Implementación de una plataforma de alto rendimiento usando PVFS2

Hugo Camacho, Santiago Gómez, Mirna Ponce y Georgina Castillo

H. Camacho, S. Gómez, M. Ponce y G. Castillo.
Universidad Politécnica de Altamira. Programa Académico de Ingeniería en Tecnologías de la Información, Carretera Tampico-Mante, Entronque con Libramiento Corredor Industrial km. 1.5, Altamira, Tams.
hugo.camacho@upalt.edu.mx

M. Ramos.,V.Aguilera.,(eds.). Ciencias Administrativas y Sociales, Handbook -©ECORFAN- Valle de Santiago, Guanajuato, 2013.

Abstract

Now a days derived from the services demand information and transfer data of any type, there is no doubt the importance of computer systems can operate uninterrupted and error free, 365 days a year.The existence Platform HP Calso known as cluster servers or just clusters play san important role in solving problems of science, engineering and the developmentand implementation of many commercial applications. The requirements of these applicationsare greater asthey grow,so itrequires an increaseinmemory andprocessor speeds. In order to improve performanceand make programming easier for these applications can be used parallel file systems. PVFS2 is aparallel file system developed by afree multi-institutional team of storage experts parallel networks.Building a cluster of computers offer teachers, researchers and students from the Polytechnic University of Altamira(*UPALT*) process large amounts of data ina faster way, believing significantly reduce the data access times.

11 Introducción

Los clúster son un conjunto de computadores completos conectados mediante una red disponible comercialmente (por ejemplo, una red LAN) que se comporta como un recurso de cómputo único (sólo un nodo o computador tiene acceso externo). En otras palabras, un clúster es un conjunto de varios ordenadores conectados a través de un conmutador (switch) y una red de alta velocidad de bajo coste (Ethernet), de tal forma que el conjunto es visto como un único ordenador, siendo más rápido y con mayor capacidad de almacenamiento que los comunes ordenadores de sobremesa. Existen otros tipos de conexiones para conectar un clúster además de Ethernet son: Myrinet e Infiniban. Para que el sistema de almacenamiento en el clúster presente también un coste efectivo se debería aprovechar el ancho de banda y la capacidad de los discos incluidos en todos los computadores del clúster en lugar de tener que añadir almacenamiento más caro en algunos nodos que actúen como servidores o tener que añadir arrays de discos y redes de área de almacenamiento (SAN-Store Area Network).

Los sistemas de ficheros distribuidos (o basados en NAS -Network Attached Store) permiten que las aplicaciones puedan acceder a un sistema de almacenamiento compuesto por los discos de los nodos de un clúster. Si es paralelo ofrecerá mejores prestaciones para aplicaciones paralelas con alta necesidad de E/S. Lo consiguen permitiendo que varios nodos (clientes) puedan acceder en paralelo a un mismo fichero y a múltiples ficheros. Usualmente los clúster son utilizados para incrementar la capacidad de almacenamiento, disponibilidad, tolerancia a fallos y rendimiento. Comparado de lo que se puede esperar de un solo ordenador fabricado para cubrir estas características, un clúster representa una solución más económica respecto al coste/beneficio. De un clúster se espera que ofrezca los siguientes servicios:

1. Alto rendimiento
2. Alta disponibilidad
3. Tolerancia a fallos
4. Escalabilidad

Este trabajo está organizado de la siguiente forma: la sección 2 se describe los aspectos relacionados con el diseño de un cluster de computadores. La sección 3 muestra los aspectos relevantes de la segunda versión de PVFS y la sección 4 discute la construcción del cluster. Finalmente la sección 5 presenta algunos resultados y la sección 6 plantea las conclusiones y trabajos futuros.

11.1 Diseño de una plataforma de alto rendimiento

La construcción de un clúster es relativamente sencilla y económica. Ya que tiene una gran flexibilidad para trabajar con distinto hardware y sistema operativo (clúster heterogéneo), siendo esta la opción más económica para su construcción (*véase la figura 1*). Sin embargo lo ideal es que todos los ordenadores tengan las mismas características en cuanto a hardware y sistema operativo (clúster homogéneo).

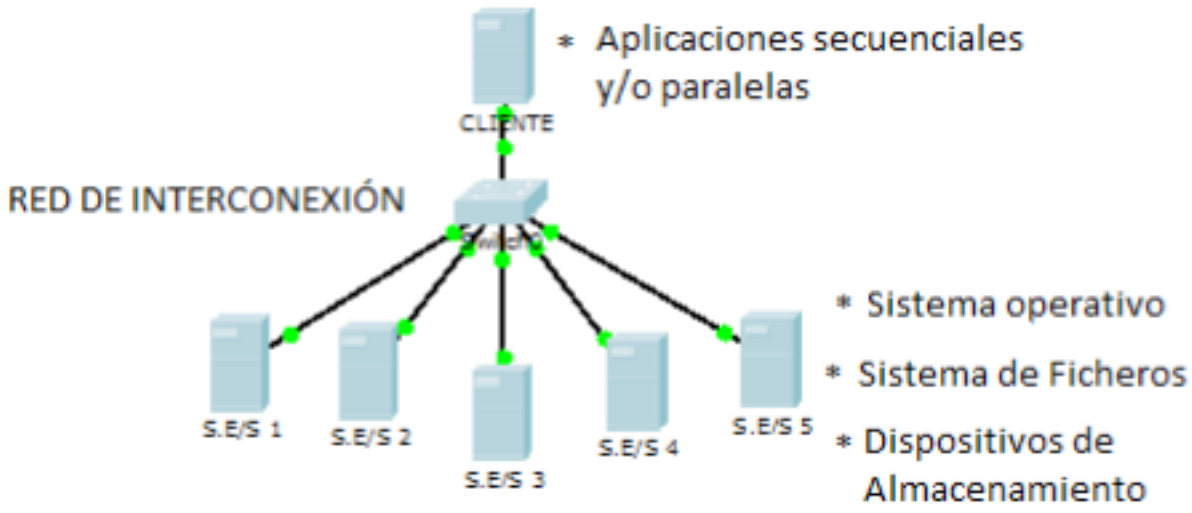
Para que un clúster sea operativo, no es suficiente con conectar entre sí los ordenadores. Es necesario proveer un sistema operativo, así como un sistema de ficheros que se encargue del manejo del clúster.

Ya que serán estos los que se encarguen de interactuar con el usuario y los procesos que corran en él para optimizar su funcionamiento.

Algunos sistemas operativos que pueden utilizarse para la configuración del clúster son: GNU/Linux (W. B. Ligon III, 1999) (Carns, Walter B. Ligon III , Robert B., & Rajeev Thakur , 2000), FreeBSD y algunos otros sistemas de paga como: Windows con su versión NT o Server, Mac OS X con xgrid o xserver.

Los sistemas de ficheros que pueden utilizarse por ejemplo: PVFS2 (PVFS2), OrangeFS (OrangeFS) y AbFS (Diaz, 2012).

Figura 11 Arquitectura de un Cluster de computadores



En (Anguita, 2011) un sistema de ficheros es un software que organiza y gestiona los datos almacenados en dispositivos de almacenamiento y los presenta a los usuarios o programas como ficheros o directorios (carpetas) lógicamente organizados en una jerarquía de directorios (estructura o árbol de directorios).

Los sistemas de fichero guardan información dentro de los dispositivos de almacenamiento; por ejemplo, discos duros, discos flexibles, CDs, etc; es decir, se encarga de obtener las propiedades físicas de los distintos dispositivos de almacenamiento proporcionando una interfaz única que es visible para los usuarios. Cada Sistema Operativo tiene un sistema de archivo montado por defecto; por ejemplo, NTFS en Windows, EXT2, EXT3 y actualmente EXT4 en Linux, UFS en Solaris o EFS en IRIX; los cuales son llamados sistemas de ficheros de discos o sistemas de ficheros locales. Sin embargo existen distintos tipos de sistemas de ficheros además de los mencionados en el párrafo anterior. Los sistemas de ficheros en red; estos sistemas de ficheros permiten compartir ficheros y directorios entre distintas maquinas que se encuentran conectadas a una red, de manera que se tiene la impresión de trabajar de forma local. Encontramos que este tipo de sistemas se clasifican en: sistemas de ficheros distribuidos y sistemas de ficheros paralelos. La diferencia principal en estos sistemas es que el segundo permite a los clientes acceder a los ficheros de forma paralela, reduciendo así el tiempo de acceso a los ficheros.

En esta investigación nosotros proponemos la implementación de un cluster de computadores para la universidad politécnica de Altamira utilizando el sistema de ficheros PVFS2 (Carns, Walter B. Ligon III , Robert B., & Rajeev Thakur , 2000), el cual es un sistema de ficheros paralelo gratuito para entornos GNU/Linux.

11.2 Arquitectura del sistema de ficheros pvfs2

PVFS2 (*Parallel Virtual File System*) es un sistema de ficheros paralelos desarrollado para trabajar en clusters de computadores. Ofrece almacenamiento y recuperación de datos para aplicaciones paralelas y seriales. Ha sido desarrollado a partir de la primera versión de éste mejorando la modularidad y añadiendo una fuerte integración con MPI-IO (MPICH).

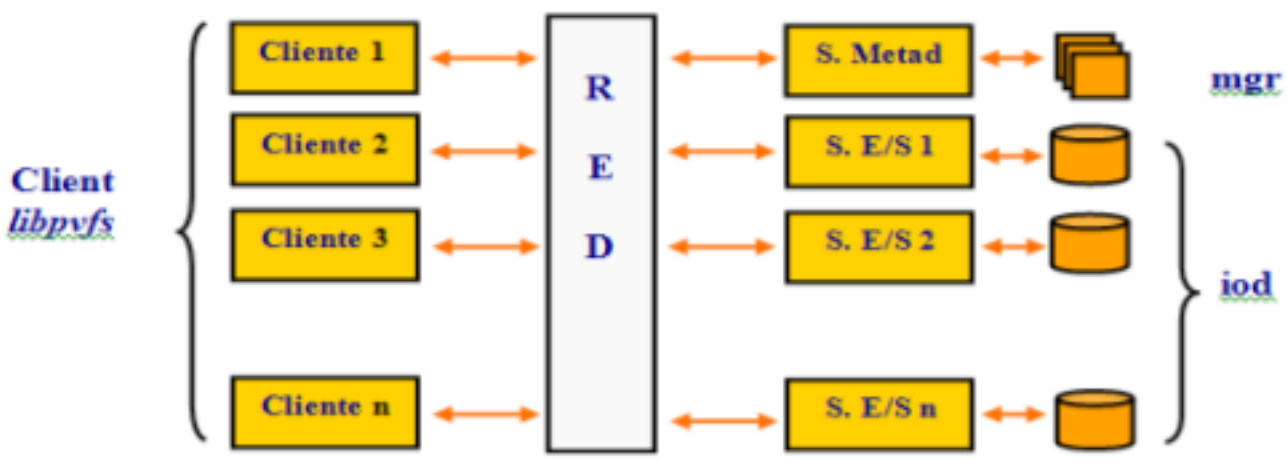
Estructura Hardware : PVFS2 está compuesto principalmente por tres elementos (véase figura 11.1):

- Servidor de Metadatos (mgr). El servidor de metadatos se encarga de mantener los atributos de los ficheros y directorios, tales como permisos, propietarios y localización de los datos. Algunas de las operaciones que se pueden realizar en un servidor de metadatos son: crear, eliminar, abrir o cerrar algún fichero. Esto se logra, mediante la comunicación del cliente con el servidor de metadatos a través de la librería libpvfs.

- Servidores de E/S (iod). Los servidores de E/S se encargan de almacenar y gestionar los accesos a los datos localizados en los directorios de PVFS.

- Clientes (client). A través de los clientes acceden los usuarios a los datos almacenados en los directorios de PVFS. Para poder hacerlo disponen de una librería llamada pvfs2lib. A su vez, también existe un módulo, insertado en el kernel de Linux, que ofrece un punto de acceso al sistema a través del sistema virtual de ficheros (VFS) de Linux. Adicionalmente se puede acceder al sistema a través de la librería de MPI-IO.

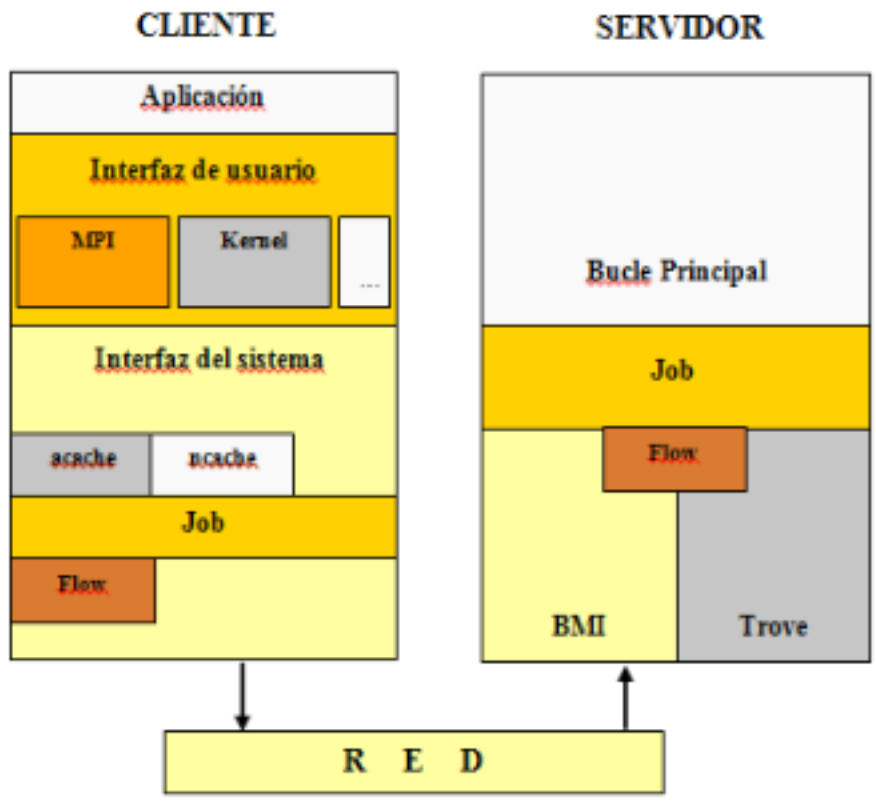
Figura 11.1 Componentes de PVFS2.



Arquitectura de Software: Los módulos que conforman la capa software de PVFS2 y su interacción se muestran en la figura 3. PVFS2 utiliza una semántica no bloqueante entre las distintas capas. El cliente se comunica con la interfaz del sistema PVFS2 a través de la capa de aplicación utilizando diferentes tipos de interfaces a nivel de usuario (Kernel-VFS, MPI-IO, etc.) La capa de interfaz de sistema es una API puesta a disposición de los clientes, que utiliza una serie de máquinas de estado para ejecutar las operaciones solicitadas por la capa superior. Estas máquinas de estado lanzan nuevas operaciones, ejecutadas y controladas por la capa JOB, cada una de las cuales tiene un número asociado para ser identificadas. Las operaciones ejecutadas por la capa JOB, se envían a los servidores a través de la capa FLOW, que se encarga de mover tanto operaciones como datos de los clientes a los servidores. La capa FLOW, a su vez, utiliza la capa BMI (Buffered Message Interface), para poder tener accesos a la red. BMI pone a disposición de la capa FLOW, distintos módulos que dan soporte a diversos tipos de red. Estos módulos hacen transparente al cliente el tipo que se está utilizando. Actualmente BMI soporta TCP (por defecto) [6], Infiniband, Gamma, Via y Myrinet. Las capas de software en el servidor son prácticamente las mismas que en los clientes, excepto que los servidores tienen una capa llamada TROVE, que es la encargada de almacenar los datos en los dispositivos de almacenamiento. Esta capa actúa tanto en los servidores de metadatos como en los servidores de datos. En los primeros, dando acceso a las distintas bases de datos donde se almacenan los metadatos.

En los segundos, dando acceso a las bases de datos que se utilizan para ubicar a un determinado datafile y también brindando acceso a los archivos tipo LINUX que almacenan los trozos de archivo del sistema de ficheros. En la Figura 3, se muestran tres partes importantes de la capa de software de PVFS2, que es diferente a la implementada en la primera versión. Por una parte la acache y la ncache, son dos caches que existen en el lado cliente, para almacenar de forma temporal los atributos de los ficheros que se han accedido últimamente, y los nombres de los ficheros, respectivamente. Y por otra parte las máquinas de estados finitos en el lado cliente y en el lado servidor. Estas máquinas de estado son las encargadas de ejecutar las distintas operaciones recibidas desde la capa de interfaz del sistema. Por ejemplo, existe una máquina de estados para la creación de ficheros en el lado cliente y una máquina para crear ficheros en el lado servidor. Estas máquinas de estado son un conjunto de pasos que van ejecutando las operaciones de acuerdo a los resultados de pasos anteriores y posteriores. Ha sido necesario el cambio de algunas de estas máquinas de estado para lograr la replicación de datos.

Figura 11.2 Arquitectura de software PVFS2.



11.3 Implementación de plataforma de alto rendimiento

El desarrollo de este trabajo consiste en la implementación de una plataforma de computación de alto rendimiento para fines de investigación que permita en parte la reutilización de los que quipos de computo de bajos recursos con los que cuenta el Programa Educativo de Ingeniería en Tecnologías de la Información de la Universidad Politécnica de Altamira.

Una de las ventajas de la implementación, es el de permitir que las tareas sean repartidas de forma balanceada entre cada uno de los servidores de E/S que forman el cluster.

Este cluster ofrecerá el procesamiento de grandes cantidades de datos de una manera más rápida, por lo que se reduce de forma considerable los tiempos de acceso a datos.

La idea viene derivada de la necesidad de adquirir servidores dedicados para realizar tareas de procesamiento, pero el costo elevado de estos se convierte en un problema para instituciones donde los recursos económicos son de difícil acceso; es por ello que un clúster es una excelente solución para resolver las necesidades a un bajo coste. Además que se involucrara a los estudiantes para que participen y así pueden ir adentrándose a temas de investigación.

La implementación inicia desde la selección del equipo y montaje en un rack, como el armado la red de interconexión. En este caso se ha estudiado el uso switches administrables.

El uso de estos dispositivo permite reducir el número de mensajes broadcast (implementación de vlan´s (VLAN´s)) y mejorar su rendimiento dado que es posible implementar redundancia en las conexiones para ofrecer alta disponibilidad y tolerancia a fallos. Posteriormente se instalara el sistema operativo, para ello se ha decidido utilizar un distribución GNU/Linux, así como el sistema de ficheros PVFS2.

11.4 Resultados experimentales

En esta sección se muestran algunos resultados obtenidos en un Cluster de computadores (Camacho) que cuenta con las características siguientes:

- Ocho nodos con:

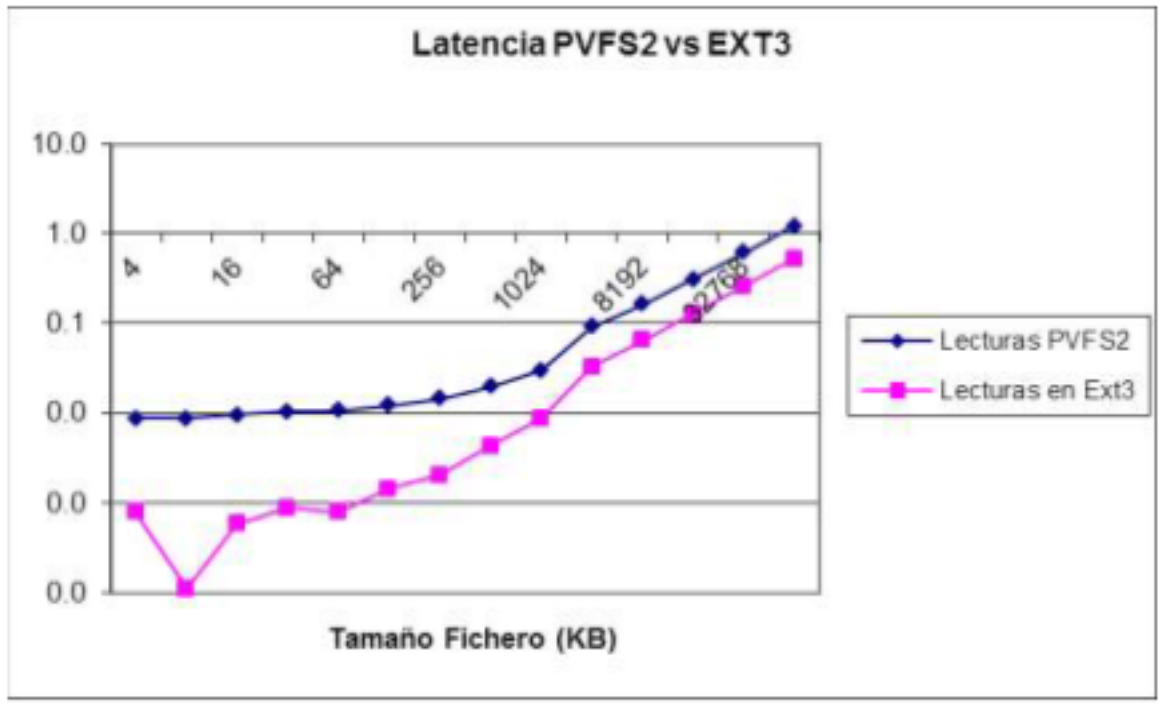
  2 Procesadores AMD Athlon MP 2400+ (frecuencia real de 2 GHz y 256 KB de cache de nivel 2), 2 GB de memoria principal y disco duro ST380011A IDE (7500 RPM, 100 MB/s de tasa de transferencia pico) en siete de los ocho nodos y disco ST373307LW SCSI (1000 RPM, 320 MB/s de tasa de transferencia pico, 59,9 MB/s de ancho de banda sostenido) en el nodo con conexión al exterior.

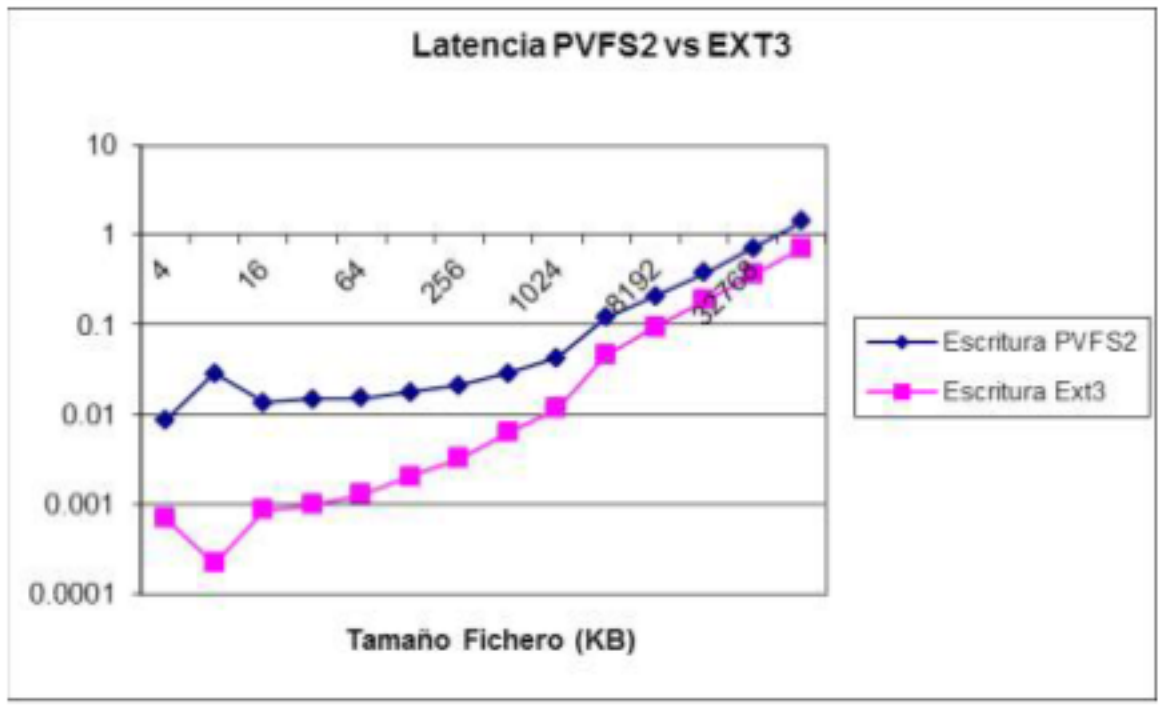
- Un switch Gigabit Ethernet 3COM 4900

Se ha utilizado el benchmark Interleaved or Random (IOR) (Interleaved-Or-Random Filesystem Benchmarking) para evaluar las prestaciones de PVFS2 y EXT3 (Grafico 11).

Los resultados para PVFS2 se obtuvieron con la configuración siguiente: 1 servidor de metadatos, 1 servidor de E/S y 1 cliente, empleando una unidad de striping de 64MB y distintos tamaños de fichero desde 4KB hasta 64 MB donde se lee y escribe. De esta forma el fichero se encuentra almacenado en un servidor en lugar de encontrarse distribuido en distintos servidores.

Grafico 11 Latencia PVFS2 Vs EXT3.





En las ejecuciones realizadas con Ext3, el cliente accede a su sistema de ficheros local; no hay por tanto acceso remoto. En cambio, en las ejecuciones realizadas con PVFS2, el cliente accede a un servidor remoto. Como se puede observar los accesos con PVFS2 tanto para escritura como para lectura presentan una importante penalización frente a los accesos locales (*figura 4*). En los resultados mostrados en las figuras no se ve reflejado la penalización que supone acceder a disco, porque el tamaño de los accesos no supera el tamaño de la cache del sistema de ficheros local. La presencia de la cache queda reflejada en el hecho de que el ancho de banda obtenido con ext3 supera el ancho de banda pico del disco duro (100 MB/s).

El tamaño de la cache que se utiliza en el sistema de ficheros local depende de la cantidad de memoria disponible en el servidor. En las pruebas realizadas los servidores sólo atiende a las peticiones de los clientes, no están ejecutando ninguna otra aplicación.

En estas ejecuciones se ha comprobado que la cache de disco utilizada es de 1 GB. Conforme aumenta la cantidad de memoria ocupada, se reduce la cache de disco; por ejemplo, si se ejecutan en el servidor aplicaciones que ocupan el 70% de la memoria principal la cache de disco se reduce a 225 MB, y si ocupan el 80%, se reduce a 106 MB.

No obstante, para obtener buenas prestaciones es necesario acceder a distintos servidores los cuales almacenen partes del fichero (*Grafico 11.1*). Para estos resultados se utilizó la configuración siguiente: 1 servidor de metadatos, 2 servidores de E/S y una variedad de clientes desde 1 a 4.

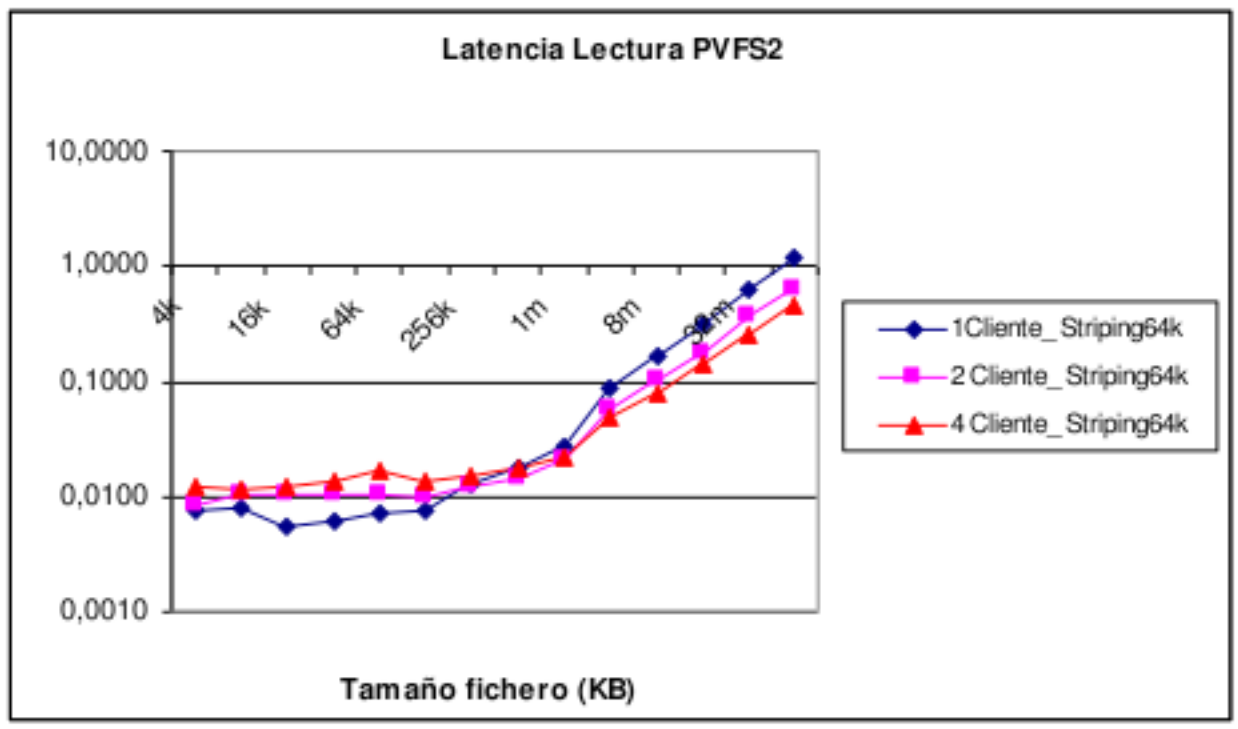
Se debe tener en cuenta al interpretar los resultados que en las lecturas se accede a los datos que previamente se han escrito; por tanto estas lecturas se benefician al igual que las escrituras de la cache que incluye el sistema de ficheros local. Como se muestra en la Grafico 11.1 Latencia en Lectura/Escritura en PVFS2.

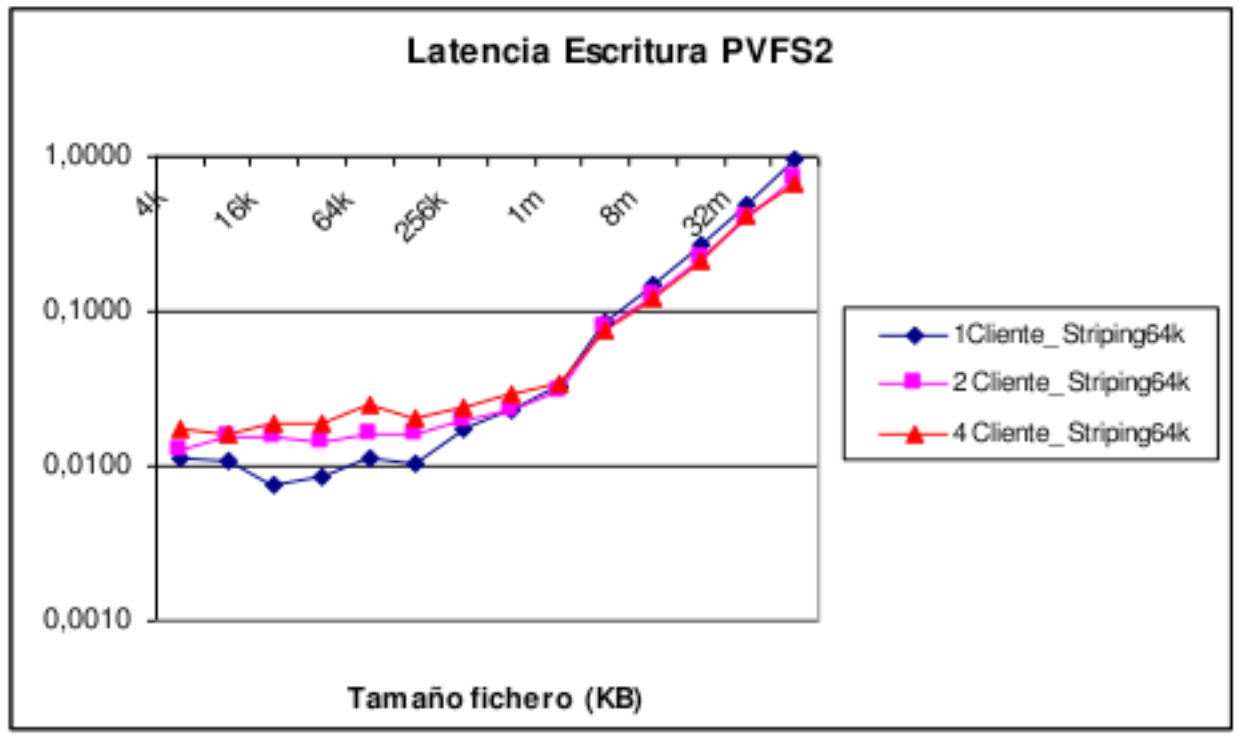
El implementar un número mayor de servidores el tiempo de acceso a los datos se reduce con respecto a la implementación de un único servidor.

Esto se debe a la distribución de los datos en varios servidores, lo que permite poder acceder a partes del fichero en paralelo. Mejor aún es el hecho de poder gestionar un fichero con distintos clientes, debido a que cada uno de ellos puede gestionar una parte del fichero, por lo que el paralelismo incrementa aún más.

Estos resultados permiten deducir que la implementación de un cluster de computadores en la universidad politécnica de Altamira es buena idea para lograr mayor rendimiento en el procesamiento de grandes cantidades de datos.

Grafico 11.1 Latencia en Lectura/Escritura en PVFS2.





11.5 Conclusiones y trabajos futuros

A lo largo de este trabajo se ha descrito las características de los cluster y sistemas de fichero, centrándose principalmente en la implementación de un clúster de computadores para procesar grandes cantidades de datos.

Una vez realizado el análisis sobre los cluster y PVFS2, se ve claramente que la implementación ofrece prestaciones similares o mejores comparadas con los sistemas dedicados y de alto coste para el procesamiento de datos. Las pruebas realizadas han permitido ver las ventajas de se pueden esperar con la utilización de esta implementación. No obstante se hace hincapié a que el uso de más servidores de E/S permite obtener mejores prestaciones.

Como trabajo futuro en pro de las tecnologías verdes se pretende aprovechar los recursos del cluster mediante la Virtualización de servicios dedicados. Es decir, crear múltiples servidores virtuales en un cluster con el fin de obtener el máximo beneficio de la tecnología implementada y así contribuir en la reducción del consumo energético, buscando impactar de manera favorable al medio ambiente.

11.6 Referencias

Anguita, M. (2011). *Sistemas de FIcheros.* Granada: Universidad de Granada.

Camacho, H. E. (s.f.). Tesis Master "Cache en SIstemas de Ficheros Distribuidos y Paralelos". Granada, España.2007

Carns, P. H., Walter B. Ligon III , Robert B., R., & Rajeev Thakur . (2000). PVFS: A Parallel File System for Linux Clusters. 11. Obtenido de http://www.pvfs.org/

Diaz, A. F. (2012). Two-level Hash/Table approach for Metadata Management in Distributed File Systems. *Journal of Supercomputing*.

Interleaved-Or-Random Filesystem Benchmarking . (s.f.). Obtenido de http://freecode.com/projects/ior

MPICH. (s.f.). *Message Passing Interface*. Obtenido de www.mpich.org

OrangeFS. (s.f.). Obtenido de http://www.pvfs.org/cvs/pvfs-2-8-branch.build/doc/pvfs2-faq/

PVFS2. (s.f.). Obtenido de www.pvfs.org

VLAN´s. (s.f.). Obtenido de http://www.cisco.com/en/US/docs/switches/lan/catalyst6500/ios/12.2SX/configuration/guide/vlans.html

W. B. Ligon III, R. B. (1999). Beowulf: Low-Cost Supercomputing Using Linux. *IEEE Software magazine special issue on Linux*, Volume 16, Number 1, pp 79.