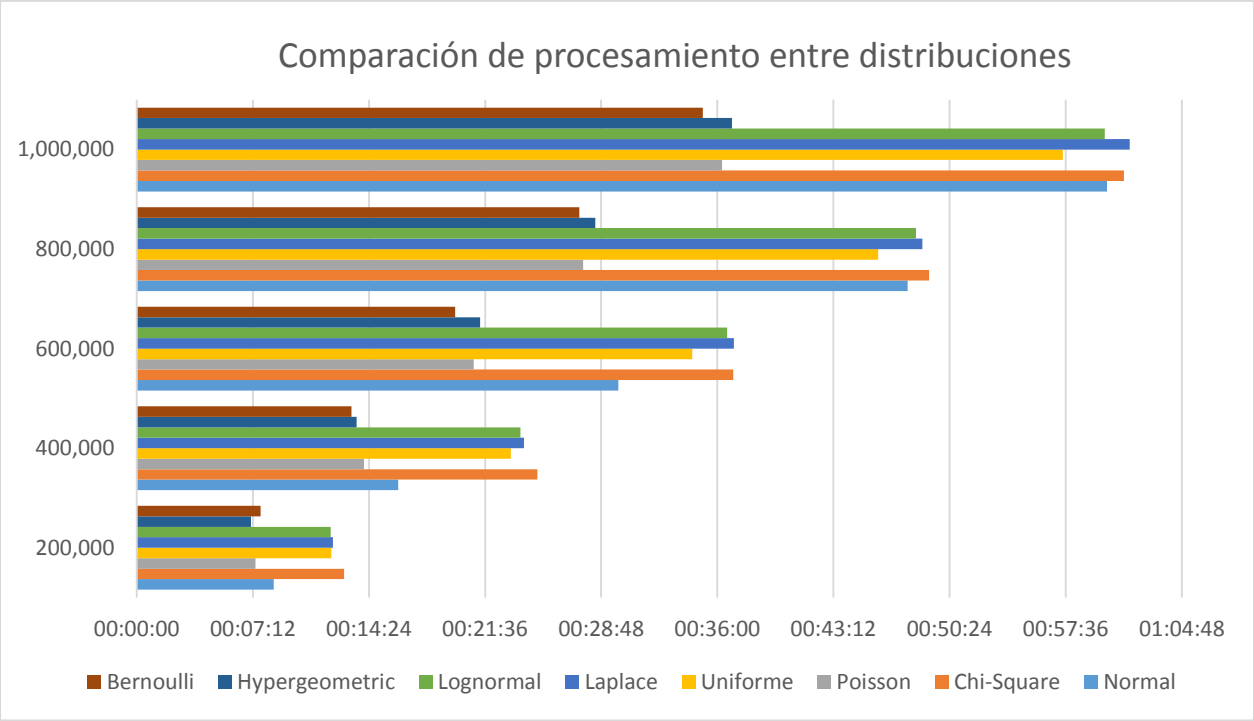
Figura 6. Instrucciones ejecutadas para la prueba de estrés.

```
1. 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);
2. STORE numeros INTO 'pig_output_numeros1millonbernoulli' USING PigStorage('\t');
```

Fuente propia: Verduzco et al., 2017

Cada recolha de dados foi processada três vezes e obteve-se o tempo médio necessário para completar a operação. A Figura 6 mostra os resultados obtidos.

Figura 7. Comparación de procesamiento entre distribuciones.



Fuente propia: Verduzco et al., 2017

Como mostrado na Figura 7, os tempos de execução variam dependendo do número de distribuição a partir da qual os dados. Distribuições mais de tempo de execução são qui-quadrado e Laplace.

Conclusões

Este documento descreve a implementação de uma plataforma de mineração de dados orientada para o desenvolvimento e aplicação de execução mostrado. A coisa notável sobre esta plataforma é que ela consiste de tecnologia SoC, o que implica um custo reduzido. Outros aspectos a destacar são o pequeno espaço que ocupam suas instalações e baixo consumo de energia associado com o seu funcionamento. Os vários testes realizados permitem afirmar que esta plataforma fornece os algoritmos necessários para executar a educação universitária orientada, pesquisa e funcionalidade treinamento. Todos estes aspectos fazem esta plataforma uma alternativa atraente para as instituições com baixo orçamento que querem ter sistemas HPC.

O trabalho futuro será destinada a melhorar a usabilidade da plataforma configurando diferentes quadros especializados em mineração de dados.

Bibliografía

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