MONTAJE DE UN CLÚSTER DE ALMACENAMIENTO ADMINISTRADO POR PVFS (PARALLEL VIRTUAL FILE SYSTEM)

GERARDO CHONA QUINTERO - U00010984

HUGO PEÑA PERDOMO - 100002633

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
TELECOMUNICACIONES Y TECNOLOGÍAS WEB
BUCARAMANGA
2008

MONTAJE DE UN CLÚSTER DE ALMACENAMIENTO ADMINISTRADO POR PVFS (PARALLEL VIRTUAL FILE SYSTEM)

GERARDO CHONA QUINTERO - U00010984
HUGO PEÑA PERDOMO - 100002633

Proyecto de grado presentado como
Requisito parcial para optar el título
De "Ingeniero de Sistemas"

Director: Ing. Roberto Carvajal Salamanca
Asesor: Ing. Wilson Castaño

UNIVERSIDAD AUTÓNOMA DE BUCARAMANGA
FACULTAD DE INGENIERÍA DE SISTEMAS
TELECOMUNICACIONES Y TECNOLOGÍAS WEB
BUCARAMANGA
2008

Nota de aceptación:

_____

_____

_____

_____

_____

_____

_____

_____

_____

_____

Firma del presidente del jurado

_____

Firma del jurado

_____

Firma del jurado

Bucaramanga, Enero de 2009.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

INTRODUCCIÓN

En la actualidad las universidades y en general muchas organizaciones mantienen una infraestructura de computadores grande y costosa que es necesaria para realizar las tareas diarias de la organización, pero que durante muchas horas del día o de la noche permanece ociosa. En las universidades es más evidente esta problemática, encontrándose aulas completas de computadoras sin ningún tipo de uso gran parte del tiempo.

Por otra parte y en el caso de instituciones educativas, existen muchos proyectos de investigación que requieren alto poder de cómputo y de almacenamiento, el cual puede ser obtenido a través de la compra de súper computadores con gran capacidad de almacenamiento y disponibilidad de procesadores de alto rendimiento cada uno, pero de gran costo.

Desde hace varios años se han intentado soluciones menos costosas para la implementación de la computación paralela y de sistemas de almacenamiento masivo, como clústeres de computadores, aplicaciones distribuidas especializadas e infraestructuras grid. Todas estas tecnologías han logrado solucionar los problemas para los que fueron diseñados en su época y algunos de estas soluciones son usadas en nuestros días.

La tecnología grid, que es una concepción mayor que el clúster, brinda a entidades educativas, centros de investigación y empresas privadas, la oportunidad de compartir recursos tanto de hardware como de software con otras instituciones aumentando incalculablemente su potencial computacional a través

de una topología física interconectada a través de redes de área local y de área amplia, en la que se incluyen equipos de alto rendimiento y agrupaciones que hacen uso del procesamiento y almacenamiento en paralelo.

La presente propuesta de investigación busca llevar a cabo una exploración y prueba de herramientas de software libre que permitan el montaje de una infraestructura de almacenamiento. El papel fundamental de las arquitecturas es que los sistemas de almacenamiento están pasando de ser modelos de recursos específicos a modelos de utilidad, lo que proporciona un almacenamiento transparente y bajo demanda, al permitir que todos los recursos de almacenamiento de la empresa estén disponibles para cualquier operación.

Muchas aplicaciones científicas y de ingeniería requieren el acceso a grandes cantidades de datos (terabytes o petabytes). El tamaño y el número de estos datos ha venido creciendo rápidamente en los últimos años y seguirá creciendo a medida que nuevos experimentos, los costos de la computación, el almacenamiento de datos y las nuevas aplicaciones de la ciencia computacional se desarrollen. Se han desarrollado soluciones de software que permiten aumentar la capacidad de cómputo del hardware disponible actualmente.

Por tal motivo, analizar, seleccionar y probar tecnologías de almacenamiento de grandes volúmenes de información, se considera un tema importante de investigación para la Universidad Autónoma de Bucaramanga, con miras a aprovechar los recursos disponibles (computadores y redes de interconexión entre otros).

1. CLÚSTER

1.1 CLÚSTER

En muchas ramas de las ciencias, la complejidad de los problemas que se estudian, requieren el acceso a una supercomputadora para el procesamiento y almacenamiento de datos, siendo estas máquinas tan poderosas que pueden desarrollar varios miles de millones de operaciones por segundo y escalar su capacidad de almacenamiento a los niveles de terabytes o petabytes.

Algunos fabricantes, como es el caso de IBM, se dieron a la tarea de crear computadores con capacidades superiores a los convencionales, ofreciéndolos a precios exagerados. Para solucionar el problema de costos y aumentar el poder de cómputo Thomas Sterling, Donald Becker y otros científicos de la NASA iniciaron el proyecto “clúster” y comenzaron a desarrollar el primer clúster de computadores personales. Éste tuvo una eficiencia de 70 megaflops (millones de operaciones de punto flotante por segundo).

Actualmente la tecnología clúster es cada vez más estudiada y utilizada en el mundo por estudiantes, científicos e investigadores, y se encuentra en constante evolución.

Un clúster es la agrupación de computadores que trabajan con un fin común. El cómputo en clústeres surge como resultado de la convergencia de varias tendencias que incluyen, la disponibilidad de microprocesadores de alto rendimiento más económicos y redes de alta velocidad, el desarrollo de herramientas de software para cómputo distribuido de alto rendimiento, y la

creciente necesidad de potencia computacional para aplicaciones en las ciencias computacionales y comerciales.

Existen 3 tipos de tecnología clúster:

- El clúster de alto rendimiento. Donde un número grande de computadores están interconectados por una red de comunicación; agrupan hardware y software libre para trabajar en conjunto como si fueran un solo sistema. Este tipo de clúster se utiliza para problemas complejos que requieren una cantidad enorme de potencia computacional.

- El clúster de servidores virtuales. Permite que un conjunto de servidores de red cooperen para repartir la carga de tráfico de clientes entre ellos.

- El clúster de alta disponibilidad. Que permite que el grupo de computadores cooperen y hagan copia de la información que manejan.

Existen diferentes razones para realizar clústeres, pero la más importante tal vez sea que el almacenamiento de información sea más eficiente y transparente, como si se realizara en un solo sistema de almacenamiento. Las ventajas en capacidad de almacenamiento hacen que este tipo de soluciones para el almacenamiento de datos sean consideradas muy atractivas y en constante evolución.

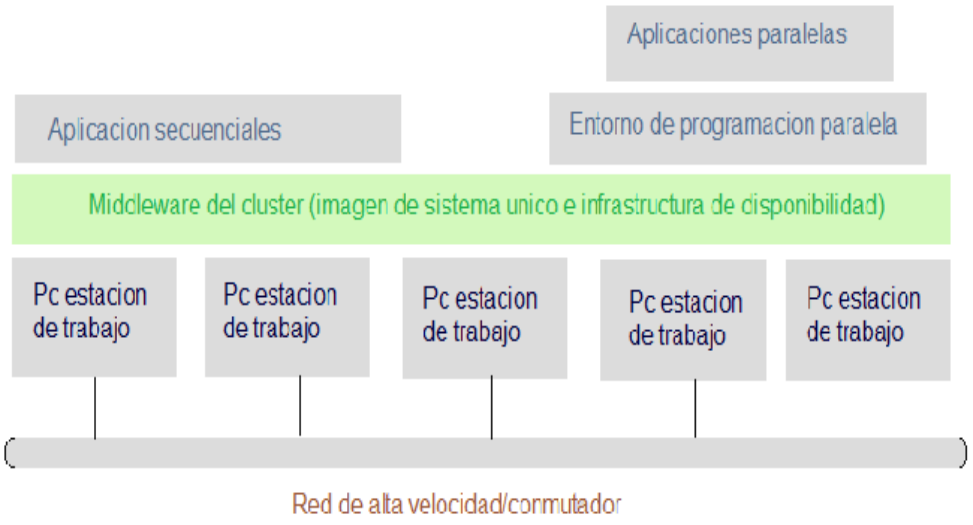
Un clúster trabaja generalmente en una red de área local (LAN) y si las máquinas están en un espacio cercano físicamente logran una buena comunicación; en un concepto más extenso se encuentra grid que trabaja con redes de área extensa (WAN). Una grid es un tipo de sistema paralelo y distribuido que permite compartir, seleccionar y añadir recursos que se encuentran distribuidos a lo largo de dominios administrativos "múltiples" basados en su disponibilidad, capacidad, rendimiento, costo y calidad de servicio que requiere un usuario.

1.2 ARQUITECTURA DE CLÚSTER

En la arquitectura típica de clúster los computadores se conectan a través de una red de área local (LAN) de alta velocidad o mediante un conmutador (SWITCH). Cada computador puede trabajar de forma independiente. Además, en cada computador se instala una capa de software intermedia (middleware) que permite el funcionamiento de todos los computadores como un único clúster.

El middleware del clúster proporciona al usuario una imagen unificada conocida como imagen del sistema único (single-system image). El middleware también es responsable de proporcionar alta disponibilidad, distribuyendo la carga y respondiendo a los fallos de los componentes.

Figura 1. Arquitectura de computador clúster.



Fuente: William Stallings. Organización Y Arquitectura De Computadoras [texto] Capitulo: Procesamiento paralelo, organización y arquitectura de computadores [Citado Marzo 2008]

1.3 FUNCIONAMIENTO DE UN CLÚSTER

La tecnología clúster está dividida en dos partes fundamentales:

- Un sistema operativo robusto y estable elaborado para hacer modificaciones al kernel de Linux con compiladores y aplicaciones especiales que permiten que los programas que corran el sistema operativo (S.O) puedan tomar las ventajas que brinda un clúster.

- La interconexión de hardware entre los nodos del clúster; se han creado diversas interfaces para la comunicación, pero las más común es conectarse a través de una red Ethernet dedicada de alta velocidad para comunicarse, asignar tareas, recursos, entre otras cosas.

La tecnología de clústeres de alto rendimiento para GNU/LINUX más conocida es la tecnología Beowulf. Esta tecnología se desarrolló en la NASA y puede proporcionar potencial de cómputo del tipo de una supercomputadora, utilizando computadoras personales sencillas.

1.4 CLÚSTER DE ALMACENAMIENTO

Las grandes compañías afrontan un incremento enorme en las cantidades de datos necesarios para gestionar su actividad diaria, lo que ha cambiado radicalmente la manera en que las organizaciones evalúan, seleccionan y despliegan la tecnología.

A medida que más compañías se abren paso en el uso de clústeres almacenamiento para satisfacer sus necesidades, el almacenamiento en clústeres se ha convertido en una solución extremadamente atractiva para mejorar sus infraestructuras informáticas, mejorando la acceso y el análisis a los datos con el fin de proporcionar mejores resultados.

Tres grandes tendencias están motivando esta necesidad de utilizar sistemas de almacenamiento en clústeres en varios mercados e industrias:

1. La avalancha de contenido digital y datos desestructurados.

2. La necesidad de contar con hardware estandarizado y con software inteligente para lograr ahorros en equipos y ventajas en el rendimiento.

3. Lograr un avance tecnológico fundamental ya que la solución a problemas de almacenamiento se está moviendo hacia arquitecturas informáticas en clúster.

Aplicaciones actuales que usan video, audio, imágenes, archivos de investigación y otros archivos digitales de gran tamaño tienen requerimientos para los cuales los sistemas de almacenamiento tradicionales no están diseñados. Los contenidos digitales y los datos desestructurados consumen grandes cantidades de capacidad de almacenamiento, crecen más rápidamente y requieren una alta tasa de transferencia y un alto nivel de concurrencia.

Los sistemas de almacenamiento, tradicionalmente monolíticos y de gran tamaño, fueron reemplazados por otros de tamaño medio (aunque todavía monolíticos). En la actualidad, sin embargo, el sector está evolucionando hacia sistemas de clústeres de almacenamiento en donde se ve mayor rendimiento y son fácilmente ampliables.

Lo mejor de estos sistemas de almacenamiento en clústeres es que funcionan con un hardware estándar y disponen de un software que permite a los diferentes componentes, o nodos, trabajar conjuntamente a la perfección como un único sistema. Los sistemas de almacenamiento tienen más potencia colectiva, cuestan entre un 40 y un 60% menos que los sistemas tradicionales de almacenamiento y su escalabilidad es muy superior.

Por ejemplo, mientras que los sistemas tradicionales apenas alcanzaban varios Gigabytes por cada sistema de archivos, los sistemas en clústeres pueden escalar hasta terabytes y petabytes.

La apuesta por el almacenamiento en clústeres es vital para superar las cuatro grandes limitaciones que los sistemas tradicionales de almacenamiento tienen a la hora de trabajar con contenido digital y datos desestructurados.

El almacenamiento tradicional:

- Genera “cuellos de botella” en el rendimiento.

- Es complejo y de difícil escalabilidad.

- Contiene puntos únicos inherentes de fallo en los sistemas en resumen, la evolución a sistemas de almacenamiento en clústeres ofrece a los negocios sistemas que son más fiables y fáciles de gestionar. Un sistema de almacenamiento adecuado permite a los ejecutivos de empresas y a sus respectivos negocios redefinir sus presupuestos de almacenamiento y reducir el coste total de su actividad permitiendo a la compañía “escalar a medida que crece” y manejar cantidades de datos digitales de una manera segura y eficiente.

Características que ofrecen un clúster de almacenamiento son:

- Arquitecturas en clústeres modulares que permitan al sistema crecer de manera flexible y que proporcionen ventajas en el acceso simultáneo a los datos.

- Software inteligente integrado que proporcione facilidad de gestión y un interfaz de fácil manipulación a fin de poder llevar a cabo cambios en la capacidad del sistema o en los niveles de protección sin esfuerzo alguno.

- Sistemas optimizados para alojar archivos de gran tamaño, un alto rendimiento y un gran núme0ro de usuarios simultáneos.

- Equilibrio de carga que permita distribuir a partes iguales las conexiones entre clientes y aplicaciones dentro de un clúster mediante sistemas de software inteligente

- Uso de hardware estándar que aproveche las ventajas de los estándares Ethernet como estructura de interconexión y utilice protocolos estándar para comunicarse con otras aplicaciones y clientes, reduciendo los costes innecesarios y la complejidad de configuración.

2. SISTEMAS DE ALMACENAMIENTO DE ALTO RENDIMIENTO

Los sistemas de almacenamiento de alto rendimiento hacen referencia a una rama de la computación que se centra fundamentalmente en la solución de problemas que hacen un uso intensivo del almacenamiento de datos. Hace unos años, los sistemas de almacenamiento de alto rendimiento dominados por sistemas grandes y especializados (costosos) que se encontraban principalmente en centros de investigación. A medida que la capacidad de almacenamiento de los sistemas pequeños ha aumentado, ha cambiado la relación costo-rendimiento y las cargas de almacenamiento se han desplazado a los sistemas de PC.

Muchos problemas de almacenamiento masivo se pueden solucionar mediante la realización de sistemas de almacenamiento en paralelo, o mediante sistemas de almacenamiento distribuidos o por los sistemas denominados tolerante a fallos, todos estos nos lleva a solucionar los problemas que se plantean a la hora de poder reducir los costos y aumentar el rendimiento en el almacenamiento de datos.

A continuación se explica cada uno de los sistemas de almacenamiento mas conocidos y con los que los sistemas han logrado encontrar solución a la hora de manejar gran cantidad de información.

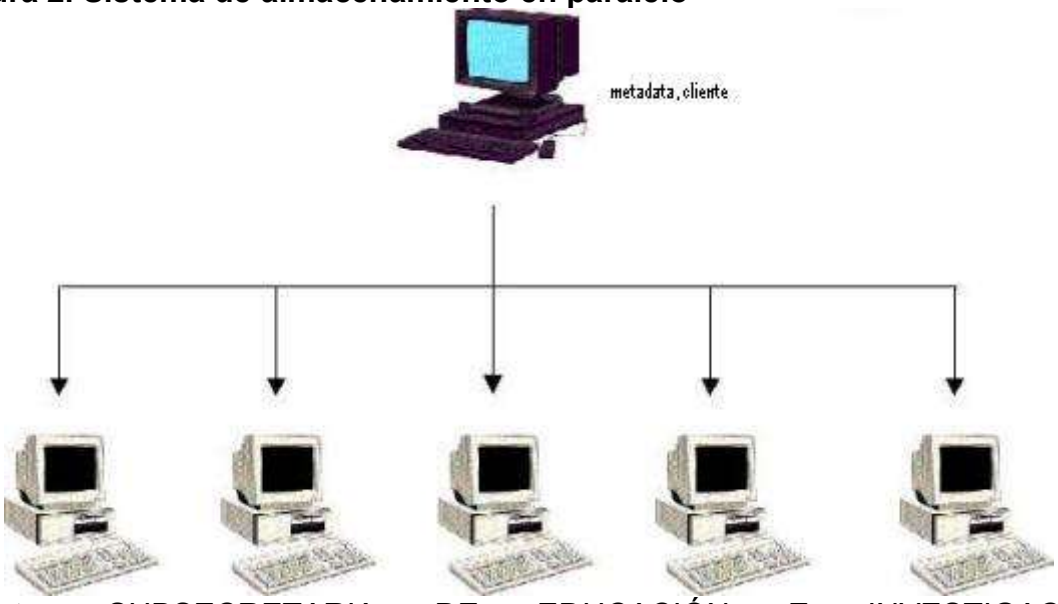
2.1 SISTEMAS DE ALMACENAMIENTO EN PARALELO

Son aquellos que tienen la capacidad de realizar varias operaciones de almacenamiento de manera simultánea. Generalmente estos sistemas suelen manejar grandes cantidades de información del orden de los terabytes y pueden

procesar muchas peticiones de manejo de información en poco tiempo, demostrando su eficacia a la hora de aumentar su desempeño. Los sistemas paralelos están formados por varias computadoras que comparten la información, los recursos y la memoria.

Un sistema de almacenamiento en paralelo está diseñado para eliminar el servidor de archivos como el principal cuello de botella de entrada/salida, El servidor de archivos se encuentra presente en los sistemas de almacenamiento distribuido.

Figura 2. Sistema de almacenamiento en paralelo



Fuente: SUBSECRETARIA DE EDUCACIÓN E INVESTIGACIÓN TECNOLÓGICAS, DIRECCIÓN GENERAL DE EDUCACIÓN SECUNDARIA TÉCNICA [online] Sistema de almacenamiento en paralelo [citado Agosto 2008], <URL: http://sectec.ilce.edu.mx/sectec40/image>

Básicamente el funcionamiento de este sistema está basado en tener un servidor de metadatos; lo que se hace es que desde un cliente se manda a guardar determinada información y ésta es enviada al servidor de metadatos que es el

encargado de distribuirlo en partes iguales a lo largo del sistema. El servidor de metadatos almacena la información de referencia la cual indica donde están guardados cada uno de las partes del archivo. A la hora de hacer uso de un archivo almacenado en el clúster se hace la petición al servidor de metadatos y este con base a la referencia de ubicación en el sistema de cada parte del archivo almacenado, proporciona la información de ubicación de cada archivo y el sistema se encarga de enviar cada parte al cliente para que pueda hacer uso de él en su totalidad.

Los sistemas de archivos con características de paralelismo se encuentran: PVFS, Lustre, G Farm, Google fs, Gluster fs.

2.2 SISTEMAS DE ALMACENAMIENTO DISTRIBUIDOS

Son aquellos cuyos componentes hardware y software, que están en computadores conectados en red, se comunican y coordinan sus acciones mediante el paso de mensajes para lograr el almacenamiento de información.
Se establece la comunicación mediante un protocolo prefijado por un esquema cliente-servidor de archivos.

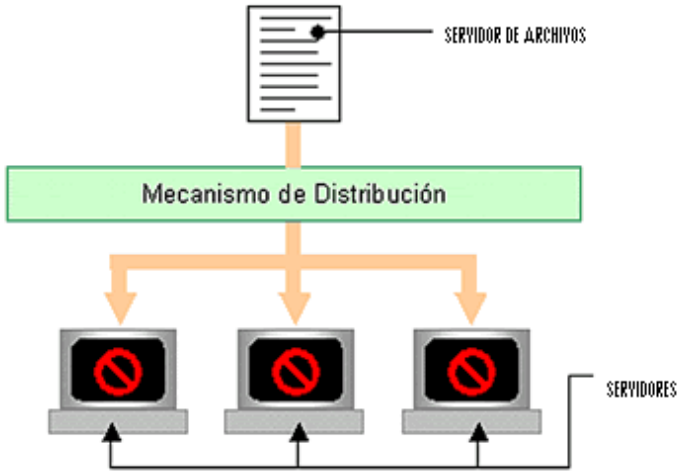
Algo importante que caracteriza al almacenamiento distribuido es el gran rango de aplicaciones que tiene. Las tres más importantes:

- Copias de seguridad de los archivos. Posiblemente se trate de la más útil. Siempre conviene tener una copia de los archivos más relevantes del PC para poder recuperarlos en caso de daño del disco duro o fallo del sistema.

- Compartir archivos en red. Cuando hay que realizar trabajos o proyectos en grupo se trabaja con una única copia. De ese modo cualquier participante puede editar el documento sin necesidad de andar con distintas versiones del mismo.

- Poder editar un documento desde distintas localizaciones sin la necesidad de utilizar memorias externas.

Figura 3. Sistema de almacenamiento distribuido.



Fuente: TextosCientíficos.com [online], Sistema de almacenamiento distribuido [marzo 2008] <URL: http://www.textoscientificos.com/imagenes/redes>

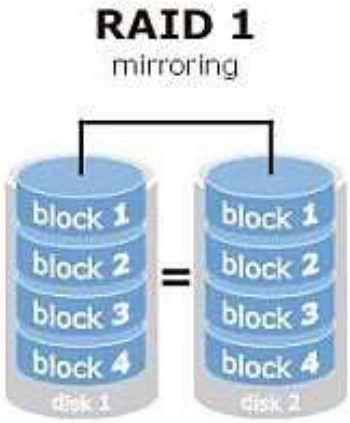
El punto débil de esta aplicación es sin duda alguna la seguridad. Hoy en día la red es demasiado vulnerable a ataques externos por lo que las grandes empresas desconfían mucho de este tipo de servicios. Otro inconveniente que tiene es que no es recomendable usarlo como almacenamiento personal o privado ya que al ser gratuitas muchas de ellas, los proveedores no se hacen cargo en caso de pérdida de archivos.

Algunos de los sistemas de archivos utilizados para almacenamiento distribuido son: GFS, NFS (Network File System), Coda, Gluster fs.

2.3 SISTEMA DE ALMACENAMIENTO TOLERANTE A FALLOS

Hace referencia a un sistema de almacenamiento que usa múltiples discos duros entre los que se replica los datos. los beneficios de un sistema de almacenamiento tolerante a fallos respecto a un único disco son uno o varios de los siguientes: mayor integridad, mayor tolerancia a fallos, mayor rendimiento y mayor capacidad.

Figura 4. Sistema de almacenamiento tolerante a fallas.



Fuente: Derecho y nuevas tecnologías [online], El Derecho y la Sociedad de la Información [Citado Agosto 2008]
<URL: http://www.derechonntt.com/wp-content/uploads>

La tolerancia a fallos es un aspecto crítico para aplicaciones a gran escala, ya que algunas aplicaciones o simulaciones que puedan tardar del orden de varios días.

3. SISTEMAS DE ARCHIVOS

Los sistemas de archivos estructuran la información guardada en una unidad de almacenamiento (normalmente en un disco duro) de un computador, que luego será representada ya sea textual o gráficamente utilizando un gestor de archivos.

La mayoría de los sistemas operativos poseen su propio sistema de archivos.

Lo habitual es utilizar dispositivos de almacenamiento de datos que permiten el acceso a los datos como una cadena de bloques de un mismo tamaño, a veces llamados sectores. El software del sistema de archivos es responsable de la organización de estos sectores en archivos y directorios y mantiene un registro de qué sectores pertenecen a qué archivos y cuáles no han sido utilizados. En la práctica, un sistema de archivos también puede ser utilizado para acceder a datos generados dinámicamente, como los recibidos a través de una conexión de red (sin la intervención de un dispositivo de almacenamiento).

Los sistemas de archivos tradicionales proveen métodos para crear, mover, renombrar y eliminar tanto archivos como directorios.

3.1 SISTEMAS DE ARCHIVOS DISTRIBUIDOS

3.1.1 Red Hat global file system (GFS). Es un sistema de archivos para clúster. Red Hat global file system permite proporcionar datos entre nodos del clúster. GFS permite instalar aplicaciones y ejecutarlas sin mucho conocimiento de la infraestructura de almacenamiento. Permite llevar a cabo las operaciones con archivos como si estuvieran corriendo en un sistema de archivos local.

GFS proporciona un método versátil de la creación de redes de almacenamiento de acuerdo con el rendimiento, la escalabilidad, y las necesidades económicas de su entorno de almacenamiento.

Red Hat global file system habitualmente se utiliza en clústers de aplicaciones de empresas para brindar acceso de alta velocidad a una imagen de sistema de archivo consistente en los nodos del servidor. Esto permite a los nodos del clúster leer y escribir simultáneamente en un único sistema de archivos compartido. Los clústers de aplicación habituales donde red Hat GFS se implementa hoy incluyen:

- Base de datos (incluyendo Oracle Rac).
- Aplicaciones y servidores web.
- Aplicaciones domésticas.
- Clústers de procesamiento de alto rendimiento donde la escalabilidad y el rendimiento son claves.

Red Hat global file system permite a un clúster de servidores Linux compartir datos en un conjunto común de almacenamiento, permitiéndole:

- Simplificar ampliamente su infraestructura de datos.
- Instalar y parchear las aplicaciones una vez, para todo el clúster.
- Reducir la necesidad de copias redundantes de datos.
- Simplificar las tareas de recuperación de desastres y back-up.

- Maximizar el uso de los recursos de almacenamiento y minimizar los costos del mismo.

- Administrar su capacidad de almacenamiento como un todo versus la partición.

- Disminuir sus necesidades de almacenamiento general reduciendo la duplicación de datos.

- Escalar clústers de manera transparente, incorporando almacenamiento o servidores.

- Se acabo el almacenamiento particionado con técnicas complicadas.

- Agregar servidores simplemente montándolos al sistema de archivo común.

- Lograr el máximo tiempo de productividad de aplicaciones.

Además Red Hat global file system es:

- El único sistema de archivo clúster originario de 64-bit en Linux para las cargas de trabajo de empresas - soporte para x86, amd64/em64t, e Itanium.

- El sistema de archivo de GFS en clúster es más escalable soportando hasta 100 nodos.

- Estrechamente integrado con red Hat Enterprise Linux (rastrea las versiones de kernel y tiene soporte durante toda su vida útil).
- El primer sistema de archivo clúster open sources (GPL) para cargas de trabajo de empresas.

- Completamente compatible con Posix, que significa que las aplicaciones no deben ser escritas nuevamente para usar GFS.

3.1.2 Network file system (NFS). Es un sistema de archivos de red permite a los hosts remotos montar sistemas de archivos sobre la red e interactuar con esos sistemas de archivos como si estuvieran montados localmente. Esto permite a los administradores de sistemas consolidar los recursos en servidores centralizados en la red.

Hay dos versiones de NFS actualmente en uso. La versión 2 de NFS (nfsv2), que tiene varios años, es ampliamente soportada por muchos sistemas operativos. La versión 3 de NFS (nfsv3) tiene más características, incluyendo manejo de archivos de tamaño variable y mejores facilidades de informes de errores. NFSV2 utiliza el protocolo de datagrama de usuarios (UDP) para proporcionar conexiones de red sin supervisión (Stateless) entre el cliente y el servidor. nfsv3 puede utilizar bien sea UDP o el protocolo de control de transmisiones (TCP) ejecutándose sobre una red IP.

Bajo condiciones normales la conexión UDP desatendida minimiza el tráfico de la red, ya que el servidor NFS envía un cookie al cliente después que éste tiene acceso al volumen compartido. Esta cookie es un valor aleatorio guardado en el lado del servidor y es pasado junto con las peticiones RPC desde el cliente. El servidor NFS puede ser reiniciado sin afectar a los clientes y las cookies permanecen intactas. Sin embargo, debido a que UDP es sin supervisión, si el servidor se cae de forma inesperada, los clientes UDP continúan saturando la red con peticiones para el servidor. Por esta razón, TCP es el protocolo preferido cuando se conecte a un servidor nfsv3.

3.1.3 Coda. Es un sistema de archivos que tiene sus orígenes en afs2. Tiene múltiples características que son deseables en la mayoría de sistemas de archivos. Además, tiene algunas características propias.

1. Puede funcionar sin conexión
2. Es software libre
3. Gran rendimiento gracias a la caché persistente en el cliente
4. Replicado de servidores
5. Modelo de seguridad para autenticación, cifrado y control de acceso
6. Funcionamiento continuado durante fallos de red
7. Ajuste del ancho de banda de red

Coda utiliza una caché local para proporcionar acceso a los datos del servidor cuando ocurren desconexiones en la red. Durante el funcionamiento normal, un usuario lee y escribe al sistema de archivos con normalidad, mientras el cliente obtiene todos los datos que ha marcado como importantes en el caso de una desconexión de red. Cuando se pierde la conexión de red, el cliente coda sirve los datos desde su caché local y registra cualquier actualización. A este estado se le llama funcionamiento sin conexión. Al restablecerse la conexión, el cliente coda pasa del funcionamiento sin conexión hacia un estado transitorio de "reintegración" donde las actualizaciones registradas se envían a los servidores. Cuando todas las actualizaciones se han reintegrado, el cliente vuelve al estado normal de funcionamiento con conexión.

Otra característica de coda es su método de replicación de datos. Coda permite a todos los servidores recibir actualizaciones, logrando una mayor disponibilidad de los datos del servidor en caso de que la red se particione. Estas características únicas introducen la posibilidad de copias divergentes de los mismos archivos o directorios, llamadas "conflictos". Las actualizaciones procedentes del modo de funcionamiento desconectado pueden potencialmente solaparse con las

actualizaciones de otros usuarios sobre los mismos objetos, impidiendo la reintegración. La replicación optimista puede potencialmente causar actualizaciones concurrentes en diferentes servidores del mismo objeto, evitando la replicación. Se conoce al caso como conflicto "local/global" y "server/server". Coda tiene un amplio abanico de herramientas de reparación, tanto manuales como automatizadas, para manejar y evitar ambos tipos de conflicto.

3.2 SISTEMAS DE ARCHIVOS TOLERANTES A FALLOS

3.2.1 Gluster fs. Es un sistema de archivos capaz de escalar a varios petabytes. Por medio de diversos nodos agregados al clúster de almacenamiento. Es un sistema de archivos en red basado en conexiones INFINIBAND RDMA o TCP/IP y su interconexión es en un gran sistema de archivos de red en paralelo. Este sistema de archivo tiene la gran característica que se puede acomodar a las necesidades del usuario según como se configure el sistema de archivos. Gluster fs puede ser un sistema de archivos tolerante a fallos, distribuido o paralelo.

Gluster fs pueden escalar hasta petabytes o terabytes. Gluster fs corrige errores sobre su propia la marcha. El acceso al almacenamiento de respaldo o réplica en el caso de que se configure con tolerancia a fallos es directamente a los archivos y a las carpetas (estilo NFS).

3.2.2 Gfarm fs. Es un sistema de archivos open source. Tolerante a fallos y con características en paralelo. Por lo general utiliza un grupo de nodos para poder escalar en un buen número de gigabytes o petabytes.

Es una implementación de referencia de una arquitectura grid llamada Datafarm diseñada para poder escalar a petabytes.

3.3 SISTEMAS DE ARCHIVOS EN PARALELO

3.3.1 Lustre. Es un sistema de archivos distribuido open source con muchas características de sistemas de archivo en paralelo. Este entra en la clasificación de archivos en paralelo debido a que cuenta con un servidor de metadatos. El nombre es una mezcla de Linux y Clústers. El proyecto intenta proporcionar un sistema de archivos para Clústers de decenas de miles de nodos con petabytes de capacidad de almacenamiento, sin comprometer la velocidad o la seguridad, y está disponible bajo la GNU GPL.

Clúster file system son los diseñadores, desarrolladores y mantenedores de lustre con colaboraciones de otras compañías y particulares.

Muchos de los supercomputadoras más rápidos del mundo son Clústers que utilizan el sistema de archivos lustre como almacenamiento, como los sistemas en ornl, PNNL, LLNL y LANL.

Se considera a cada archivo almacenado en el sistema de archivos lustre un objeto. Lustre presenta a todos los clientes una semántica Posix estándar y acceso concurrente lectura y escritura para los objetos compartidos. Un sistema de archivos lustre tiene cuatro unidades funcionales. Estas son: meta data server (MDS) para almacenar los metadatos; un Object Storage target (OST) para guardar los datos reales; un Object Storage server (OSS) para manejar los OSTS; cliente(s) para acceder y utilizar los datos. Los OSTS son dispositivos de bloques. Un MDS, OSS, y un OST pueden residir en el mismo nodo o en nodos diferentes.

Lustre no administra directamente los OSTS, y delega esta responsabilidad en los OSSS para asegurar la escalabilidad para grandes Clústers y supercomputadores. En un Massively Parallell Processor (MPP), los procesadores pueden acceder al sistema de archivos lustre redirigiendo sus peticiones e/s hacia el nodo con el servicio lanzador de tareas si está configurado como un cliente lustre. Aunque es el método más sencillo, en general proporciona un bajo rendimiento. Una manera ligeramente más complicada de proporcionar un rendimiento global muy bueno consiste en utilizar la biblioteca Liblustre. Liblustre es una biblioteca de nivel de usuario que permite a los procesadores montar y utilizar el sistema de archivos lustre como un cliente, sorteando la redirección hacia el nodo de servicio. Utilizando Liblustre, los procesadores pueden acceder al sistema de archivos lustre, incluso si el nodo de servicio en el que se lanzó el trabajo no es un cliente lustre. Liblustre proporciona un mecanismo para mover datos directamente entre el espacio de aplicación y los OSSS de lustre sin necesidad de realizar una copia de datos a través del núcleo ligero, logrando así una baja latencia, y gran ancho de banda en el acceso directo de los procesadores al sistema de archivos lustre.

3.3.2 Google file system (GFS). Es un sistema de almacenamiento basado en las necesidades de Google diseñado por Sanjay Ghemawat, Howard Gobioff y Shun-Tak Leung y presentado por primera vez en Lake George, Nueva York, octubre de 2003.

Al no ser un sistema de archivos de uso generalista, GFS ha sido diseñado teniendo en cuenta las siguiente premisas: que un componente falle es la norma no la excepción, los archivos son enormes (archivos de muchos Gb son comunes), es muy común que un archivo cambie porque se le añaden datos pero es muy raro que se sobrescriban los datos existentes, el codiseño de las aplicaciones y de la API del sistema de archivos proporciona un beneficio global.

Un clúster GFS consiste en un máster y múltiples Chukserver que dan servicio a múltiples clientes. Cada uno de estos servidores está normalmente implementado sobre una máquina Linux ejecutando el servidor como un proceso a nivel de usuario.

Los archivos están divididos en bloques de tamaño fijo y cada bloque es identificable por un número único (global e inmutable) de 64 bits, llamado Chunk Handle, que es asignado por el máster cuando el bloque se crea. Por fiabilidad, cada bloque está replicado en varios Chunkserver.

El máster es el encargado de mantener todos los metadatos del sistema de archivos. Eso incluye el espacio de nombres, información para el control de acceso, el mapa con la distribución de archivos en los Chunkserver y la localización actual de los trozos. Los clientes interactúan con el Máster para las operaciones de metadatos, pero todas las comunicaciones del traspaso de datos se realizan directamente con los Chunkserver.

3.3.3 Parallel virtual file system (PVFS). Es un sistema de archivos que proporciona un sistema de almacenamiento de alto rendimiento y proporciona características de escalabilidad. Es un sistema de almacenamiento de alto rendimiento en paralelo. PVFS proporciona cuatro importantes capacidades en un solo paquete:

- Espacio de nombres de archivo a través de la máquina
- Acceso transparente para las empresas
- Distribución física de los datos a través de múltiples discos en varios nodos del clúster

- Alto rendimiento de espacio y acceso a las aplicaciones

Para ser utilizado un sistema de archivos en paralelo, se debe proporcionar un espacio de nombres igual para todo grupo PVFS. Es un sistema de archivos en el cual todos los nodos pueden acceder al mismo directorio y al mismo tiempo. Una vez montados los archivos y directorios, se puede operar con todas las herramientas, tales como LS, CP, y RM.

Con el fin de proporcionar un alto rendimiento al momento de acceder a los datos almacenados en el sistema de archivos, PVFS se propaga a través de múltiples nodos del clúster, llamados e/s nodos. Al propagar datos a través de los diversos nodos de e/s, se tienen varios caminos a los datos a través de la red y múltiples discos en los que se almacenan los datos. Esto elimina los cuellos de botella que se ocasionan en un solo camino, por lo cual el potencial total de ancho de banda es elevado.

Con PVFS, los clientes pueden evitar hacer solicitudes al sistema de archivos a través del núcleo de enlace con la API nativa. Esta biblioteca implementa un subconjunto de las operaciones de Unix en la que PVFS se pone en contacto directamente con los servidores en lugar de pasar por el núcleo local. Esta biblioteca puede ser utilizada por las aplicaciones, como la Romio MPI-IO biblioteca, para la alta velocidad de acceso PVFS

4. CLÚSTER DE ALMACENAMIENTO ADMINISTRADO POR EL SISTEMA DE ARCHIVOS PVFS

4.1 ALMACENAMIENTO EN CLÚSTERS

Los sistemas de archivos en Clústers tienen como principal función mostrar los datos almacenados en diferentes nodos como un solo sistema de archivo de alta capacidad. Unos de los objetivos es no tener que hacer uso de ninguna interfaz de programación (api) para tener acceso a los datos almacenados en el clúster. Existen diferentes implementaciones de sistemas de archivos para Clústers, algunos hacen uso de servidor de metadatos y otros no. Los servidores de metadatos pueden convertirse en cuellos de botellas para el desempeño de operaciones intensivas y pueden representar un punto de falla, haciendo que estos sistemas no sean totalmente distribuidos. También se debe decir que estos servidores de metadatos permiten una restauración del sistema más rápida, ya que contienen información del estado de todo el clúster. Sistemas de archivos como PVFS y lustre hacen uso de ellos, y otros sistemas, como Gluster fs, mantienen una posición contraria al uso de servidores de metadatos.

4.2 ARQUITECTURA DE PVFS2

PVFS es un sistema de archivos paralelo escalable y de alto desempeño diseñado para ser utilizado en Clústers de computadoras, en particular por aplicaciones paralelas. Para poder ofrecer un servicio de estas características, PVFS reparte los archivos en varios nodos del clúster. De esta forma elimina cuellos de botella

en los procedimientos de entrada/salida e incrementa el ancho de banda potencial para múltiples clientes.

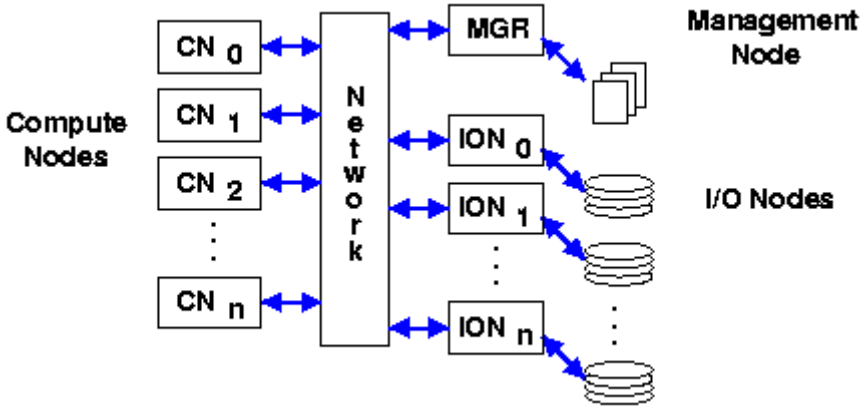
Provee tres servicios básicos a los usuarios:

- Un espacio de nombres consistente entre los nodos del clúster, que permite a los programadores acceder a los archivos desde múltiples nodos.

- Distribución física de los datos entre los discos de los nodos que permite evitar cuellos de botella tanto en la interface del disco como también en la red proveyendo mayor ancho de banda a los recursos de entrada/salida (e/s)

- Interface de e/s que permite que los usuarios controlen cómo serán distribuidos los datos y habilitar modos de acceso.

Figura 5. Diagrama del sistema PVFS



Fuente: OpenSolaris PVFS Porting Project [online] Diagrama del sistema PVFS [citado Agosto 2008]
<URL: http://www.opensolaris.org/os/community/edu/os_contest/pvfs/pvfs1.PNG>

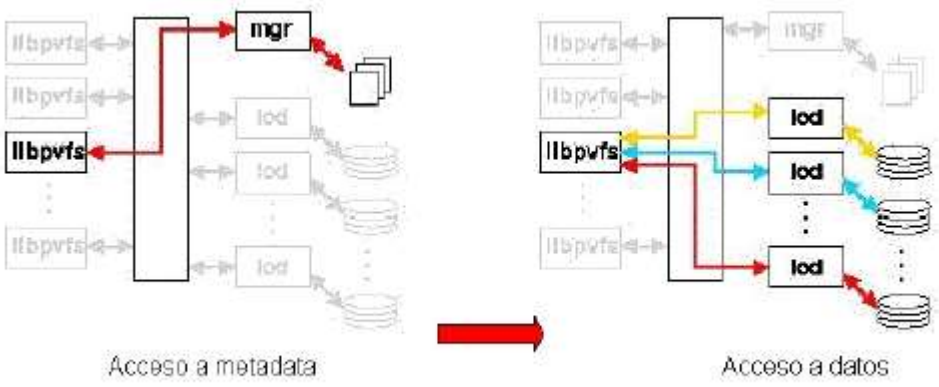
Como se observa en la figura 5 las máquinas que integran el sistema PVFS pueden tomar uno o más de los siguientes roles:

- Servidor Metadatos. Existe un único servidor Metadatos por sistema de archivos PVFS en el cual corre el demonio MGR. Contiene información correspondiente a los archivos y directorios que posee como ser permisos, dueño, ubicación de los datos distribuidos en los servidores de e/s. Es contactado por los clientes cuando necesitan leer un directorio o bien crear, eliminar, abrir o cerrar un archivo.

- Servidor de e/s. Puede haber uno o más. Cada servidor de e/s (IOD) aporta una porción de su disco local para integrar la partición PVFS. Lleva a cabo las operaciones de acceso a los archivos sin intervención del servidor Metadatos.

- Cliente. Puede haber uno o más clientes. En ellos se corren las aplicaciones que acceden a los archivos y directorios de la partición PVFS. Cuando una aplicación desea abrir, cerrar, crear o eliminar un archivo se comunica directamente con el MGR a través de la biblioteca Libpvfs. Una vez que el servidor metadatos localiza el archivo, le devuelve la ubicación a la aplicación. Luego ésta puede utilizar la biblioteca para acceder directamente al servidor de e/s correspondiente para leer o escribir sin necesidad de comunicarse con el MGR.

Figura 6. Flujo de metadatos y datos en PVFS



Fuente: Guía de Innovación tecnológica en Clúster e Grid [online] Flujo de metadatos y datos en PVFS [Agosto 2008], <URL: http://guialivre.governoeletronico.gov.br/guiaonline/guiacluster>

5. MONTAJE DE UN CLÚSTER DE ALMACENAMIENTO UTILIZANDO PVFS

5.1 SISTEMA OPERATIVO

Para llevar a cabo el montaje de un clúster de almacenamiento utilizando PVFS, es indispensable tener en cuenta que se requiere tener instalado el sistema operativo (S.O) DEBIAN.

5.2 CARACTERISTICAS MINIMAS PARA LOS EQUIPOS

Es importante enunciar que para obtener mayor rendimiento en el funcionamiento del clúster de almacenamiento, los equipos deben cumplir mínimos requisitos.

HW Disco duro Mayor o igual de 40GB

Memoria RAM Mayor o igual 512

Procesador Intel Pentium 4 o AMD Sempron o superiores (Dual core, AMD2 etc)

Nota: entre más grandes y mejores las anteriores características, mejor desempeño y desarrollo

5.3 INSTALACIÓN PVFS

5.3.1 Descargar las fuentes

http://www.pvfs.org/pvfs2/download.html

Archivo: pvfs-2.7.1.tar.gz

5.3.2 Dependencias. Los siguientes paquetes son requeridos por pvfs2:

- berkely db (versión 3 o 4)
- aio support (provista por libc6)
- pthreads
- gcc 2.96 o una versión más nueva (recomendado gcc 3.x)
- gnu make

5.3.3 Instalación de librerías. Como root del sistema se ejecuta en consola lo siguiente:

```
# aptitude install libdb4.4-dev libc6 gcc make
```

5.3.4 Recomendación. Recomendación adicional: versión del kernel del sistema operativo.

2.6.0 (o más nuevo)

Esta versión de kernel es necesaria en los clientes, más no en los servidores de archivos

5.3.5 Instalación de pvfs2 en todos los nodos. Después de tener lo necesario se procede a instalar pvfs2 en todos los nodos.

Se copia el paquete al directorio /usr/src y se sitúa en el mismo:

```
# cp pvfs2-2.7.1.tar.gz /usr/src
# cd /usr/src
```

Se descomprime el paquete creando un directorio:

```
# tar -xzf pvfs-2.7.1.tar.gz
```

Se creamos un enlace simbólico a un el directorio pvfs2 y lo se verifica:

```
# ln -s pvfs-2.7.1 pvfs2
# ls -lf
```

5.4 INSTALACIÓN DE PAQUETES

- Se ingresa al directorio creado con el enlace:

  # cd pvfs2

- Se ejecuta el script de auto configuración del paquete para poder compilar

  # ./configure

- Se compila el paquete:

  # make

- Se instala el paquete compilado:

  # make install

5.5 CREAR ARCHIVO DE CONFIGURACIÓN EN SERVIDOR(ES) I/O CON PVFS2-GENCONFIG

Este script permite generar el archivo de configuración global del sistema de archivos, en el cual se especifica:

- El protocolo de red usado a usar en el sistema de archivos.

- El puerto por el que se va a establecer la comunicación.

- La lista de los servidores i/o (el nombre de host de cada servidor).

- El servidor de metadatos (el nombre de host del servidor de metadatos).

- El archivo donde se escribirán los log del sistema en cada servidor.

- El directorio en cada servidor donde se almacenaran los datos en el sistema de archivos.

- Se ejecuta en consola en cada uno de los servidores:

  # pvfs2-genconfig /etc/pvfs2-fs.conf

5.6 ARRANCAR SERVIDORES

Antes de arrancar el servidor por primera vez, se debe hacer con un argumento especial que permite la creación de un nuevo espacio de almacenamiento en caso de que este no exista.

pvfs2-server /etc/pvfs2-fs.conf -f
Para arrancar en modo normal el servidor se ejecuta lo siguiente:
pvfs2-server /etc/pvfs2-fs.conf

5.7 EDITAR INICIO DEL SISTEMA (RC)

Pasos para arranque de servidores. Para que los servidores arranquen desde el inicio el demonio de pvfs2 es necesario hacer algunas modificaciones.

- Se copia el archivo el cual contiene un script para tal fin:

  # cp /usr/src/pvfs2-1.5.1/examples/pvfs2-server.rc /etc/init.d/pvfs2-server

- Se actualiza el runlevel 2 del sistema operativo:

  update-rc.d pvfs2-server start 2

- Para que dicho escript funcione en DEBIAN, es necesario hacer unas modificaciones:

  Comentar con # el script functions
  #. /etc/rc.d/init.d/functions

- Se busca en el archivo la siguiente línea:

  Daemon ${pvfs2server} --pidfile ${pvfs2_pidfile} ${pvfs2_fs_conf} \
  ${pvfs2_server_conf}

- Es reemplazada por:

  start-stop-daemon --start --exec ${pvfs2server} \
  --pidfile ${pvfs2_pidfile} ${pvfs2_fs_conf} ${pvfs2_server_conf}

- Ahora se busca la siguiente línea:

  kill `cat /var/run/pvfs2.pid`

- Y es cambiada por las siguientes dos líneas:

  pid_miprocess=`ps aux | grep pvfs2-server | grep -v grep|awk '{print $2}'`
  kill $pid_miprocess
- Una vez realizadas estas modificaciones se puede iniciar y detener el servidor pvfs2 como cualquier servicio del sistema operativo con start y stop de la siguiente manera:

  # /etc/init.d/pvfs2-server start

  # /etc/init.d/pvfs2-server stop

5.8 CONFIGURACIÓN DEL CLIENTE

Para poder acceder al sistema de archivos pvfs2 es necesario agregar el módulo al kernel del sistema operativo (esto se abordará mas adelante). Además es necesario agregar información que le permita encontrar el sistema de archivos paralelo pvfs2.

Para tal fin agregamos en el archivo de configuración de dispositivos y particiones fstab presente en los sistemas Unix.

- Se ejecuta en consola como root:

gedit /etc/fstab

- Se agrega lo siguiente:

tcp://hostname_de_cada_server:3334/pvfs2-fs /mnt/pvfs2 pvfs2 defaults,noauto 0 0

Debe haber una línea similar por cada servidor (de metadatos o servidores i/o), cambiando el hostname por el de cada servidor. Se debe especificar el punto de montaje en el cliente (en este caso /mnt/pvfs2), el tipo de sistema de archivos (pvfs2) y algunas configuraciones por omisión en cuanto a permisos.

- Se debe crear un archivo similar al fstab propio de pvfs2, el cual será usado por algunas librerías para localizar el sistema de archivos. para tal fin se ejecuta lo siguiente en consola:

mkdir /mnt/pvfs2

- El anterior comando crea un directorio el cual va a ser el punto de montaje del sistema de archivos. ahora se crea el archivo pvfs2tab

touch /etc/pvfs2tab

- Y se le agrega permiso de lectura

chmod a+r /etc/pvfs2tab

5.9 PROBAR LA INSTALACIÓN

Para probar la configuración hasta ahora hecha, en el cliente se puede ejecutar lo siguiente:

pvfs2-ping -m /mnt/pvfs2

Después del -m se debe colocar el punto de montaje del sistema de archivos.

El anterior comando verifica que todo haya sido correctamente configurado.

5.10 COMPILAR MÓDULO CON SOURCES DEL KERNEL

5.10.1 Compilar módulo del Kernel

- Para compilar un módulo del kernel se necesita tener instalados las fuentes del kernel (kernel-sources) de la versión del kernel que se tenga instalado en el sistema operativo, para esto se hace lo siguiente:

  # cat /proc/version

- Linux versión 2.6.18-6-686 (DEBIAN 2.6.18.dfsg.1-22etch2) (dannf@debian.org) (gcc versión 4.1.2 20061115 (prerelease) (DEBIAN 4.1.1-21))

  # aptitud search linux-source

- p linux-source-2.6.18 linux kernel source for version 2.6.18 with DEBIAN.

  # aptitud install linux-source-2.6.18

- Los sources del Kernel quedan instalados en las siguientes carpetas dependiendo la distribución:

  DEBIAN (/usr/src/linux-source 2.6.xx-xx-arch/)

  Fedora(/usr/src/redhat/build/kernel-2.6.23/linux-2.6.xx.arch/)

- Se configuran los sources del kernel con la configuración que ya se hizo durante la instalación del sistema operativo, seguido se preparan las fuentes y se compilan.
  #cp /boot/config-2.6.xx-x-686 .config

- #make silentoldconfig responde por defecto todas las preguntas basándose en su archivo existente .config, además agrupa en pantalla preguntas ya respondidas

- #make scripts en la preparación del kernel se crearán algunos scripts.

- #make prepare para preparar el kernel, anteriormente compilado.

Ahora el Kernel ya se encuentra listo para agregarle el módulo del cliente de PVFS en /usr/src/pvfs2. Se hace lo siguiente para compilar el módulo con los sources del kernel.# cd /usr/src/pvfs2

./configure –with-kernel=/usr/src/linux-2.6.##.#

#make && make install && make kmod && make kmod_install && depmod -a && make install_doc se inserta el módulo compilado en el kernel asi:

insmod /usr/src/pvfs2/src/kernel/linux-2.6/pvfs2.ko

- Para verificar se puede hacer con el comando lsmod

  # lsmod |grep pvfs

  pvfs2 98932

5.10.2 Anotación para antes de montar el pvfs. En este punto se debe arrancar el servidor y el cliente pvfs2 antes de poder montar el sistema de archivos pvfs2. La aplicación cliente de pvfs2 consiste en dos programas llamados pvfs2-client-core y pvfs2-client. No se debe correr pvfs2-client-core por si solo y pvfs2-client no puede iniciarse sin antes haberse iniciado el servidor.

Por ejemplo:

cd src/apps/kernel/linux-2.6/
./pvfs2-client -f -p ./pvfs2-client-core
pvfs2-client starting

spawning new child process

about to exec ./pvfs2-client-core

waiting on child with pid 17731

Con el -f pvfs2 mantiene la aplicación cliente ejecutándose en primer plano; el argumento -p es necesario a menos que el pvfs2-client-core esté instalado y se puede encontrar en su path.

5.11 PARA MONTAR PVFS2

1) #mkdir /mnt/pvfs2 (si no se ha creado previamente)

2) #pvfs2-client starting o pvfs2-client -f pvfs2-client-core (si es la primera ves)

3) #mount -t pvfs2 tcp://node2:3334/pvfs2-fs /mnt/pvfs2 (nede2: el servidor de metadatos)

4) #mount | grep pvfs

tcp://lain.mcs.anl.gov:3334/pvfs2-fs on /tmp/mnt type pvfs2 (rw)

5) Para verificar su funcionamiento básico:

df -h

tcp://node2:3334/pvfs2-fs 38g 0 38g 0% /mnt/pvfs2-client

5.12 PARA MATAR LOS DEMONIOS PVFS2

- Para terminar

  # killall pvfs2-client

- Para desmontar el módulo de PVFS del kernel

  # rmmod pvfs2

6 HERRAMIENTAS PARA PROBAR UN CLUSTER DE ALMACENAMIENTO

6.1 IOZONE

Es una herramienta software que permite obtener Benchmarks de rendimiento de e/s sobre archivos para operaciones como lectura, escritura, relectura, reescritura, lectura randómica y otras más. Uno de los usos de IOZONE es el de poder expresar el rendimiento de e/s en función del número de procesos.

En general hay tres modos de uso:

1. Regular

2. Throughput (rendimiento)

3. Clúster

El modo regular utiliza un único hilo para llevar a cabo los experimentos. Es el modo por defecto. El modo Throughput (rendimiento) permite al usuario especificar el número de hilos (en una sola máquina) con los que se va a ejecutar el experimento. El modo de clúster utiliza un archivo de configuración especial para permitir a los clientes múltiples experimentos. Ejemplos de uso:

iozone –a

Ejecución en modo automático que realiza todos los test, con bloques de escritura y lectura variando de 4kb a 16 Mb para archivos de 64kb a 512mb. Toma bastante tiempo.

iozone -ra -y 10k -g 64m

Ejecución en modo automático que realiza todos los test, con bloques de escritura y lectura de mínimo 10kb, para archivos de máximo 64mb. Con -r se genera una salida tipo excel.

iozone -s 4096

6.1.1 Nomenclatura IOZONE

- -s indica el tamaño del archivo a escribir en kb, en este caso 4096 kb.
  # iozone -t 1 -r 4m -s 100m -i 0 -i 1

- -t indica que se está ejecutando en modo Throughput. El numero 1 indica que se está ejecutando con un hilo.
- -r indica el tamaño de bloque con que se escribirá en el experimento en unidades de 4 megabytes.

- -s indica el tamaño del archivo a escribir en el experimento.

- -i indica los test que se van a realizar, en este caso solo los dos primeros tests (escritura, lectura).

6.2 DD

Con el comando DD se puede crear archivos de diferentes tamaños, leer dichos archivos y medir el tiempo de dichas operaciones. Algunos ejemplos son:

- Crear un archivo de bloques de 1mb de tamaño total 256mb

  # time dd if=/dev/zero of=file bs=1m count=256

- Leer un archivo de bloques de tamaño de 1mb de tamaño total 256mb

  # time dd if=file of=/dev/null bs=1m count=256

6.3 BONNIE++

Es un programa para poner a prueba los discos duros y sistemas de archivo en las máquinas de cómputo. BONNIE++ prueba algunos aspectos importantes en cada ensayo y da un resultado de la cantidad de trabajo realizado por segundo y el porcentaje de tiempo de CPU usado.

6.3.1 Nomenclatura BONNIE++

- -d directorio a utilizar para las pruebas.

- -s tamaño del archivo (s) de IO medido en la ejecución en megabytes. Si el tamaño es mayor que 1Gb entonces múltiples archivos y este será el que se utiliza para almacenar los datos, y cada uno de los archivos será hasta 1Gg de tamaño. Puede especificar el tamaño en gigabytes o el fragmento de tamaño en kilo-octetos si añade 'g' o 'k' para el final de la serie, respectivamente. si el tamaño especificado es 0 entonces esta prueba se evitará.

- -n el número de archivos de prueba que se van a crear. Esto se mide en múltiplos de 1.024 Mb. Esto se debe a que nadie va a querer prueba de menos de 1.024 Mb y se necesita el espacio adicional en el disco.

El valor por defecto para esta prueba es probar con los archivos de 0 bytes. Para utilizar los archivos de otros tamaños puede especificar el número de directorios donde el “máx” es el tamaño máximo y “mín” es el tamaño mínimo (por defecto es 0 si no se especifica). Si el mínimo y el máximo se especifican a continuación, todos los archivos tendrán azar un tamaño de la gama mínimo-máximo. Si se especifica un número de directorios a continuación, los archivos se distribuyen uniformemente entre los que muchos sub-directorios. si el número es 0 entonces esta prueba se evitará.

- -r el tamaño en megabytes. si se especifica este los demás parámetros se controlarán para asegurarse de que tiene sentido para una máquina que gran parte de la RAM. usted no necesita hacer este uso, en general, como debería ser capaz de descubrir el tamaño de RAM. nota: si especifica un tamaño de 0 a continuación, todos los controles serán discapacitados.

- -x número de ensayos de funcionamiento. Esto es útil si usted desea llevar a cabo más de una prueba. Se volcado de salida continua en formato CSV o bien hasta que el número de pruebas se han completado, o está muerto.

- -u id de usuario para su uso.

- -q el modo silencioso. Si se especifica a continuación, algunos de los mensajes informativos extra será suprimido.

- -f modo rápido, salta por car-IO pruebas.

- -b no escribir Buffering. fsync () después de cada escritura.

- -p número de procesos para servir de semáforos.

- -y semáforo para esperar antes de cada prueba.

Ejemplos

bonnie++ -d /mnt/test/ -n 0 -u 0 -r 2048 -s 81920 -f -b

bonnie++ -s 16386 -n 1:4096:4096:1024 -d /mnt/test/ -u root

6.4 CLUSTER CSSH (CSSH)

Es una herramienta para los administradores de sistemas. A CSSH se le indica el nombre o IP de máquinas que se van a trabajar en paralelo. Abre una sesión CSSH con cada una de ellas en una ventana independiente. Se puede trabajar sobre cada una de ellas de forma individual o con todas las maquinas a la vez.

Es útil cuando se trabaja con granjas de servidores, clústers. Replica cada pulsación de teclas, se puede incluso abrir un vi en todas las sesiones a la vez, moverse por el archivo y editarlo en todos los servidores al mismo tiempo.

Ejemplo

```
#cssh pv1, pv2, pv3, pv4, pv5
```

7. CONCLUSIONES Y RECOMENDACIONES

El almacenamiento de alto rendimiento es una excelente alternativa para aprovechar los recursos de máquinas y redes de datos para aumentar la capacidad de almacenamiento, con el propósito de resolver problemas complejos que demandan este potencial.

Como resultado del proyecto realizado se generaron las primeras bases y recomendaciones para el establecimiento, configuración y operación de almacenamiento masivo administrado por sistemas de archivos, fundamentales en el proceso de aprovechamiento de discos para grandes necesidades de almacenamientos.

Una arquitectura de clúster de almacenamiento requiere de un sistema operativo robusto, multiproceso, multiusuario y con características deseables de facilidad de uso; se requiere además de un sistema de archivos que actúe entre el sistema operativo y las aplicaciones con la finalidad de proveer al clúster una interfaz única de acceso al sistema de archivos (PVFS), herramientas de optimización y mantenimiento del sistema y escalabilidad. En el desarrollo de este proyecto se utilizó la versión de DEBIAN ETCH 4.0 y el Sistema de archivos Parallel Virtual File System (PVFS)

A través de la experiencia del presente proyecto, no era viable ni factible realizar una grid de datos ya que para que esta sea funcional se debe antes haber implementado una grid computacional, pues la grid de datos es una pequeña parte de una grid computacional. Por ende para cumplir el objetivo de montar la grid de

datos se debe primero montar una grid computacional, la cual consume mucho recurso humano, tecnológico y de tiempo. Además que la Grid computacional y la grid de datos, actualmente en Colombia se encuentran en una fase de investigación y de prueba funcional, debido a que es una tecnología muy reciente en el país. Por lo cual se hizo necesario buscar un camino diferente para cumplir con el objetivo de conseguir un almacenamiento masivo mediante arquitecturas computacionales.

Con el estudio realizado a los sistemas de almacenamiento de alto rendimiento se encontró que existen sistemas de almacenamiento paralelo, distribuido y tolerante a fallos y que cada uno de ellos cuenta con diferentes características funcionales propias y sistemas de archivos que se acomodan a dicha arquitectura.

Con el estudio sobre los sistemas de archivos para clúster que existen en la actualidad y de los proyectos que se han llevado a cabo entre los diferentes tipos de sistemas de archivos, se llega a la conclusión que el sistema de archivos PVFS reúne características apropiadas para la solución del objetivo de manejar almacenamiento masivo en un clúster, tales como, la escalabilidad, la flexibilidad, la compatibilidad con otras aplicaciones, el rendimiento de operaciones de E/S, la coexistencia con otros sistemas de archivos entre otras.

Desde el estudio realizado en el PVFS se demuestra que se puede trabajar sin servidor de archivos. ya que al utilizar el sistema de archivos en paralelo PVFS se elimina la necesidad del servidor de archivos el cual se puede constituir en un cuello de botella para algunas soluciones y se reemplaza con un servidor de metadatos haciendo que se reduzca el tiempo de e/s, obteniendo un mayor rendimiento y escalabilidad a la hora de almacenar información.

BIBLIOGRAFÍA

Egee enabling grids for e-science, proyecto egee, [citado 21/08/2007], disponible internet <url: http://public.eu-egee.org/intro/>.

Foster ian, what is the grid? a three point checklist [online], 2 ed., argonne national laboratory & university of chicago, julio 20, 2002.[citado 16/09/2008], disponible en internet <URL: http://www-fp.mcs.anl.gov /~foster/ articles/whatisthegrid.pdf >.

Grid Colombia, orígenes, última actualización julio 25, 2006, [citado 7/09/2008], <URL: http://urania.udea.edu.co/grid-colombia/origin.php>.

Scalability and High Performance, by Design [online], PVFS Parallel Virtual File System [citado Agosto de 2008], <URL: http://www.pvfs.org >.

PVFS: A Parallel File System for Linux Clusters[online], Philip H. Carns Walter B. Ligon III, *Parallel Architecture Research Laboratory, Clemson University, Clemson, USA. <URL:* http://www.parl.clemson.edu/pvfs/el2000/extreme2000.html>

Glite. Lightweight Middleware for Grid Computing, [online], EGEE, Enambling Grids for E-science. [citado Agosto 2008] < URL: http://glite.web.cern.ch/glite/>.

Wikipedia, Enciclopedia Libre [online], Almacenamiento Distribuido [citado Agosto 2008],<URL: http://es.wikipedia.org/wiki/almacenamiento_distribuido>

Monografias.com [online], Arquitectura Paralela, [citado Julio 2008] <URL: http://www.monografias.com/trabajos16/arquitectura-paralela/arquitectura-paralela.shtml>

MKM Publicaciones [online], Diario Informático, El cambio de paradigma hacia el almacenamiento en clústeres [citado marzo 2008],<URL: http://www.mkm-pi.com/mkmpi.php?article1040>

Departamento de informática, Universidad de Valladolid [online] Tolerancia de fallos [citado Marzo 2008], http://www.infor.uva.es/~bastida/arquitecturas%20avanzadas/tolerant.pdf

Wikipedia, Enciclopedia Libre [online], Sistema de archivos. [Citado Agosto 2008] <URL: http://es.wikipedia.org/wiki/sistema_de_archivos>

Red Had Magazine [online], Red Hat Global File System, The open source cluster file system for enterprise deployments [citado Agosto 2008] <URL: http://www.redhat.com/gfs/>

Wikipedia, Enciclopedia Libre [online], Coda (sistema de archivos) [citado Agosto 2008] <URL: http://es.wikipedia.org/wiki/coda_(sistema_de_archivos)>

Gluster Non-stop clustered Storage [online] GlusterFS [citado Julio 2008] <URL: http://www.gluster.org/docs/index.php/glusterfs>

Center for Computational Sciences [online] Computación Paralela [citado mayo 2008] <URL: http://www.ccs.tsukuba.ac.jp/ccs/exreview2007/file/comp_tatebe.pdf>

Wikipedia, Enciclopedia Libre [online], Lustre (Sistema de Archivos) [citado Julio 2008] <URL: http://es.wikipedia.org/wiki/lustre_(sistema_de_archivos)>