

Análisis del rendimiento en Clústeres basados en PelicanHPC

Analysis of performance in PelicanHPC-based Clusters

John A. Aguirre-Carrasco^A, Luis M. Alfaro-Chirinos^B y Alejandrina Huaylla-Quispe^C

ORCID:0000-0002-1522-2963^A, ORCID:0000-0002-0475-6994^B y ORCID: 0000-0002-0258-8632

C

(Recepción: 15/04/2023 y aceptación 05/10/2023)

Resumen— La implementación de clústeres de alto rendimiento tienen aplicación en entornos de agrupamiento de nodos físicos independientes y de manera virtualizada mediante un middleware especializado o un Sistema Operativo dedicado, con el objetivo de obtener mayor recurso computacional a partir de equipos básicos. Ante ambas opciones, surge la pregunta: ¿Qué implementación arroja mejores resultados en cuanto al rendimiento?, por lo que el presente estudio tiene por finalidad realizar un análisis al rendimiento de clústeres en dos escenarios: diez equipos de cómputo homogéneos interconectados a una red local por cable trenzado CAT 6a y un switch; y de otra parte, diez máquinas virtualizadas en el software VirtualBox, con recursos repartidos equitativamente como son: memoria RAM, procesadores y disco. La metodología adoptada fue experimental en cinco ciclos repetitivos y trabajando en función a las medias del tiempo de ejecución. El algoritmo de prueba es un contador de números primos codificado en lenguaje C++, generando su archivo de salida y posterior ejecución a nivel de consola sobre la terminal en el Sistema Operativo PelicanHPC (variante de CD live del SO Debian). Los resultados son uniformes inicialmente, trabajando de 1 a 10 nodos, donde el resultado demuestra más de 30% de mejora en el clúster tradicional frente a un progreso irregular del clúster virtualizado, puntos que serán objeto del análisis descriptivo en base a resultados obtenidos.

Palabras clave: clúster de alto rendimiento, paralelismo, virtualización, PelicanHPC.

Abstract— The implementation of high-performance clusters has application in environments of grouping of independent physical nodes and in a virtualized manner through specialized middleware or a dedicated Operating System, with the aim of obtaining greater computational resources from basic equipment. Given both options, the question arises: Which implementation yields better results in terms of performance? Therefore, the purpose of this study is to carry out an analysis of the performance of clusters in two scenarios: ten homogeneous computing devices interconnected to a local network, by braided CAT 6a cable and a switch; and on the other hand, ten virtualized machines in the VirtualBox software, with resources distributed equally such as: RAM memory, processors and disk. The methodology adopted was experimental in five repetitive cycles and working based on the average execution time. The test algorithm is a prime number counter coded in C++ language, generating its output file and subsequent execution at the console level on the terminal, on the PelicanHPC Operating System (live CD variant of the Debian OS). The results are initially uniform, working from 1 to 10 nodes, where the result demonstrates more than 30% improvement in the traditional cluster compared to irregular progress in the virtualized cluster, points that will be the subject of descriptive analysis based on the results obtained.

Keywords: high performance cluster, parallelism, virtualization, PelicanHPC.

A. John A. Aguirre Carrasco Universidad Nacional Micaela Bastidas de Apurímac, jaguirre@unamba.edu.pe

B. Luis Miguel Alfaro Chirinos, Universidad Nacional Micaela Bastidas de Apurímac, lalfaroch@unamba.edu.pe

C. Alejandrina Huaylla Quispe, Universidad Nacional Micaela Bastidas de Apurímac, ahuaylla@unamba.edu.pe

1 INTRODUCCIÓN

El uso de los clústeres nace a partir de la década de los 60, donde se proyecta la conexión de computadores en una primeriza ARPANET (Advanced Research Projects Agency Network) o en español Red de la Agencia de Proyectos de Investigación Avanzada, toman como solución el agrupamiento de ordenadores funcionales independientes para componer una supercomputadora que disponga de altos recursos computacionales (tales como procesadores, memorias principales, disco duro, etc.). [1]- [2]

Gracias al avance de la tecnología en todo el campo de la informática, mejora también el tema de redes y

comunicaciones, que permite reforzar el vínculo computador - clúster [2], [3], [4], por la naturaleza de conectividad hacia una Red de Área Local (LAN) y la implementación de protocolos de Internet tales como el TCP/IP [1].

La necesidad y las nuevas tecnologías permiten asimilar también el tema de la programación concurrente y distribuida, más específicamente el paralelismo, que consiste en enfocar el trabajo de más de un procesador en una sola tarea específica, contando con los recursos del ordenador [3], [5]. Entonces las bases del clúster computacional toman una nueva perspectiva operacional, clasificada más adelante en las definiciones de Clústeres de Alto Rendimiento, Clústeres

de alta disponibilidad y clústeres de alta eficiencia [6].

Los clústeres de alta eficiencia o HPCC (High Performance Computing Clúster) son el enfoque de este trabajo de investigación [7], [5]. Mas específicamente en los resultados que arrojan dos implementaciones desarrolladas hoy en día: la tradicional con ordenadores independientes en unidades funcionales y maquinas virtualizadas, que es el modo tendencia de trabajo hoy por hoy en los data center de las diferentes organizaciones [8].

La motivación que nos lleva a realizar este trabajo es alcanzar un análisis del rendimiento en dos escenarios de implementación: una tradicional y otra mediante un software que virtualiza equipos. Con este análisis se pone a disposición de los investigadores dos panoramas claros de cómo el rendimiento puede variar dependiendo de los recursos con los que cuenta o podría contar para implementar un clúster, utilizando para ello el proceso deductivo - inductivo del método científico [9]. A su vez que según el enfoque de la solución nos lleva a nuevas preguntas como: ¿es mejor o peor implementar un clúster virtualizado?, ¿cuál es el máximo rendimiento obtenido en un clúster tradicional?, ¿existe la implementación mixta (tradicional y virtualizada) en clústeres y cómo es el análisis de su rendimiento?, etc., preguntas que nacen a partir del análisis de rendimiento de clústeres.

Los aportes de esta investigación se encuentran en el análisis experimental singular en cada implementación propuesta, para después contrastar ambos resultados estadísticos; de tal manera que se pueden observar patrones con una tendencia a la baja o patrones irregulares anticipando a los resultados. Para lo cual en el punto dos, vemos los métodos y procedimientos utilizados a lo largo de la experimentación, en el punto tres tenemos los resultados obtenidos de manera experimental, en el punto cuatro las discusiones a partir de investigaciones anteriores, en el punto cinco las conclusiones del grupo de investigación y finalmente las referencias que son una parte esencial de todo trabajo de investigación científico.

2 MÉTODOS Y PROCEDIMIENTOS

2.1 Clúster de alto rendimiento implementado mediante virtualización

La virtualización de Sistemas Operativos trabaja sobre un software que simulación bajo arquitecturas x86/amd64 llamado Oracle VM VirtualBox en su versión 7.0.4, donde es posible implementar la instalación de sistemas virtualizados llamados "sistemas invitados", que se apoyan física y lógicamente sobre un computador, denominado "anfitrión" [5], [10], [11].

Se debe señalar que los data center actuales trabajan de esta manera para lograr un trabajo distribuido y compartido debido al alto coste de los equipos, con la indicación que el trabajo sobre licencia de operación resulta conveniente por las

funciones completas del virtualizador [10], [12].

Como componente middleware y Sistema Operativo que se utilizó para la virtualización se tiene a PelicanHPC en su versión GNOME 4.1; que resulta una distribución de otro SO libre: Debian [13]. Este middleware toma las siguientes características:

TABLA 1
Características de maquina virtualizada sobre PelicanHPC

Componente	Características
Procesador	1 procesador 2.9GHz
Memoria principal	2GB
Disco	10 GB

Evaluación experimental para diez máquinas virtuales

El equipo anfitrión es un equipo portátil Dell Alienware R4 17, con un procesador I7 de séptima generación de 8 núcleos a 2.9GHz, memoria principal de 32GB DDR4 y un disco de 1TB SSD M.2 NvME PCIe. El Sistema Operativo del computador anfitrión es Windows 10 Home con una arquitectura de 64 bits [14].

Características que son suficientes para soportar la virtualización de diez máquinas virtuales, las cuales trabajaron en una red interna, trabajando sobre el adaptador PCnet -FAST III (Am79C973).

2.2 Clúster de alto rendimiento implementado de forma tradicional

Es el modo de trabajo primigenio de implementación, desde la década de los 60 y de los cuales se tiene resultados positivos [12]. El trabajo empírico trabaja en base a una red en un entorno punto a punto sobre el protocolo TCP/IP, por lo que se tomará el uso de un switch y el trabajo de envío y recepción de paquetes mediante la dirección MAC de la tarjeta de red del equipo. Las características del equipo son las siguientes:

TABLA 2
Características de los equipos de cómputo

Componente	Características
Procesador	I5 4GEN 3.2GHz 4 NÚCLEOS
Memoria Principal	8GB DDR3 1333MHz
Almacenamiento	512GB SSD SATA
Sistema Operativo	Windows 10 Pro 64 bits

Evaluación experimental para diez equipos de cómputo de escritorio

Sobre el Switch para la conectividad local (red LAN) se tiene un equipo en la marca D-Link DES-1024D de 10/100Mbps de transmisión y 24 puertos y diez patch cord armados con cable UTP Cat6A, RJ45 Cat6A SATRA crimpados de acuerdo al estándar de cableado T-568B [15], con dimensiones desde 1 metro a 4 metros. Se indica que los patch cord fueron

testeados y su funcionamiento estuvo correcto hasta la obtención de resultados.

2.3 Sobre la aplicación

Se tomó por casos prácticos el algoritmo de búsqueda de números primos debido al orden de su complejidad $O(n^2)$, que nos permite obtener los resultados en tiempos prudentes [16].

TABLA 3

Búsqueda de números primos con límite superior en base a exponentes del número dos desde 2^0 hasta 2^{17}

Lim	Números primos	Conteo
2^0	---	0
2^1	2	1
2^2	2,3	2
2^3	2,3,5,7	4
2^4	2,3,5,7,11,13	6
2^5	2,3,5,7,11,13,17,19,23,29,31	11
2^6	2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61	18

Evaluación práctica en papel

Donde su programación plasmada en un algoritmo en el lenguaje C++ (IDE Zinjal) resulta ser sencillo de la siguiente manera:

```
#include <iostream>
#include <iomanip>
#include <time.h>
#include <unistd.h>
#include <math.h>
using namespace std;
bool primo(int n){
    bool rpta=false;
    int cont=0;
    for(int i=1;i<=n;i++){
        if(n%i==0){
            cont++;
        }
    }
    if(cont==2){
        rpta=true;
    }return rpta;
}
int Numprimos(int m){
    int contador=0;
    for(int j=2;j<=m;j++){
        if(primo(j)){
            contador++;
        }
    }
    return contador;
}
int main(int argc, char *argv[]) {
    int k=pow(2,0);
    cout<<setw(20)<<"N";
```

```
cout<<setw(20)<<"S";
cout<<setw(20)<<"TIEMPO"<<endl;
for(long p=k;p<=131072;p=p*2){
    cout<<setw(20)<<p;
    double tiempoinvertido=0.0;
    clock_t inicio=clock();
    cout<<setw(20)<<Numprimos(p);
    clock_t fin=clock();
    tiempoinvertido+=(double)(fin-
    inicio)/CLOCKS_PER_SEC;
    cout<<setw(20)<<tiempoinvertido<<endl;
}
return 0;
}
```

2.4 Booteo de PelicanHPC

Se contó con dos contextos: el virtualizado y el tradicional, en caso del contexto virtualizado, nos limitamos a vincular el origen al archivo ISO del SO PelicanHPC: "*pelicanhpc-v4.1-gnome.iso*".

Para el contexto tradicional se booteo el archivo ISO de PelicanHPC "*pelicanhpc-v4.1-gnome.iso*", mediante el ejecutable Rufus4-1.exe, a un esquema de partición MBR, sobre diez memorias pendrive de 4GB de capacidad. Se debe indicar que el riesgo operacional de los equipos no se ve comprometido debido a la naturaleza LIVE del SO, sabiendo que las modificaciones dentro del SO LIVE no se mantienen al reiniciar el equipo [17].

2.5 Ejecución sobre consola

Para la ejecución se toma como referencia al archivo C++ del algoritmo de búsqueda de números primos, ubicada en la carpeta *Documents*. Para lo cual procede la generación del archivo de salida, con el cual se ejecutará el algoritmo. Procedemos a primeramente trabajar en modo super usuario, mediante el código: "*user@pell:~/Documents\$ sudo su*", a continuación se procedió a la creación del archivo de salida (el que necesitamos para la ejecución) mediante el comando: "*root@pell:/home/user/Documents# g++ *nombre_archivo.cpp* -o *nombre_salida.out**". Concluido este paso, procedemos a la ejecución del archivo de salida, mediante el comando: "*root@pell:/home/user/Documents# mpirun -np 1 ./*nombre_salida.out**", para a continuación tener el siguiente resultado:

N	S	TIEMPO
1	0	1e-06
2	1	1e-06
4	2	1e-06
8	4	1e-06
16	6	2e-06
32	11	4e-06
64	18	1.2e-05
128	31	3.9e-05
256	54	0.000149
512	97	0.000493
1024	172	0.001895
2048	309	0.007481
4096	564	0.029209
8192	1028	0.115536
16384	1900	0.462548
32768	3512	1.83687
65536	6542	7.3272
131072	12251	29.3093

Fig. 1. Resultados de ejecución del archivo de salida, correspondiente al algoritmo de búsqueda de números primos, donde la columna N corresponde al límite en base a exponentes en base dos, que va desde 2^0 hasta 2^{17} . La columna S corresponde a los números primos que se encontraron en dicho intervalo y la columna TIEMPO cuantifica el tiempo en unidad de segundos.

Se hace la indicación que el procedimiento explicado se replicó trabajando desde un nodo (maestro), incrementándose secuencialmente hasta los diez nodos (un nodo maestro + nueve nodos esclavos); en los dos contextos inicialmente presentados: forma virtualizada y forma tradicional.

Además de obtener los resultados, se obtuvo la media de cinco pasadas en cada caso de evaluación para obtener resultados significativos, mostrados a continuación en la sección resultados.

3 RESULTADOS

Los resultados se muestran de manera independiente, en los dos contextos, primeramente, en el entorno virtualizado; donde se tomaron tres casos interpolados, mostrando los resultados en un diagrama lineal. Pasamos a mostrar los resultados experimentales.

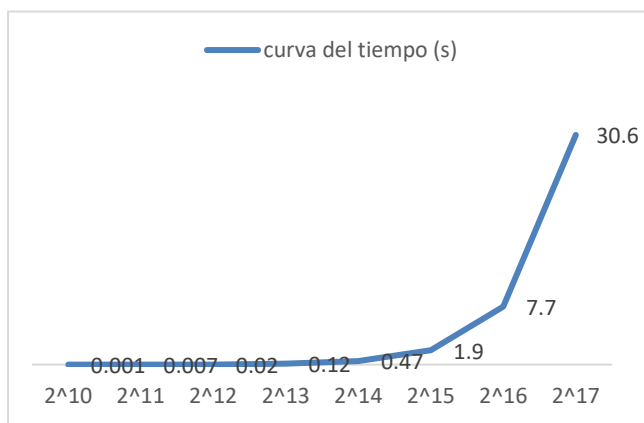


Fig. 2. Análisis de rendimiento en máquinas virtualizadas a un nodo, tomados desde el valor 2^{10} al 2^{17} , debido a la fácil lectura.

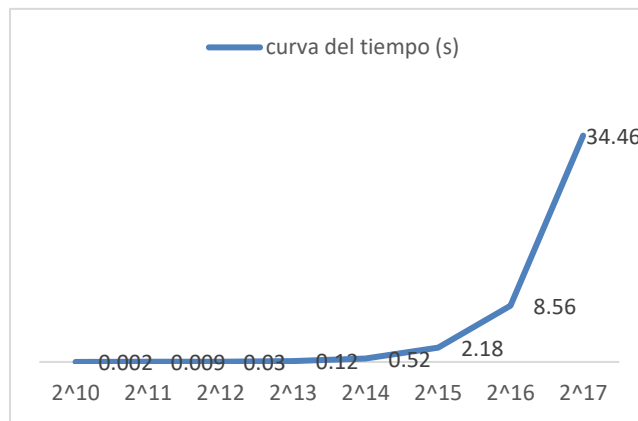


Fig. 3. Análisis de rendimiento en máquinas virtualizadas a cinco nodos, tomados desde el valor 2^{10} al 2^{17} , donde se aprecia una ligera mejora frente a un nodo (Fig. 1).

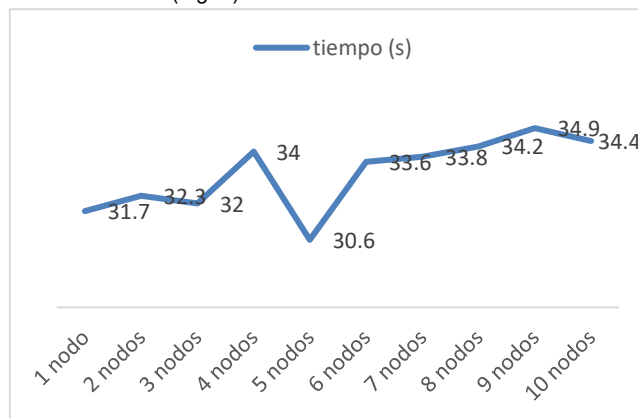


Fig. 4. Análisis de rendimiento en máquinas virtualizadas a diez nodos, tomados desde el valor 2^{10} al 2^{17} , donde se aprecia una caída en el rendimiento comparado a la Fig. 2 y Fig. 3

Una evaluación en la búsqueda de números primos en el límite superior ($2^{17}=131072$) da un resultado notable, a la par que representan valores en la media de cinco evaluaciones, que pueden visualizarse en el siguiente gráfico.

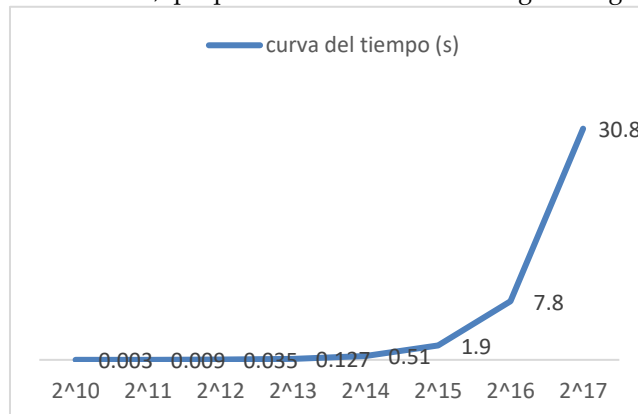


Fig. 5. Análisis de rendimiento en máquinas virtualizadas desde un nodo hasta los diez nodos, tomando el mayor de los valores (2^{17}), donde se aprecia un desarrollo irregular

El contexto que se verá ahora pertenece a la forma tradicional, donde analógicamente veremos datos interpolados a un nodo, cinco nodos y diez nodos, como se muestran a continuación.

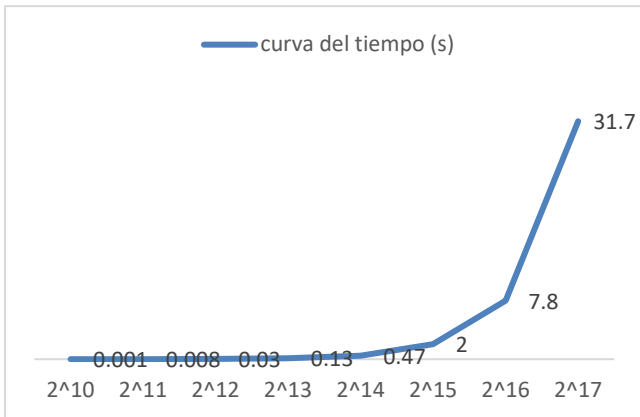


Fig. 6. Análisis de rendimiento en el clúster tradicional a un nodo, tomados de similar manera, desde el valor 2¹⁰ al 2¹⁷,

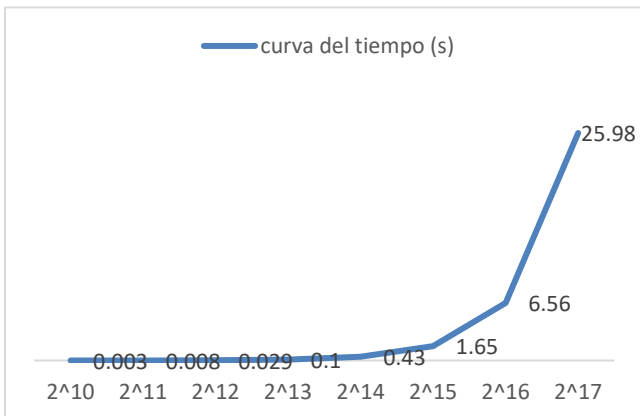


Fig. 7. Análisis de rendimiento en el clúster tradicional a cinco nodos, tomados del intervalo desde 2¹⁰ al 2¹⁷, Notando una mejora frente a los resultados mostrados en la Fig. 6.

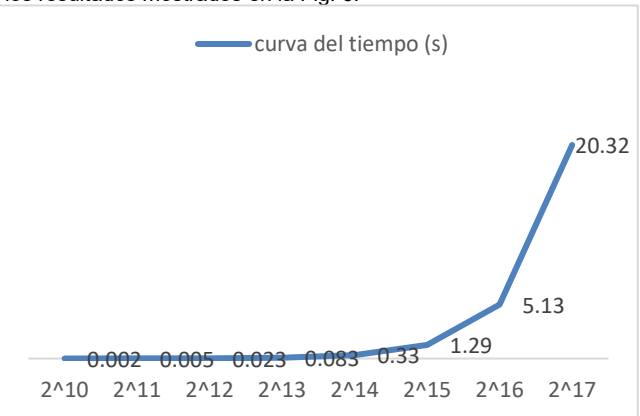


Fig. 8. Análisis de rendimiento en el clúster tradicional a diez nodos, tomados del intervalo desde 2¹⁰ al 2¹⁷, Notando una mejora frente a los resultados mostrados en la Fig. 6 y Fig. 7

La comparativa a lo largo de los diez nodos, trabajándose de uno a uno frente al mayor de los valores de búsqueda, se plasma en la siguiente figura:

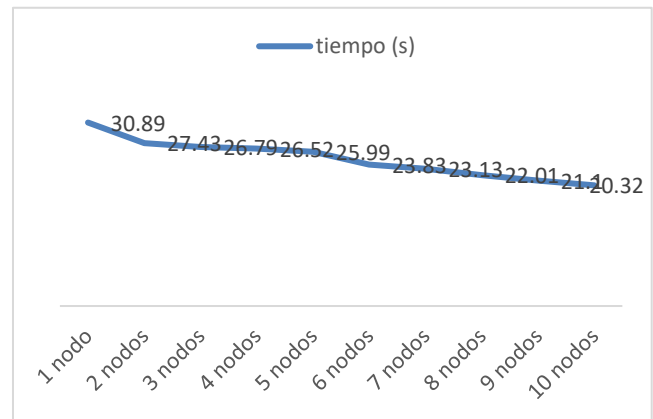


Fig. 9. Análisis de rendimiento en el clúster tradicional desde un nodo hasta los diez nodos con respecto al mayor de los valores de búsqueda, denotando un decremento lógico correcto durante el incremento de nodos computacionales.

4 DISCUSIONES

Se discute la diferencia de equipos, debido al carácter heterogéneo entre ambos contextos (clúster virtualizado y clúster tradicional) por lo que una comparación objetiva y empírica se considerará al comparar en un entorno homogéneo donde las capacidades tanto de hardware y software se equiparan de manera imparcial.

No obstante, se debe rescatar los resultados macro, donde podemos ver una atrofia computacional cuando se sobrecarga el componente de hardware al virtualizar maquinas sobre una única maquina anfitrión, aunque los recursos sean relativamente “potentes”, tarde o temprano será evidente la caída operacional-computacional en bloque, resumen que se puede sustentar en la Fig. 5 donde no existe una conclusión clara, más bien una proyección no lógica donde el tiempo se incrementa mientras se incrementa los nodos virtuales (cuando la proporción entre ambas partes debería ser inversa, es decir a más nodos, menor tiempo de ejecución). De otra parte, en el contexto tradicional, se logra apreciar una mejora del 34% de un nodo a diez nodos (el tiempo pasa de 30.89s a 20.32s), siendo el camino más recomendable para implementar clústeres de alto rendimiento.

Los resultados de la investigación convergen con en trabajo de Miguel G Xavier et. al. [5], [18]; puntualizando que el rendimiento nativo de los clústeres (implantación tradicional) tiene mejores resultados que el entono virtualizado. Una desventaja clara en la implementación de clústeres de computadoras es la parte energética, puesto que al tener varios nodos funcionales; también el consumo energético y la facturación por la misma serán directamente proporcionales, por lo que implementar sistemas energéticos como por ejemplo energías fotovoltaicas o eólicas pueden compensar su costo-operatividad, aunque el coste inicial en la implementación de estos últimos no resulta barato pero que a la larga se obtiene buenos resultados coste - beneficio.

5 CONCLUSIONES

Se concluye que la manera de obtener rendimiento computacional al implementar clústeres computacionales de alto rendimiento (HPCC); es realizando una conexión de área local o una red LAN donde se puedan conectar físicamente los nodos, apoyado sobre una red de alta velocidad que soporte un buen ancho de banda (en Gbps), aportando mayor performance al cometido. Además, que el manejo actual de servidores en data center, viendo la caída que manifiestan al sobrecargarse de máquinas virtuales [19]; pueda hacerse un balanceo equitativo (de similar designación de recursos), de tal manera que no se llegue a saturar el performance y el rendimiento. Por otra parte, buscar financiamiento para la adquisición de estos equipos no es nada fácil en una institución gubernamental del estado peruano.

REFERENCIAS

- [1] W. Chuliang, L. Minglu, Zhigang Wang and L. Xinda, "Automatic Performance Tuning for the Virtualized Cluster System," 2009.
- [2] A. Y. J, J. P. Walters and S. P. Crago, "Supporting High Performance Molecular Dynamics in virtualized clusters using IOMMU, SR-IOV and GPUDirect," 2015.
- [3] J. Xunfei, B. Tuguldur and N. Lam, "High performance cluster monitoring system," *Conference: 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018.
- [4] Z. Wu, C. Zhangxin, C. C. Douglas and T. Weigin, "High Performance Computing and Applications," *Second international conference HPCA 2009 Shanghai, China*, 2010. <https://doi.org/10.1007/978-3-642-11842-5>
- [5] M. G. Xavier, M. V. Neves, F. D. Rossi, T. C. Ferreto, T. Lange and C. A. F. De Rose, "Performance Evaluation of Container-based Virtualization for high performance computing environments," 2013. <https://doi.org/10.1109/PDP.2013.41>
- [6] T. Thandar, C. Sung-Do and S. P. Jong, "Availability Modeling and Analysis on Virtualized Clustering with rejuvenation," *international journal of computer science and network security*, vol. 8, no. 9, 2009.
- [7] R. M. Yoo, C. J. Hughes, L. Konrad and R. Rajwar, "Performance Analysis of High Performance Computing Applications on the Amazon Web Services Cloud," 2013.
- [8] C. Paidamoyo, L. Chunyuan, M. Nikhil, C. Lawrence and R. Henao, "Survival Cluster Analysis," 2020.
- [9] R. Hernández, C. Fernández and P. Baptista, *Metodología de la Investigación*, Mexico: McGraw-Hill, 2014.
- [10] U. Deshpande, B. Wang, S. Haque, M. Hines and K. Golapan, "MemX: Virtualization of Cluster-wide Memory," 2010. <https://doi.org/10.1109/ICPP.2010.74>
- [11] D. Hasenkamp, A. Sim, M. Wehner and K. Wu, "Finding Tropical Cyclones on a Cloud Computing Cluster: using parallel virtualization for large-scale climate simulation analysis," 2010. <https://doi.org/10.1109/CloudCom.2010.68>
- [12] B. Tierney, W. Johnston, B. Crowley, G. Hoo, C. Brooks and D. Gunter, "The NetLogger Methodology for High Performance Distributed Systems Performance Analysis," 1998. <https://doi.org/10.2172/764331>
- [13] DistroWatch, "https://distrowatch.com/," [Online]. Available: <https://distrowatch.com/table.php?distribution=pelicanhpc>. [Accessed 20 septiembre 2023].
- [14] DELL Technologies, "https://www.dell.com," [Online]. Available: <https://www.dell.com/support/manuals/es-pe/alienware-17-laptop/alienware-17-r4-setupandspecifications/especificaciones?guid=guid-1eae1626-6999-4d20-a522-a574d6b4761c&lang=es-mx>. [Accessed 18 septiembre 2023].

- [15] FS community, "https://community.fs.com," [Online]. Available: <https://community.fs.com/es/blog/t568a-vs-t568b-difference-between-straight-through-and-crossover-cable.html>. [Accessed 20 septiembre 2023].
- [16] M. C. Cowgill, R. J. Harvey and I. T. Watson, "A Genetic Algorithm Approach to cluster Analysis," 1999. [https://doi.org/10.1016/S0898-1221\(99\)00090-5](https://doi.org/10.1016/S0898-1221(99)00090-5)
- [17] Keepcoding TechSchool, "https://keepcoding.io," [Online]. Available: <https://keepcoding.io/blog/que-es-live-cd/>. [Accessed 20 septiembre 2023].
- [18] M. G. Xavier, M. V. Neves and C. A. F. De Rose, "A Performance Comparison of Container-based Virtualization Systems for Mapreduce Clusters," 2014. <https://doi.org/10.1109/PDP.2014.78>
- [19] A. J. Younge, R. Herschel, J. T. Brown, G. V. Laszewski, J. Qiu and G. C. Fox, "Analysis of Virtualization Technologies for High Performance Computing Environments," 2011. <https://doi.org/10.1109/CLOUD.2011.29>

BIOGRAFÍAS

John Abraham Aguirre Carrasco, Ingeniero Informático y Sistemas egresado de la Universidad Nacional Micaela Bastidas de Apurímac.

Luis Miguel Alfaro Chirinos, Ingeniero Informático y Sistemas, maestro en Gestión Pública de la Universidad Cesar Vallejo, docente adscrito al departamento de Ingeniería, Escuela Académica Profesional de Ingeniera informática y Sistemas de la Universidad Nacional Micaela Bastidas de Apurímac.

Alejandrina Huaylla Quispe, Ingeniero de Sistemas e Informática, maestra en Gestión Pública de la Universidad Cesar Vallejo, Egresada de la maestría en Proyectos de Inversión de la Universidad Tecnológica de los Andes, docente contratado adscrito al departamento de Ingeniería, Escuela Académica Profesional de Ingeniera informática y Sistemas de la Universidad Nacional Micaela Bastidas de Apurímac.