PVFS (Parallel Virtual File System)

Sergio González González
Instituto Politécnico de Bragança, Portugal

sergio.gonzalez@hispalinux.es

Jónatan Grandmontagne García
Universidad de Bragança, Portugal

thestalker44@hotmail.com

Breve explicación del sistema PVFS, en qué consiste y sus características.

1. Introducción

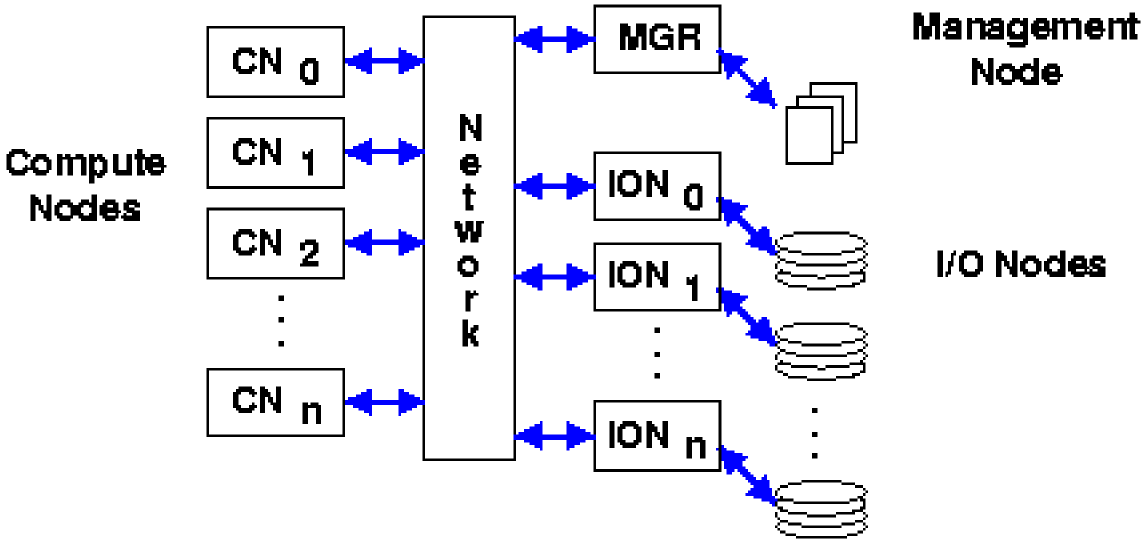
PVMS intenta proveer de un sistema de archivos en red distribuído de alta eficiencia y escalable, normalmente utilizado en entornos de clustering. PVFS es un proyecto de Software Libre que no requiere hardware especial o modificiciones en el núcleo para que funcione. Las características de este sistema de archivos distribuído son:

- Un sistema consistente de nombres
- Acceso transparente para las utilidades existentes (ls, cd, etc.)
- Distribución física de los datos a través de múltiples discos en distintos nodos
- Alto rendimiento en espacio de para las aplicaciones

PVFS provee un mismo espacio de nombre para todo el cluster y es accesible por las utilidades habituales. PVFS se monta en todos los nodos y en el mismo directorio simultáneamente, permitiendo el acceso simultáneo a todos los ficheros del sistema PVFS, a través del mismo esquema de directorios. Una vez que el sistema está montado, podremos trabajar con las herramientas típicas, como ls, cp y rm

Para conseguir un alto rendimiento en el acceso a los datos concurrentemente, PVFS distribuye los datos en múltiples nodos del cluster, denominados *I/O nodes*. Distribuyendo los datos en múltiples nodos, los clientes poseen diferentes rutas hacia los datos, eliminado de esta forma los cuellos de botella (bottlenecks) y mejorando o aumentando el ancho de banda para múltimples clientes.

PVFS permite prescindir de las llamadas al kernel en los accesos al sistema de archivos, gracias al uso de una API nativa. Esta implementa un subconjunto de operaciones UNIX que permiten contactar directamente con los servidores PVFS.



Vista del sistema PVFS

La imagen superior muestra como se asignan los nodos para el uso de PVFS. Estos son divididos en nodos de computación (compute nodes) donde se ejecutan las aplicaciones, y los nodos de gestión que manejan las operaciones con los metadatos y los nodos de entrada/salida (I/O) que almacenan la información. Los nodos de administración y entrada/salida también pueden ser utilizados como nodos de computación.

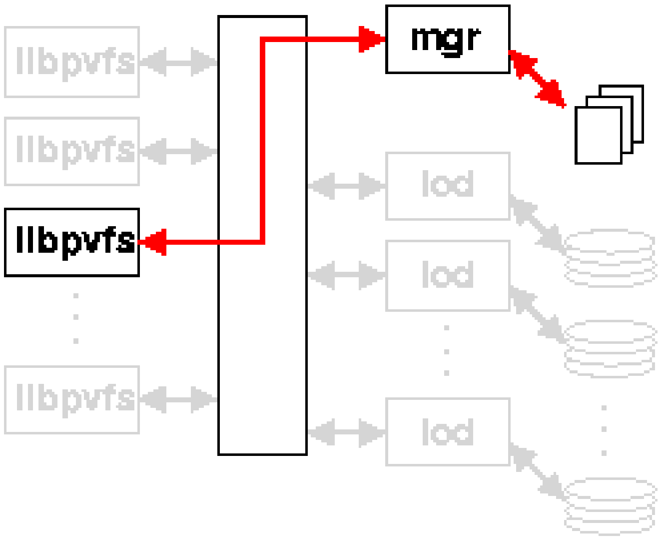
2. Componentes PVFS

Hay cuatro grandes componentes, que son:

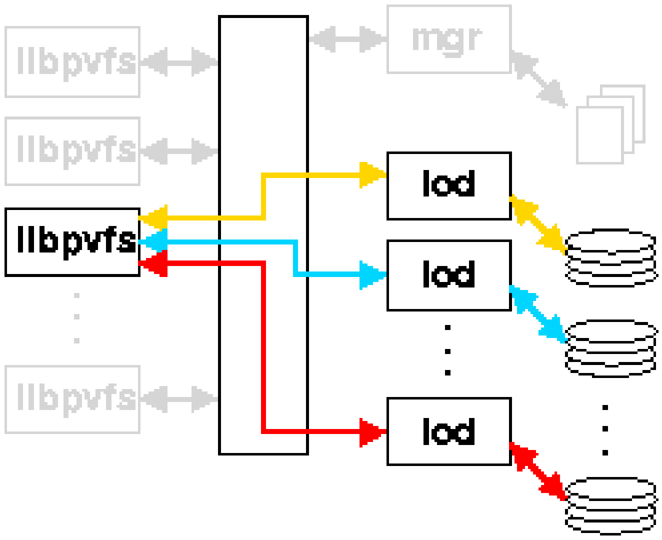
- Servidor de metadatos (mgr)
- Servidor de E/S (I/O) (iod)
- API nativa de PVFS (libpvfs)
- Soporte en el kernel de PVFS

Los dos primeros componentes son demonios que se ejecutan en los nodos del cluster. El servidor de metadatos (mgr) gestiona los metadatos de todos los ficheros. El uso de los demonios que operan automáticamente con los metadatos, eliminan algunas de las deficiencias de otras soluciones de almacenamiento en red, las cuales tienen que implementar complejos esquemas para asegurar la consistencia en los metadatos.

El segundo demonio es el servidor de E/S (I/O) (iod). Este gestiona el almacenamiento y recuperación de los datos almacenados en el disco local del nodo. Estos servidores crean los ficheros en el sistema de archivos existente en el disco local del nodo, utilizando las llamadas *read(), write() y nmap()* para el acceso a esos archivos. Esto significa que puedes utilizar cualquier sistema de archivos local para almacenar los datos: ext2, ext3, reiserfs, RAID.



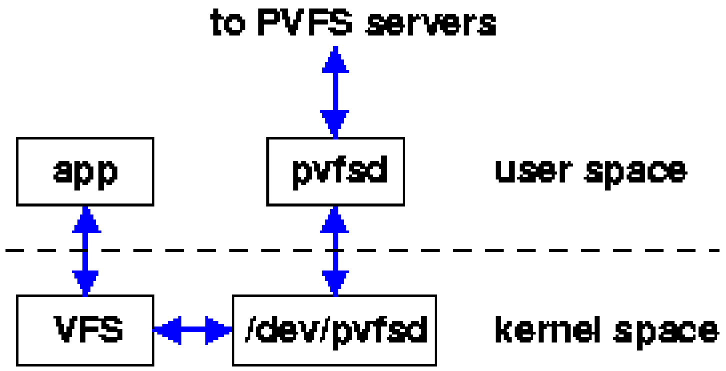
Acceso a metadatos



Acceso a datos

La API nativa de PVFS proporciona acceso en espacio de usuario a los servidores PVFS. Esta librería maneja las operaciones necesarias para mover datos entre los buffers de usuario y los servidores PVFS, manteniendo las operaciones transparentes al usuario. Los gráficos anteriores muestran el flujo de datos en el sistema PVFS para las operaciones con metadatos (arriba) y el acceso a los datos (abajo). Para las operaciones con metadatos, las aplicaciones se comunican mediante la librería con el servidor de metadatos. Cuando se accede a los datos, el servidor de metadatos se elimina de la ruta de acceso y se contacta con los servidores de E/S.

Finalmente, el soporte PVFS para el kernel Linux provee la funcionalidad para montar sistemas PVFS en los nodos Linux. Esto permite a los programas existentes acceder a los datos almacenados en PVFS sin modificaciones.



Flujo de datos a través del Kernel

La figura anterior muestra el flujo de datos a través del kernel, cuando el soporte del núcleo está activo.

3. Interfaces de aplicacion

Para que cualquier sistema pueda utilizar PVFS, existen distintas interfaces de acceso. Estas son:

- API nativa de PVFS
- Interfaz para el núcleo Linux
- Interfaz ROMIO MPI-IO

La API nativa de PVFS provee una interfaz similar a UNIX para el acceso a los archivos almacenados en PVFS.

La interfaz para el núcleo Linux permite a las aplicaciones acceder a los datos de la forma tradicional.

ROMIO implementó las llamadas MPI2 I/O en una librería portable. Esto permite a los programadores de aplicaciones paralelas que utilizan MPI, el acceso a los datos de PVFS gracias a la interface MPI-IO.

4. Licencia de este documento

Bibliografía

Documentación sobre clusters

[PVFS HomePage (http://www.pvfs.org/)]

[Experiences with the Parallel Virtual File System (PVFS) in Linux Clusters (http://www.linuxclustersinstitute.org/Linux-HPC-Revolution/Archive/PDF02/13-Milfeld_K.pdf)] Kent Milfeld, Avijit Purkayastha, Chona Guiang.

[Beowulf PVFS (http://www.nas.nasa.gov/SC2000/GSFC/beowulf_pic2.html)] Ryan Spaulding.

<rspaulding@mail.arc.nasa.gov>