Implementación del clúster OpenSSI

Como se habló en la sección [**Optando por OpenSSI como un clúster de alto desempeño y balanceo de carga**], la distribución de procesos en los nodos del clúster es la principal característica de funcionalidad de OpenSSI. De una forma ideal, los procesos buscaran ejecutarse en un ambiente igual o muy similar del nodo donde provienen, tal vez con mejores recursos de memoria y procesador. Hablando en términos del programa, éste estaría ejecutándose en nodos externos, por lo que es indispensable que el procesador externo tenga el conjunto de instrucciones necesarias para poder procesar el código compilado en el nodo anfitrión, de otra forma el programa no podría ejecutarse.

Con base en estas observaciones, se concluye que la construcción de un clúster OpenSSI como en otros tipos de clústeres es muy recomendable que se haga con hardware homogéneo, aunque muchas de las veces debido a los recursos con los que se pueden disponer en nuestro centros de investigación no es posible cumplir con este requerimiento, no indispensable pero si recomendado, pues además de proveer un ambiente de ejecución adecuado para los procesos, facilita la administración del software instalado en los nodos del clúster.

3.1. Requerimientos para la implementación de OpenSSI

3.1.1 Requerimientos de Hardware

La elección del hardware a utilizar en un clúster puede definirse primeramente en base a conocer la magnitud del problema o problemas que se quieren resolver, el hardware que puede adquirirse en el mercado, los recursos económicos y humanos con los que se cuentan, tanto para administrar el clúster cómo para programar y ejecutar las aplicaciones. Cuando se planea la adquisición, se propone la compra del “mejor” equipo de computo.

La descripción de estos nodos es una propuesta ideal para ensamblar el clúster, en ésta también se describe el tipo de red o canal de comunicación que se recomienda, aunque por supuesto es hardware que cumple con los requerimientos básicos. Cabe aclarar que se propone la construcción de un clúster especifico, pues su propósito, problema o proyecto en particular, que se ha propuesto es para la ejecución de banco de datos.

En el capítulo **N**, en las pruebas de rendimiento se mostrará la utilidad de OpenSSI en relación a motores de banco de datos, por tanto se propondrá el diseño de un clúster que pueda en el mayor de los casos responder a las necesidades generales.

Se identifican como nodos aquellas unidades individuales de procesamiento en el clúster. Para la implementación de un clúster OpenSSI, el hardware soportado es para las arquitecturas IA32 y compatibles, es decir, en términos del hardware disponible en el mercado se refiere a los procesadores Intel y AMD con tecnología para procesamiento de palabras de 32 bits. OpenSSI ha sido desarrollado y probado para funcionar en estos dos tipos de procesadores y es recomendable no ensamblar clústeres con ambos tipos de procesadores.

Los factores en la elección de uno u otro tipo de procesador son principalmente su costo y también por supuesto el software que se podrá aprovechar al máximo. Como ejemplo, se puede mencionar que por cuestiones de mercado, los procesadores AMD han sido más económicos sin querer decir con esto que tengan un menor rendimiento o tecnologías más obsoleta o atrasada, pero en cuanto al desarrollo de aplicaciones de software comerciales y no comerciales, como por ejemplos los compiladores, estas han sido desarrolladas y optimizadas para los procesadores Intel. Esto no quiere decir que sea imposible de implementar un clúster de bajo costo con hardware de bajo costo, simplemente se requiere en la mayoría de los casos un estudio de las herramientas disponibles para optimizar las aplicaciones de los usuarios.

Para este trabajo de tesis, el hardware requerido se encuentra disponible en el Centro de Investigación de Computo de la Facultad de Ingeniería, C.I.C.F.I.; allí se implementará la infraestructura de un clúster conformado por 3 nodos. A continuación se listara el hardware disponible de manera ilustrativa, no se abarcara en detalles los dispositivos periféricos (teclado, mouse, etc.), estas quedan a criterio de las necesidades reales que se pueda dar a estos dispositivos en el clúster

|  |  |
| --- | --- |
| **Procesador** | AMD Athlon(tm) Processor LE-1640  Processor Speed: 2.6 GHz.  L2: 1 MB. |
| **Memoria (RAM):** | 2 GB DDR-2 800 MHz. |
| **Disco Duro (Hard Disk):** | 250 GB, SCSI de 15,000 rpm. |
| **Placa Madre (Motherboard):** | ASUSTeK Computer INC. M2N-MX SE Plus  System Bus Speed: 800 MHz.  System Memory Speed: 667 MHz. |
| **Tarjeta de Red** | Intel (R) Pro/100 Network Conecction  Fast Ethernet |

3.1.2 Requerimientos de Software

**Sistema Operativos:**

Debian GNU/Linux es un sistema operativo libre que soporta un total de doce arquitecturas de procesador e incluye los entornos de escritorio KDE, GNOME, Xfce y LXDE. La versión 5.0.8 Lenny con kernel 2.6 es compatible con las versiones openssi 1.9.6 y la versión alpha de openssi 2.0.

**Sistema de Archivos:**

Ext3: Sistema de archivo utilizado para las particiones Linux, incluye la característica de *journaling* que previene el riesgo de corrupciones del sistema de archivos y es de los mas utilizados por presentar mejor desempeño en el manejo de archivos.

Cluster File System (CFS): Sistema de archivos distribuidos pertenecientes a un grupo de servidores que trabajan en conjunto para proporcionar un servicio de alto rendimiento para sus usuarios en vez de un único servidor con un conjunto de clientes. Para los usuarios del cluster el sistema de archivos es transparente, simplemente un sistema de archivos. El programa de manejo del sistema de archivos se encarga de distribuir las solicitudes a través de los elementos del almacenamiento del cluster. Los sistemas de archivos en cluster (CFS) permiten que múltiples servidores puedan acceder al mismo sistema de archivos. CFS resuelve las desventajas y complejidades de los discos duros proveyendo una solución más simple para la administración del almacenamiento

**Herramienta de Monitoreo:**

Openssi webView: Aplicación grafica de monitoreo de OpenSSI, con ella puede realizarse parte de las tareas comunes de monitoreo que se harían con los comandos. El propósito de esta herramienta es la de proveer una visión general del estado del cluster, graficar las funciones claves del sistema. Permite al administrador del cluster controlar el estado de los recursos de cada nodo y de la infraestructura en general.

3.3. Organización de los nodos

En esta sección se da una descripción de la topología de la red y de los componentes de la conexión entre nodos del clúster, como son el switch y el cable de red utilizados. Cabe mencionar que esta configuración es parte de la red actual del Centro de Investigación en Computación de la Facultad de Ingeniería (CICFI).

La topología de red utilizada es la estrella, se caracteriza por tener todos sus nodos conectados a un controlador central, en este caso, un switch de 8 nodos de los cuales 4 nodos pertenecen al clúster. Todas las comunicaciones pasan a través del switch, siendo éste el encargado de controlarlas. Por este motivo, el fallo de un nodo en particular es fácil de detectar y no daña el resto de la red, pero un fallo en el controlador central desactiva la red completa.

El cable con el que están conectados los nodos del cluster al switch que los une, es estructurado de la marca Kron, para velocidades de transmisión de datos Ethernet 10/100/1000.

Los equipos conectados a este switch se conectan a una velocidad auto negociable de 10Base-T/100Base-TX, según la tarjeta de red.

3.4. Instalación del clúster OpenSSI

El proceso de instalación es un poco dificultoso al inicio, principalmente por lo incompleto y esparzo que resulta ser la documentación oficial[*http://openssi.org/cgi-bin/view?page=docs]*. Es bastante aconsejable estar muy bien familiarizado con la distribución que se estará utilizando, en este caso en particular Debian Lenny, de modo que se puedan investigar y solventar los posibles problemas.

En un entorno OpenSSI debe existir un nodo especial denominado nodo init o master. Éste arranca directamente desde el sistema de archivos raíz. El resto de los nodos arrancan desde la red. Para arrancar los nodos non-init o esclavos por red es necesario Etherboot (definiciones) o PXE (definiciones). Etherboot es software libre y se instala en un medio que permita el arranque, como un CD o un disco duro, o bien se graba en la memoria ROM de la tarjeta de red. La mayoría de las tarjetas de red son compatibles con Etherboot. PXE, por otro lado, se encuentra integrada en algunas tarjetas de red y normalmente se habilita desde el menú de la BIOS.

Debido a que el propósito principal de este trabajo de tesis, es proveer una guía metodología completa de la instalación del nodo master, agregar nodos esclavos y la instalación del una herramienta web para el monitorización visual del cluster OpenSSI.

A continuación se describirá los procesos para implementar un cluster OpenSSI en una distribución de Debian Lenny con los recursos de hardware disponibles en el C.I.C.F.I.

3.4.1. Instalación del Sistema Operativo Debian Lenny

Estas instrucciones asumen que se realizara una instalación limpia del sistema operativo, eso quiere decir que no se estará actualizando o modificando las configuraciones existentes de alguna de las computadoras. Para mas detalles de este proceso, se provee el **Anexo 1. Instalando Debian** en donde se encuentran los pasos para este proceso.

1. Instalar Debian Lenny (<http://www.debian.org/releases/lenny/debian-installer/>) en el nodo master. No existe la necesidad de instalar esta distribución en los demás nodos.
2. Es recomendado que se realice la partición del disco duro usando las herramientas que provee durante la instalación. El sistema de archivos recomendado es ext3 debido a su capacidad transaccional. La tabla de particiones debería estar conformada de la siguiente manera: **/boot**, **/swap**, y **/**.
3. Configurar GRUB (<http://www.gnu.org/software/grub/>) como gestor de arranque. OpenSSI no soporta LILO (<http://lilo.alioth.debian.org/>).
4. Configurar las tarjetas de red con una dirección IP estática. Tanto la tarjeta de red para la conexión a una red externa, como la tarjeta de red para la interconexión con los demás nodos. Durante el proceso de instalación del OpenSSI se configurara la tarjeta de red para la interconexión con los demás nodos esclavos.

3.4.2. Instalación del OpenSSI

Los siguientes procedimientos son los necesarios para la instalación del kernel de OpenSSI en el nodo master. Estos procedimientos pueden encontrarse detallados en el **Anexo 2. Instalación el nodo init o master**.

1. Agregar las siguientes líneas al archivo /etc/apt/sources.list

*deb http://deb.openssi.org/openssi 1.9.6-lenny-preview openssi extras*

*deb-src http://deb.openssi.org/openssi 1.9.6-lenny-preview openssi extras*

1. Agregar las siguientes líneas al archivo /etc/apt/preferences

*Package: \**

*Pin: origin deb.openssi.org*

*Pin-Priority: 1001*

1. Si es necesario, configurar el http proxy.
2. Configurar la dirección IP estatica para la tarjeta de red que servirá de interconexión para los nodos esclavos

#Setting for the Cluster Interconnect

allow-hotplug eth1

iface eth1 inet static

address 192.168.64.1

netmask 255.255.255.0

broadcast 192.168.64.255

1. Ejecutar

*# apt-get update*

*# apt-get dist-upgrade*

1. Agregar los instaladores necesarios de las tarjetas de red al archive /etc/mkinitrd/modules, los cuales seran usandos durante el arranque de los nodos del cluster.
2. Ejecutar

*#apt-get install openssi*

Este comando instalara openssi y creara el primer nodo, init o master, del cluster. Durante la creación del primer nodo, mediante el comando *ssi-create*, aparecerán algunas preguntas relacionadas a las configuraciones del cluster. Para los detalles del mismo, se podrán obtener en la referencia al **Anexo 2. Instalación del nodo init o master**.

1. Reiniciar el nodo master

3.4.2 Agregar nuevos nodos

Una vez instalado el sistema operativo en el nodo master, instalado el kernel openSSI en el mismo, se procede a agregar nodos para poder contar con una infraestructura Cluster. Los nodos esclavos son iniciados por medio de un método de inicio por red. Esto evita la necesidad de instalar el sistema operativo y el kernel del openssi en mas de un nodo. Para iniciar por red un nuevo nodo, es necesario que la tarjeta de red de los nodos esclavos soporten PXE o Etherboot. Estos procedimientos pueden encontrarse detallados en el **Anexo 3. Agregar nodos a la infraestructura**.

1. Si la tarjeta de red del nuevo nodo no posee soporte de inicio PXE, es necesario descargar el instalador Etherboot desde la siguiente url <http://rom-o-matic.net/gpxe/gpxe-1.0.1/contrib/rom-o-matic/> y generar un CD de inicio.
2. Si el nodo requiere un instalador que no se encuentra dentro del archivo /etc/mkinitrd/modules (en el nodo master), es necesario agregar el nombre del instalador en ese archivo y luego volver a generar el disco de inicio para que el cluster incluya ese instalador a su lista. Para realizar esto:

*#mkinitrd -o <init RD image file> <kernel-version>*

*# ssi-ksync*

1. Conectar el nodo esclavo al switch conectado al cluster, insertar el CD de inicio previamente grabado e inicio el nodo. El CD mostrara la dirección MAC del nodo esclavo y espera un tiempo buscando un servidor DHCP que pueda responderle.
2. En el nodo master ejecutar el comando *ssi-addnode –hwaddress=”*MAC\_ADDRESS*”* en donde MAC\_ADDRESS es la dirección MAC del nodo esclavo. Esto hará que el proceso de agregado de nodos inicie y solicitara las configuraciones, estas se pueden encontrar en el **Anexo 3. Agregar nodos a la infraestructura**.
3. El programa hará todo lo necesario para agregar el nodo al cluster. Es necesario esperar que el nuevo nodo se agregue. Un mensaje de “nodeup” aparecerá en la consola del primer nodo indicando el éxito de la operación. Puede comprobar el estado del cluster por medio del comando *cluster –v*.
4. Si se desea configurar el nuevo nodo como un posible nodo master, y habilitar la tolerancia a fallo (failover), es necesario realizar ciertas configuraciones en el hardware del nodo tales como *ssi-chnode* y configurar las particiones del nuevo nodo así como el sistema de inicio GRUB.
5. Repetir los pasos anteriores cada vez que se desee agregar otros nodos al cluster.

3.4.3. Monitorización visual del cluster OpenSSI

Una vez lograda la instalación del nodo master, y agregado nodos esclavos, el siguiente paso es la instalación de la herramienta de monitoreo OpenSSI webView. OpenSSI webView es una herramienta PHP desarrollada por Kilian Cavalotti y extendida por el propio equipo de la tesis. Se encuentra disponible es esta dirección <http://darthvinsus.github.com/webview/>

La herramienta permite visualizar la carga en cada nodo del cluster, monitorizar el estado de cada nodo y ver las graficas con diversas estadísticas. También es posible monitorizar los procesos en cada nodo y migrar manualmente un proceso a otro nodo. Para utilizar esta herramienta es necesario realizar los siguientes pasos.

1. Extraer directamente en la carpeta del servidor el archivo descargado desde la dirección <http://darthvinsus.github.com/webview/>

*# cd <DESTDIR>*

*# tar xfvj DarthVinsus-webview-1902726.tar.gz*

1. Establecer los permisos adecuados en el directorio de gráficos, para la generación de gráficos y la recolección de datos en el archivo config.php, para poder modificar y guardar la configuración desde la interfaz web.

*# cd <DESTDIR>/openssi-webview*

*# chown -R <USER> graphs*

*# chown <USER> config.php*

1. Agregar al cron la tarea de mantener actualizada la interfaz a través de la recolección de datos por medio de un script.

*\*/5 \* \* \* \* <USER> [ -d <DESTDIR> ] && (cd <DESTDIR> && ./graphs/update\_all.sh )*

1. Habilitar la migración de nodos por medio de la interfaz web, para ello modificar el archivo *etc/sudoers* y permitir la migración de procesos a todos los usuarios.

*<USER> ALL = NOPASSWD:/usr/bin/migrate*

Una vez realizado los pasos, acceda desde algún navegador web de su preferencia a la dirección *http:/IP\_NODO\_MASTER/webview*, donde IP\_NODO\_MASTER es la dirección IP del nodo master, configurado en la sección