Sistema Beowulf de alto rendimiento, bajo precio y fácil operación.

Daniel Brubeck Salcedo, Rogelio Quintero Razo,
Flavio Reyes.

25 de noviembre de 2008

Resumen

Se está convirtiendo en una actividad cotidiana ejecutar algoritmos que requieren un gran poder de computo el cuál una sola computadora no puede proporcionar. Para proveer al usuario de este poder se necesita que varias computadoras trabajen en conjunto, el cuál da lugar al computo en paralelo o clustering. Lo importante de este articulo es que el usuario pueda configurar, administrar, programar y parametrizar un clúster tipo Beowulf para la resolución de algoritmos o tareas de una manera rápida y eficiente.

1. Introducción

Las aplicaciones de hoy en día como el procesamiento de imágenes, rendering, simulación de distintos tipos, agentes, indexación, motores de búsqueda, servidores web y de aplicaciones, tienden a consumir mayores cantidades de recursos de procesamiento, almacenamiento y transferencia de datos que una sola computadora sería incapaz de brindar.

El cómputo de alto rendimiento requiere de equipos especializados, costosos y poco accesibles en comparación de los de uso común. Estos costos inlcluyen servicios adicionales como mantenimiento y operación.

Un clúster es un conjunto de computadoras interconectadas entre sí, que simulan ser una sola supercomputadora a través de sistemas de mensajería entre los componententes del clúster.

Actualmente existen dos maneras de realizar computo de alto rendimiento: clústers especializados de fabricantes y tipo Beowulf.[1]

Los clústers especializados de fabricantes ofrecen un alto rendimiento pero a la vez son costosos y son los que ocupan los primeros lugares en desempeo dentro de la lista top 500.[9]

Con respecto al proyecto esta opción no es viable debido a los recursos económicos que representan factores como el costo del equipo, mantenimiento y operación.

La alternativa para proveer computo de alto rendimiento para las caracteristicas del proyecto y el Laboratorio de Telemática es la de un clúster Beowulf, para aprovechar las capacidades actuales con las que cuenta una computadora personal común y paralelizar una tarea entre varias computadoras para que en conjunto se pueda llegar o superar un mismo resultado obtenido mediante una computadora o supercomputadora especializada a un bajo costo.

2. Especificaciones del proyecto

El laboratorio de Telemática de la Universidad del Caribe cuenta con un servidor y 30 computadoras. Estas computadoras pueden aprovecharse para ejecutar algoritmos que requieran de mucho tiempo de ejecución y un gran poder de cómputo. Se podrían poner estas computadoras en modalidad de un cluster para lograr un alto rendimiento con super cómputo paralelo. En este caso específico, utilizando el Laboratorio de Telemática de la Universidad del Caribe para generar un desarrollo en el tema del supercomputo paralelo. Para no afectar las configuraciones de las computadoras que requieren los alumnos y permitir los sistemas de archivos de los nodos se almacenarán en el servidor y se montarán vía red.

Tomando en cuenta las condiciones y características del equipo de cómputo con el que se cuenta en el laboratorio, se determinó la modalidad Beowulf debido a las siguientes características:

- Bajo costo.
- Servidor existente (Linux), equipo de computo no especializado, hardware casi identico.
- Especificaciones en memoria RAM y disco duro comunes.
- Boot via red (PXE).
- Infraestructura de red existente (routers, switchs, etc.).
- Libertad de programación, y sin dependencia de parches o código.
- El nodo maestro no necesariemente es mucho más poderoso que los esclavos.

Arquitectura del clúster Beowulf.

Un clúster en modalidad Beowulf es un clúster de computadoras interconectadas con las siguientes características:

- Esta compuesto por un nodo maestro y dos o más nodos esclavos.
- El nodo maestro es la puerta de salida hacia el mundo exterior de la red.
- El nodo maestro está encargado de coordinar los trabajos entre nodos.
- Los nodos del clúster están dedicados al clúster y no sirven para otros propósitos.
- La red en la cual residen los nodos está dedicada al clúster.
- Todo sistema operativo normalmente es del tipo POSIX y al igual que el sistema operativo, el software que se utilice en el clúster tiene que ser libre y de código abierto.
- Los componentes del clúster no son productos especializados.
- Los componentes del clúster son generalmente idénticos.
- El clúster resultante es usado para cómputo de alto rendimiento.

Protocolos de comunicación.

Para que los nodos se puedan comunicar entre ellos necesitan de protocolos de comunicación. Estos protocolos utilizan métodos de acceso remoto como RSH y SSH para el paso de mensajes. Los dos tipos de mensajería que se utilizan en un clúster Beowulf son:

1 LAM/MPI (Local Area Multicomputer/Message Passing Interface): Es una API que permite el desarrollo de programas de alto nivel con rutinas y datos para llevar la comunicación entre los nodos, así como la coordinación del paso de mensajes. Viene con bibliotecas para los lenguajes Fortran, C y C++.

2 PVM (Parallel Virtual Machine): Es un framework para desarrollar programas paralelos de forma eficiente, permite ver una colección de computadoras como una sola computadora paralela virtual.

3 Rocks: Es una distribución Linux de código abierto que facilita la construcción de clusters a los usuarios finales.

4 OpenMosix: Es una extensión del kernel de Linux para clustering que convierte una red ordinaria de computadoras en una supercomputadora.

Algunas de las condicionantes de administración de un clúster:

- Rocks; disco duro y ram especificos, no permite usar el método diskless y multicore.
- OpenMosix; es un kernel patch con herramientas de administración, multicore.
- PVM; mayor código en la programación, multicore y menor cantidad de funciones.

3. Solución

Las máquinas del laboratorio de Telemática se van a configurar para cargar su sistema operativo a través de un sistema de archivos para facilitar la incorporación de futuros nodos al clúster.

Por lo que al servidor del laboratorio de Telemática se le instalaron o configuraron servicios que no se necesitan en los clientes, así mismo los clientes comparten algunas carpetas de sus sistema de archivos para facilitar la administración de los nodos (no se tiene que instalar una herramienta en cada nodo, solo se necesita hacer una vez en un nodo).

Se escogió la tecnología PXE para arrancar los sitemas de archivos por red de los clientes por lo que el servidor debe de ser capaz de asignar una direccion IP a los clientes y pasarle su archivo de configuración PXE.

Configuraciones del Servidor.

DHCP (Dynamic Host Configuration Protocol).

Archivo de configuración: /etc/dhcpd.conf

El servidor es el encargado de asignar direcciones IP a los nodos del laboratorio, se configuró para entregar una dirección específica a cada uno de los nodos con el correspondiente archivo pxe para tener bien identificados y facilitar la administración de los nodos.[11]

Para agregar en la configuración del DHCP para entregar el archivo se hace de la siguiente manera:

filename "pxelunix.0";

PXE (Preboot eXecution Environment).

Cuando se levanta el servidor tftp, se proporcionan los parámetros de donde se va a obtener el loader del sistema, este loader necesita los archivos para ser enviados por el entorno PXE, estos archivos se encuentran en la carpeta pxelinux.cfg y su nombre tiene una estructura (01-MAC-ADDRESS) para poder enviar la imagen de Linux y cargarla en los nodos correspondientes.

Archivo PXE de nuestro servidor:

/srv/diskless/pxelinux.0

Configuración y parámetros de cada archivo PXE correspondiente a cada nodo.[4]

TFTP (Trivial File Transfer Protocol).

Se escogió el protocolo TFTP para enviar el sistema de archivos y el archivo de configuración PXE por red debido a que es mas ligero y rápido que utilizando el protocolo FTP, este servicio se levanta a través de xinet por lo que el archivo que se tiene que configurar es el archivo de configuración de xinet.conf.

Exports (/etc/exports).

Este archivo le indica al servidor que es lo que es lo que va a exportar para los nodos correspondientes del clúster.
Archivo de configuración /etc/exports.[3]

Hosts.

Archivo de configuración: /etc/hosts.[2]

Este archivo se tiene que configurar tanto en el servidor como en los nodos, se encarga de ligar los nombres (hostname) de las computadoras con la dirección IP. Cada que se agrega un nuevo nodo se tiene que agregar a este archivo con su dirección y nombre (hostname) correspondiente.

SSH.

Para la correcta operación del clúster y el paso de mensajes del mismo, es necesario configurar SSH en modo password-less (basado en llaves).
Para esto necesitamos realizar lo siguiente:
Ejecutar el comando:

ssh-keygen -t rsa.

Después de realizar esto, debemos de copiar el contenido del archivo id_rsa.pub en el archivo authorized_keys en cada uno de los nodos. Este archivo se encuentra oculto (ls -a) en la carpeta .ssh del path de home del usuario. Al ejecutar el comando en los nodos, copiar el contenido de id_rsa.pub en el archivo known_hosts del servidor.

Configuraciones de los nodos.

Sistema de archivos de los nodos

Los nodos van a estar utilizando Fedora Core 4 como sistema operativo, por lo que el sistema de archivos creado tiene que tener la misma estructura que se utiliza en linux. A pesar de que cada nodo tiene sus carpetas de configuraciones individuales, las carpetas que comparten los archivos binarios y ejecutables se comparten entre todos los nodos, de esta forma se facilita la gestión de programas que se instalen en los nodos por que una vez instalado en un nodo se puede utilizar en todos, salvo que necesite de un archivo de configuración específico en una de las carpetas que no están compartidas.

Las carpetas del sistema de archivos que se comparten entre todos los nodos son las siguientes: /bin, /sbin, /lib, / .

FSTAB (File System Table).

Archivo de configuración: /etc/fstab.[10]

Este archivo se pone para cada nodo cliente, debido a que el nodo esclavo no va a contar con un disco duro, debemos indicarle al nodo esclavo que carpetas son las que tiene que montar, de donde y con que permisos. Este archivo contiene las carpetas que necesita un sistema Linux básico para funcionar de forma correcta.

Servicios

Como el sistema de archivos se encuentra montado y no en el disco duro del nodo, al momento de apagar la computadora se tiene que hacer que el sistema tenga algunas consideraciones especiales y no debe de detener los servicios de red. Para esto se modificaron los archivos encargados de detener los demonios de red y nfs.

Archivo de configuración: /etc/rc.d/init.d/network

En el archivo de configuración se tienen las acciones para inicio o detención del servicio de red. Lo que se tiene que hacer en el cliente para que pueda apagar correctamente se tiene que configurar el archivo para evitar que la tarjeta de red se desactive antes de terminar de desmontar todo lo demás.

El archivo de configuración evita que se detenga el servicio y envía un mensaje de que no puede detenerse de la sigiuente forma:

Archivo de configuración: /etc/rc.d/init.d/nfs

Al igual que con el archivo network, este archivo debe de evitar que se detenga el servicio cuando se está apagando los nodos. Para esto queda configurado de forma similar al anterior.

Red.

Archivos de configuración: /etc/sysconfig/network-scripts/ifcfgeth0,
/etc/sysconfig/network

Estos archivos se encargan de entregar los parámetros iniciales sobre la red una vez montado el sistema, básicamente se tiene que indicar mediante estos archivos que va a recibir su dirección IP mediante DHCP, su dirección MAC, el tipo de red y si está activa durante el inicio del sistema.

Configuración de ifcfgeth0.[8]

Configuración de network.[5]

Hosts.

Archivo de configuración: /etc/hosts.

Este archivo se tiene que configurar tanto en el servidor como en los nodos, se encarga de ligar los nombres (hostname) de las computadoras con la dirección IP. Cada que se agrega un nuevo nodo se tiene que agregar a este archivo con su dirección y nombre (hostname) correspondiente.

Instalación de LAM/MPI.

Comandos necesarios para realizar la configuración e instalación:

tar -zxvf lam-7.1.4.tar.gz

Dentro de la carpeta creada ejecutar los siguientes comandos:

./configure –without-fc //sin soporte para Fortran (opcional)
make
make install

Un paso importante para utilizar en LAM/MPI SSH es el indicarlo en la variable de entorno LAMRSH. Esto debido a que LAM/MPI usa RSH por default para la comunicación. Esto se realiza de la siguiente manera:

export LAMRSH="ssh -x"

Archivo hostfile

En este archivo es donde se especifican los nodos existentes (boot schema) y en el cual tambien se pueden especificar la cantidad de cpu's por cada nodo (se pueden indicar de dos maneras: repitiendo el nodo la cantidad de veces según el numero de cpu's o tambien indicandolo por número, ejemplo: cpu=2). Este indicativo de la cantidad de cpu's no tiene ninguna relación con el hardware o cores que contenga cada procesador, sino que más bien es un indicativo para LAM/MPI de cuantos procesos MPI se pueden ejecutar en ese nodo.[6]

Iniciar LAM/MPI:
Lamboot.

Para iniciar con el comando lamboot el entorno de LAM, se tienen que cumplir ciertas especificaciones.[7]
[8](mpi)

lamboot -vv hostfile //vv nivel de verbose

Mpirun.

Es el comando utilizado para el control de la ejecución del programa en paralelo.

mpirun C -npty programa //cantidad total de cpu's en el clúster
mpirun N -npty programa //cantidad total de nodos del clúster.

Algo muy importante a mencionar es la opción "npty", que es la que permite imprimir a los nodos en la salida estándar del nodo maestro del clúster. Esto es muy importante para el buen control y manejo de errores y debug del programa que se vaya a ejecutar y tambien para mostrar resultados. Si no se ingresa esta directiva, solo el nodo maestro imprimira en la salida estándar del clúster (en su salida estándar).

Mpicc.

Para compilar programas no es necesario tener en ejecución el boot schema.

mpicc -o programa programa.c
mpif77 -o programa programa.f
mpic++ -o programa programa.cc

Lamwipe.

Es el comando que garantiza que el universo LAM ha sido apagado adecuadamente.

lamwipe -v hostfile

Lamclean.

Este comando elimina todos los programas que se estén ejecutando del universo LAM. Útil cuando un proceso en paralelo termina de una forma inesperada o queda en un estado no conveniente para el universo de LAM.

lamclean -v

Más comandos de LAM/MPI: mpiexec, mpitask, lamnodes, lamhalt, etc.

4. Resultados

5. Conclusiones

En este articulo se demostró como un usuario puede crear su propio clúster para satisfacer sus propias necesidades de cómputo. Dado que los algoritmos son cada vez más complejos, es muy importante que uno mismo como usuario cuente con los suficientes recursos para que las investigaciones, trabajos cientificos y demas tareas puedan llevarse acabo de una manera rapida y eficiente. De esta manera se demuestra que con equipos convencionales se puede llegar a obtener el mismo rendimiento que con una supercomputadora especializada.

Referencias

[1]

[2] http://elouai.com/hosts-linux.php.

[3] http://tldp.org/howto/thinclient-howto-5.html.

[4] http://www.alorda.jazztel.es/aol/index.htm.

[5] http://www.comptechdoc.org/os/linux/howlinuxworks/linux_hlsysconfig.html.

[6] http://www.lam-mpi.org/download/files/7.1.4-user.pdf.

[7] http://www.lam-mpi.org/download/files/7.1.4-user.pdf.

[8] http://www.linkbyte.com/support/linux_ifcfg_eth0.html.

[9] http://www.top500.org/lists.

[10] http://www.tuxfiles.org/linuxhelp/fstab.html.

[11] T. Droms, R. Lemon. *The DHCP Handbook*. Sams, 2003.