

Construcción de un flexible y potente cluster usando Raspberry Pi

Dayana Michelle Pineda Borja, Universidad EAFIT

12 de junio de 2016

Traducción de la guía:

Constructing a powerful and flexible computing cluster
using Raspberry Pi computers
(Restrepo Correa Camilo, Universidad EAFIT).¹

1. Introduction

Con la llegada de los ordenadores pequeños, razonablemente rápidos y muy baratos, la idea de facilitar el aprendizaje computacional en la aulas escolares ya no es un sueño de muchos en la comunidad académica, sino más bien, una realidad interesante que está madurando con su exploración. Como tal, ahora se emprenderá el camino de la construcción de un cluster propio, con el increíble Raspberry Pi. Con esto en mente, el objetivo de este proyecto es, evidentemente, el crear un pequeño y muy fácil cluster, que sea sencillo de configurar y poner en funcionamiento para fines educativos (después de todo, usted no podrá ser capaz de hacer cálculos muy complejos con este hardware). Este cluster será muy adaptable, fácil de manejar y mantener, y se comportará exactamente como un ordenador de alto rendimiento pero en una escala mucho más pequeña.

A pesar de la naturaleza simple de este proyecto, se parte de supuesto de que el lector tiene el conocimiento y la experiencia básica utilizando sistemas UNIX. Se trabajará para que el sistema no sea muy robusto y así su configuración sea lo más fácil posible, con el fin de prepararlo para ejecutar una aplicación muy especial, John the Ripper, que sirve de puente entre el mundo de la computación de alto rendimiento y el mundo de la Seguridad cibernética. Aunque esto podría incluso no tener sentido, es imperativo que usted entienda, de cara a esta, que el verdadero propósito de esto es para permitir que todo el que lo lea pueda adquirir y absorber una gran cantidad de habilidades que les permitirán seguir disfrutando y explorando este audaz y fantástico nuevo mundo con la eficacia y la preparación .

¹<https://github.com/SISE-EAFIT/RaspiCluster>

Índice

1. Introduction	1
2. Qué es un cluster?	4
3. Qué es una Raspberry Pi y por qué se usará?	4
4. Qué necesitarás?	4
5. Preparativos iniciales	5
5.1. Elección del sistema operativo	5
5.2. Configuración física del cluster	5
6. Configuración del nodo principal	6
6.1. Actualizar los paquetes por defecto con las últimas versiones . . .	6
6.2. Ejecutar el script de configuración	6
6.3. Instalación de todos los servicios necesarios	7
6.4. Configuración de los módulos de ambiente	9
6.4.1. Visión general de lo que se hará	9
6.4.2. Obtener el software	9
6.4.3. Preparación para compilar	10
6.4.4. Compilando módulos	10
6.4.5. Instalando los módulos	11
6.4.6. Configuración de los módulos	11
6.5. Obtener la implementación MPI	12
6.5.1. Visión general de lo que se hará	12
6.5.2. Obtener el software	12
6.5.3. Preparación para compilar	12
6.5.4. Compilando e instalando OpenMPI	13
6.5.5. Preparando el modulefile para OpenMPI	14
6.6. Configuración de la red	15
6.6.1. Creación de una IP estática	15
6.7. Modificación del archivo <i>/etc/hosts</i>	16
6.8. Pasos finales de la configuración básica	17
7. Creación de un nodo secundario	17
7.1. Obtención de las últimas versiones de los paquetes predeterminados	18
7.2. Ejecutar el script de configuración	18
7.3. Instalación de los servicios necesarios	18
7.4. Compartir carpetas por medio de NFS	18
7.5. Preparando los módulos	19
7.6. Configurando la red	20
7.7. Creación de dirección de IP estática	20
7.7.1. Modificando el archivo <i>/etc/hosts</i>	21
7.8. Pasos finales de la configuración básica	21

8. MPI, estabilidad y pruebas de rendimiento	21
8.1. MPI ejemplo de cálculo - C Versión	22
8.1.1. Visión general de lo que se hará	22
8.1.2. Obtener el software	22
8.1.3. Preparando para compilar	22
8.2. Compilando el software	22
8.3. Probando el software	23
8.4. HPL Performance Benchmark	23
8.4.1. Visión general de lo que se hará	24
8.4.2. Obtener el software	24
8.4.3. Compilar e instalar OpenBLAS	25
8.4.4. Compilar e instalar HPL	25
8.4.5. Probando el software	27
9. Jhon The Ripper	28
9.1. Visión general de lo que se ha hecho y lo que se hará	29
9.2. Obtener el software	29
9.3. Configurando el software	29
9.4. Compilar e instalar el software	30
9.5. Prerando el archivo modulefile para John The Ripper	30
9.6. Preparación de un conjunto de valores hash para las pruebas . .	31
9.7. Crackear contraseñas	32
9.8. Desempeño	32
10. Pensamientos para una futura expansión	32
11. Conclusiones	33
12. References	34

2. Qué es un cluster?

Un cluster es un conjunto de máquinas independientes, conocidos como nodos, conectados entre si mediante una red (ya sea Ethernet o Wi-Fi), de tal forma que el conjunto es visto como un único ordenador, más potente que los comunes, funcionando en paralelo y compartiendo recursos para conseguir mayor eficiencia que un solo equipo podría realizar, aunque este sea de gran rendimiento.

3. Qué es una Raspberry Pi y por qué se usará?

La Raspberry Pi son ordenadores compactos de gran potencia que se adquieren a un bajo costo, con la posibilidad de realizar todas las funciones que realizaría un ordenador normal, pero a un nivel computacional más bajo; se desarrollo con la intención de que cualquier persona de cualquier edad tenga un mayor acercamiento a la exploración de los caminos de la informática y el aprendizaje en la programación de lenguajes.

Se usara este dispositivo, ya que además de su fácil accesibilidad, permitirá un acercamiento de forma sencilla a la computación de alto rendimiento (HPC), sin la necesidad de invertir grandes cantidades de dinero.

4. Qué necesitarás?

Para la construcción de este pequeño pero potente cluster serán necesarios los siguientes componentes:

- Dos o más Raspberry Pi 1 en adelante
- Un cable micro-USB para cada Raspberry Pi
- Un router de Internet con un switch integrado o un switch de red
- Cables Cat-5 para cada Raspberry Pi, con el fin de conectar cada dispositivo a la red
- Tarjetas SD para cada Raspberry Pi de al menos 4GB de espacio (lo ideal es de 8 GB o más grande).
- Un teclado
- Un cable HDMI
- Un computador con un lector de tarjetas SD

Tener presente que estos son los componentes básicos y esenciales para la construcción del sistema, por lo tanto para el crecimiento del sistema todo lo que se tiene que hacer es empezar a expandir las cantidades de los materiales involucrados, y la realización de todas las operaciones de configuración en cada componente nuevo.

5. Preparativos iniciales

Partiendo en que los componentes están completos y listos para usarse, se dará comienzo con los preparativos

5.1. Elección del sistema operativo

La primera y más importante etapa del proceso, es la elección del sistema operativo a usar, actualmente hay muchas distribuciones² disponibles basadas en Linux (la mayoría), sin embargo cada distribución tiene grandes o pequeñas diferencias donde cada uno de los enfoques en los cuales esta desarrollado complica o facilita su uso. En recomendación la mejor opción para la construcción de un cluster con Raspberry Pi es Raspbian, sin desmeritar los demás sistemas operativos, ya este tiene gran apoyo y soporte en la comunidad y la mayor cantidad de paquetes pre-construidos, lo que hará que conseguir que el sistema funcione sea mucho más fácil. Una forma fácil de instalar Raspbian es usando NOOBS³, un sistema de instalación que cuenta con varios sistemas operativos, en el cual es solo seleccionar el software que se desea usar, este se instala en cualquier tarjeta SD con un mínimo de 4 GB de espacio, no necesita conexión a Internet para ser utilizado. La elección que se elija, se instalará automáticamente en el espacio libre de la tarjeta y ya solo es cuestión de reiniciar la Raspberry Pi y usar.

Recordar descomprimir el archivo NOOBS y copiar y pegar los archivos descomprimidos en la tarjeta SD; en caso de descargar directamente Raspbian es solo identificar el archivo .img, el cual sera copiado y pegado en la tarjeta SD.

5.2. Configuración física del cluster

Ahora que la tarjeta SD está lista, es el momento de preparar todos los componentes físicos que se utilizarán. Dependiendo de cómo se ha elegido trabajar, y el número de nodos que tendrá en su red, podría ser mejor montar físicamente el cluster antes de realizar cualquier trabajo en él. Sin embargo, si usted planea operar un simple y pequeño cluster, entonces es posible que desee evitar el establecimiento de demasiadas cosas de antemano, y simplemente empezar directamente a trabajar en el cluster.

No importa lo se haya elegido, a pesar de todo se necesitará, para la primera configuración: una conexión de red, un monitor y un teclado. Tan pronto como se conecten todos los elementos e inserte la tarjeta SD en su ranura, se conecta el cable de alimentación y se da comienzo al cluster.

Como advertencia Raspberry Pi, en la teoría es compatible con una gran cantidad de dispositivos periféricos, pero que no garantiza que

²Listado de algunos sistemas operativos para Raspberry Pi. <http://dplinux.net/listado-de-todos-los-sistemas-operativos-para-raspberry-pi/>

³Descarga desde la página de Raspberry Pi. <https://www.raspberrypi.org/downloads/>

los dispositivos particulares que se va a utilizar serán soportados, sobre todo si, por ejemplo, se desea utilizar un adaptador de HDMI a VGA, un teclado inalámbrico o un dongle WiFi. No obstante, hay muchos recursos en Internet listos para señalarle la dirección correcta, y la mayoría de estas soluciones es sólo una búsqueda en Google para encontrar la respuesta.

6. Configuración del nodo principal

Si todo hasta este punto ha salido bien, entonces probablemente le aparecerá un mensaje que le pregunta para el usuario y contraseña. Si la elección fue con Raspbian, entonces, sólo hace falta introducir *pi* como su usuario y *raspberrypi* como contraseña. Ahora verá que la primera Raspberry Pi está lista para comenzar la configuración del cluster.

6.1. Actualizar los paquetes por defecto con las últimas versiones

Una vez se haya entrado en el sistema, ahora es el momento para iniciar el proceso de obtener los paquetes más nuevos del sistema, ya que siempre es una buena idea tener la última y más actualizada versión de las utilidades del sistema base. Todo lo que se necesita hacer es introducir las siguientes líneas:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

El primer comando lo que hace es actualizar los repositorios (ver si hay algo nuevo), es decir actualizar la lista de todos los paquetes, con la dirección de dónde obtenerlos para que a la hora de hacer la búsqueda y su posterior descarga, sea más rápida. En cambio, el segundo comando actualiza el sistema con todas las posibles actualizaciones que pudiera haber, es decir no sólo actualiza el sistema operativo sino que también las aplicaciones que están contenidas en los repositorios.

6.2. Ejecutar el script de configuración

A continuación, tendrá que activar el servicio SSH en la Raspberry Pi, para que pueda conectarse a él de forma remota (En caso de querer conectarse desde otro equipo más cómodo). Para entrar en el script de configuración, introduzca el comando:

```
$ sudo raspi-config
```

Con las teclas flecha, desplácese hasta la sección de opciones avanzadas (advanced options en Inglés), vaya a la sección de ssh y a continuación, habilite el servicio y salga del script. Una vez que haya terminado, la Raspberry Pi se reiniciará y estará listo para continuar.

En caso de realizar la conexión ssh desde su computador a la Raspberry Pi, solo necesitará usar el comando

```
$ ssh usuario@ip_usuario
```

Donde usuario en este caso sería pi y la ip_usuario se podrá visualizar con el comando:

```
$ ipconfig
```

Donde se introduce la ip de la interfaz que se este utilizando, en este caso la ip del puerto ethernet.

6.3. Instalación de todos los servicios necesarios

Ahora que se a habilitado la conexión ssh, es el momento de descargar todas las utilidades que se requerirán con el fin de montar nuestro cluster. Ingresar el siguiente comando ⁴ para descargar todo lo que se necesitará es:

```
$ sudo apt-get install gfortran g++ nmap rpcbind\  
nfs-kernel-server nfs-common tcl8.5 tcl8.5-dev vim\  
tree htop libssl-dev m4 gawk libc6-dev libstdc++6 libnss3
```

Esta descarga toma tiempo en completarse, no preocuparse por si tarda más de lo que ud creería. A continuación se da una descripción breve de lo que cada comando realiza:

- **gfortran**: Ejecuta todas las opciones soportadas por el comando gcc.
GCC: es un compilador integrado del proyecto GNU para C, C++, Objective C y Fortran; La sigla GCC significa "GNU Compiler Collection".
- **g++**: Forma parte del GCC, GNU Compiler Collection (del Inglés, colección de compiladores GNU). En sistemas operativos GNU, gcc es el comando usado para ejecutar el compilador de C, mientras que g++ ejecuta el compilador de C++.
- **nmap**:(mapeador de redes) es una herramienta de código abierto para exploración de red y auditoría de seguridad. Se diseñó para analizar rápidamente grandes redes.
- **RPC**:(Remote Procedure Call) es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos.
rpcbind: Permite que los clientes NFS descubran qué puerto está utilizando el servidor NFS.
- **NFS**:(Network File System) es un protocolo de sistema de archivos distribuido que permite montar directorios remotos en el servidor.

⁴Como se verá en el siguiente comando y, en varios otros comandos, una barra invertida (\) indicará que el comando continúa, pero en la siguiente línea

nfs-kernel-server: es un paquete que permitirá compartir los directorios.
nfs-common: Paquete que permitirá acceder como cliente en el servidor nfs.

- **tcl**: Se utiliza principalmente para el desarrollo rápido de prototipos, aplicaciones "script", interfaces gráficas y pruebas. La combinación de Tcl con Tk (del inglés Tool Kit) es conocida como Tcl/Tk, y se utiliza para la creación de interfaces gráficas.
- **vim**: es un editor de textos similar al programa "VI".
- **tree**: Comando que permite visualizar toda la jerarquía de archivos y directorios a partir del directorio actual.
- **htop**: Permite ver como se esta usando la memoria de un PC, además de vigilar sus procesos. permite cambiar la prioridad de un proceso o simplemente matarlo.
- **TLS**: (Transport Layer Security) es un protocolo que garantiza la privacidad entre aplicaciones que se comunican y sus usuarios en Internet. Cuando un servidor y cliente se comunican, TLS garantiza que ningún tercero pueda interceptar o alterar cualquier mensaje. TLS es el sucesor de la Capa de sockets seguros (SSL).
- **SSL**: (Secure Sockets Layer) es una tecnología de seguridad estándar para el establecimiento de un enlace encriptado entre un servidor y un servidor web (sitio web) y un navegador cliente normalmente.
libssl: librería que soporta una porción de OpenSSL que soporta TLS (Protocolos SSL y TLS), y depende de la librería libcrypto. Para utilizarlo es necesario incluir (al menos) openssl/ssl.h y para enlazar su programa con la biblioteca libssl.
- **M4**: Es un procesador de macros, lo que significa que copia su entrada (a partir de archivos o la entrada estándar) en la salida estándar. Se comprueba cada ficha (un nombre, una cadena entre comillas, o cualquier carácter individual que no es una parte de un nombre o una cadena) para ver si es el nombre de una macro. Si es así, la ficha se sustituye por el valor de la macro, y luego que el texto es empujado de nuevo en la entrada se vuelve a analizar. Manipula archivos, realiza operaciones aritméticas y tiene funciones para el manejo de cadenas.
- **gawk**: Interpreta un lenguaje de programación de propósito especial que hace posible manejar trabajos sencillos de datos a formatear con sólo unas pocas líneas de código.
- **libc**: Es una librería de C que básicamente contiene todas las funciones del sistema que la mayoría (si no todos) los programas necesitan para funcionar en Linux. Es similar a una combinación de dos.library y exec.library, pero también contiene una gran cantidad de cosas que se encuentran en

la biblioteca de tiempo de ejecución C (como, por ejemplo, archivos `ixemul.library` o el `.lib` incluidos con SAS / C y otros compiladores para AmigaOS)

- **libnss3**: Este es un conjunto de bibliotecas diseñadas para permitir el desarrollo multiplataforma de aplicaciones cliente y servidor con seguridad habilitada. Permite usar la versión 2 y 4 de SSL, TLS, PKCS #5, #7, #11, #12, S/MIME, certificados X.509 versión 3 y otros estándares de seguridad.

A continuación, se procede a crear un par de carpetas en el directorio compartido de manera que se pueda, de ahora en adelante, llenarlo con el software que será utilizado por los demás nodos. Por lo tanto, seguir adelante y hacer lo siguiente:

```
$ sudo mkdir -p /share/apps/ && sudo mkdir -p /share/modules/
```

6.4. Configuración de los módulos de ambiente

El tiempo ha llegado finalmente para comenzar a insertar en el cluster algunos de los mejores y más potentes software. El primero a instalar es Environment Modules, un pequeño y sencillo programa que se come secuencias de comandos de TCL; le permite armar y desarmar su entorno de trabajo, por lo que puede utilizar la versión exacta que desee de un programa.

6.4.1. Visión general de lo que se hará

Con el fin de conseguir que los módulos de ambiente funcionen, es necesario compilarlos; no está disponible en cualquiera de los repositorios públicos para Raspberry Pi, y, aunque lo fuera, es un buen ejercicio tratar de poner en marcha este programa. Además, la compilación elaborada por ustedes mismos les permite controlar, con la máxima precisión, cómo quieren que se comporte el programa. En concreto, la idea es que no se instale en una carpeta por defecto sino más bien en el directorio `/share/apps` creado anteriormente, por lo que sólo hay que instalarlo una vez de forma que este siempre esté a disposición de todos los demás nodos, asimismo se puede configurar para buscar por sus modulefiles (sobre esto más adelante) donde queremos que se realice algo, y que en ninguna otra parte se realice.

6.4.2. Obtener el software

Para obtener el software procedemos a descargar el paquete comprimido, que se puede obtener fácilmente a partir de su página web utilizando un simple comando `wget` (gestor de descargas) y un comando `tar` (comando para descomprimir archivos) con el fin de acceder a todos los contenidos del archivo descargado.

```
$ cd
$ wget http://sourceforge.net/projects/modules/files \
```

```
/Modules/modules-3.2.10/modules-3.2.10.tar.gz/  
$ tar xf modules-3.2.10.tar.gz
```

6.4.3. Preparación para compilar

Una vez decomprimido el paquete ingresar a dicha carpeta:

```
$ cd modules-3.2.10
```

Después leer los archivos *README* y *INSTALL*, y, una vez que se haya finalizado su lectura proseguir con el siguiente comando para ver hasta qué punto se llega, la razón detrás de esto es que, en última instancia, cada instalación es diferente ⁵

```
$ ./configure --without-tclx
```

El script `configure` creará un archivo llamado `makefile` y el `makefile` constituye la base de compilación.

Si el comando falla, entonces significa que no es capaz de encontrar algo que se necesita en el sistema, así que esta será una buena oportunidad para familiarizarse con típicos problemas en la compilación, sin embargo, si termina correctamente, continuar con el comando de configuración. Lo que se debe hacer ahora, está a cargo de nuestro script de configuración para que se ajuste completamente a las variables de forma correcta.

```
$ ./configure --prefix=/share/apps/modules/3.2.10/  
--with-module-path=/share/modules/ --without-tclx\  
--enable-shell-funcs --enable-shell-alias --enable-shell-eval
```

El flag "prefix" permite establecer la ubicación donde se compilará el programa. Una vez realizado el comando, ahora estará listo para empezar a compilar ⁶

6.4.4. Compilando módulos

No hay mucho que decir sobre el propio proceso de compilación, ya que implica un comando muy simple que es:

```
$ make
```

El "make" se encarga de leer todos los `makefiles` que el script `configure` creó, estos archivos le dicen al `make` cuales archivos compilar y el orden que debe ser compilado, esto es muy importante, ya que podría haber cientos de archivos fuente.

Una vez que introduzca el comando, el cual tomara bastante tiempo, y si se completa correctamente se continuará a poner a prueba su binario:

⁵En general, las instalaciones de base Linux vienen con TCL pero no TclX, sin embargo no va a ser requerido por cualquiera de los scripts que se escribieran.

⁶El más observador se dará cuenta de que se está usando el compilador por defecto del sistema, y que no es en general la mejor idea. Sin embargo, no se necesita nada más que un compilador básico ANSI C, y, el compilador base gcc del sistema será suficiente. En cuanto al rendimiento, lo mismo se aplica, el programa es tan sencillo y pequeño que no es necesario compilar los módulos con cualquier otro compilador.

```
$ make check
```

Ahora, si el proceso de comprobación, o el proceso de compilación falla, probablemente significa que se realizó algo mal y que es previsible regresar al inicio de este proceso y volver a intentarlo. En teoría, no debe fallar en cualquier paso del proceso una vez que se pase del script de configuración.

6.4.5. Instalando los módulos

Habiendo superado esta enorme barrera, ahora es el momento de instalar finalmente los módulos. Para hacer esto correctamente se aconseja realizar el siguiente comando, ya que esto dará temporalmente acceso de escritura y permisos sobre las carpetas donde se va a instalar el paquete de módulos

```
$ sudo mkdir -p /share/apps/modules/3.2.10
$ sudo chown -R pi:pi /share/apps/modules
$ make install
$ sudo chown -R root:root /share/apps/modules
```

El comando *chown* permite modificar a los usuarios o grupos de dueños un archivo o carpeta en el sistema de archivos.

6.4.6. Configuración de los módulos

Suponiendo que el proceso de instalación ha ido bien, el siguiente paso es bastante simple, pero necesario con el fin de garantizar que los módulos estándar están disponibles para todos y cada uno de los nodos del cluster⁷. Con esto en mente, ingresar a la carpeta en la que se instaló los módulos, en este caso sería */share/apps/modules/3.2.10/*.

```
$ cd /share/apps/modules/3.2.10
```

Una vez allí, desplazarse a la carpeta donde se encuentran los scripts de inicialización, y verificar que efectivamente estén ahí.

```
$ cd 3.2.10/init
$ ls
```

Con el fin de terminar el proceso de arriba, todo lo que hay que hacer es ejecutar el siguiente comando:

```
$ sudo cp share/etc/profile.d/modules.sh.
```

Una vez completado el comando, la instalación de los módulos deberá estar listo para usarse. Simplemente reinicie la Raspberry Pi y al volver a iniciarla, ejecute el siguiente comando:

```
$ module
```

⁷El procesos de compartir las carpetas creadas con los demás nodos del cluster se verá más adelante en el proceso NFS

Si aparecen las instrucciones de uso, la instalación de los módulos habrá sido un completo éxito y ahora está listo para proceder con la instalación de cualquier otro paquete que desee utilizar.

6.5. Obtener la implementación MPI

Las Raspberry Pi están equipada para ejecutar y compilar las fuentes de C y C++ que requieren algunas de las nuevas características de estos idiomas. Desafortunadamente no están del todo listas para comenzar la preparación e implementación de aplicaciones paralelas, ya que no tiene una manera de comunicar las aplicaciones de forma que el uno hable con el otro mientras se están ejecutando en diferentes máquinas. Aquí es donde entra MPI, ya que define un paso de mensajes de interfaz de programas que estandariza cómo debería hablar con cada uno de los nodos cuando se ejecutan en máquinas paralelas.

6.5.1. Visión general de lo que se hará

La implementación de MPI no es realmente tan difícil, y que, afortunadamente, se basa en el paradigma familiar del *automake/autoconf*. En este caso, se elegirá apegarse a una implementación MPI probada y verdadera, el cual es OpenMPI versión 1.10.2, que es la última versión estable. Al igual que con lo que hemos hecho antes, vamos a profundizar en el mundo de la compilación una vez más con el fin de construir este software a lo que se desea. Con esto en mente, lo que se hará es lo siguiente:

- Descargar y obtener la fuente para OpenMP versión 1.10.2
- Compilar y configurar a su gusto
- Instalar en la carpeta */share* de forma que todos los nodos puedan acceder al software

6.5.2. Obtener el software

Con el fin de obtener la última versión de OpenMPI, se deberá descargar desde su página web con el siguiente conjunto de comandos:

```
$ cd
$ wget https://www.open-mpi.org/software/ompi/v1.10/\
downloads/openmpi-1.10.2.tar.gz
$ tar xzvf openmpi-1.10.2.tar.gz
```

Ahora que ha extraído el paquete, es el momento de compilarlo.

6.5.3. Preparación para compilar

Curiosamente, en realidad no hay una gran cantidad de nuevas cosas para tener en cuenta una vez que se empieza el camino de la compilación de OpenMPI, y, como tal, tiene sentido decir que prácticamente todas las advertencias y

consideraciones que se aplican a GCC se aplican a OpenMPI. Si, por alguna razón OpenMPI tiene problemas de configuración extrañas, siempre revisar su *config.log*

6.5.4. Compilando e instalando OpenMPI

Primero ingresar en la carpeta extraída

```
$ cd
$ cd openmpi-1.10.2
```

Una vez ahí, se aconseja la lectura del archivo *README*, ya que puede contener información interesante que puede ser directamente para un uso particular correspondiente de la agrupación. Una vez se haya terminado es el momento para comenzar a configurar el software a las necesidades del sistema específico. Tener en cuenta que a medida que pasa el tiempo si llega a encontrar una versión en la que confíe y encuentre completamente estables en su sistema particular, ya que, a su debido tiempo, usted encontrará que es casi inevitable encontrar errores en el OpenMPI que se haya instalado ⁸. Por lo tanto, si usted ha llegado a un acuerdo con esto, es hora de continuar

```
$ ./configure --prefix=/share/apps/openmpi/1.10.2/gcc/4.8.5/\
--build=arm-linux-gnueabi
```

Esta configuración probablemente llevará algún tiempo, pero, en teoría no debería fallar, siempre que haya cargado los módulos correctos.

El siguiente paso es algo diferente a lo que se ha hecho antes, ya que se recomienda que tanto para su construcción e instalación se realicen un solo paso, por lo que se va a preparar primeramente las carpetas y luego se prosigue con la instalación

```
$ sudo mkdir -p /share/apps/openmpi/1.10.2/gcc/4.8.5
$ sudo chown -R pi:pi /share/apps/openmpi
$ make all install
$ sudo chown -R root:root /share/apps/openmpi
```

Si todos estos se completarán sin ningún problema, entonces el software ha sido instalado. Verificar esto ejecutando el siguiente comando

```
$ /share/apps/openmpi/1.10.2/gcc/4.8.5/bin/mpicc -v
```

Una vez se haya verificado que todo está bien, ahora es el momento para darse cuenta de que el trabajo todavía se tiene que hacer antes de que se pueda utilizar los nuevos OpenMPI.As; se prosigue a crear un nuevo módulo para ello, nos permitirá utilizar con seguridad.

⁸Esto, sin embargo, no significa que la versión que va a utilizar es de ninguna manera inusable o malo.

6.5.5. Preparando el modulefile para OpenMPI

Para la creación de un módulo para OpenMPI ⁹ ejecutar los siguientes comandos:

```
$ cd /share/apps/modules/3.2.10/modulefiles
$ sudo mkdir openmpi
$ sudo touch openmpi/1.10.2_gcc-4.8.5
$ sudo chown pi:pi openmpi/1.10.2_gcc-4.8.5
$ vi openmpi/1.10.2_gcc-4.8.5
```

El comando touch actualiza los registros de fecha y hora, con la fecha y hora actual de los ficheros indicados como argumento. Si el fichero no existe, el comando touch lo crea.

Tener mucho cuidado en la escritura del módulo, ya que cualquier numeral que falte puede afectar en la falla del módulo.

```
##%Module10#####
##
## Modulefile for OpenMPI 1.10.2 built with GCC 4.8.5
##
##
proc ModulesHelp {} {
    global version modroot

    puts stderr "Prepares the environment to use OpenMPI 1.10.2"
}

module-whatis "Prepares the environment to use OpenMPI 1.10.2"

set                topdir      /share/apps/openmpi/1.10.2/gcc/4.8.5
set                version     1.10.2
set                sys         armv6

module load gcc/4.8.5

prepend-path       PATH         $topdir/bin
prepend-path       C_INCLUDE_PATH $topdir/include
prepend-path       CXX_INCLUDE_PATH $topdir/include
prepend-path       CPLUS_INCLUDE_PATH $topdir/include
prepend-path       LD_LIBRARY_PATH $topdir/lib
prepend-path       LD_RUN_PATH    $topdir/lib
prepend-path       LIBRARY_PATH  $topdir/lib
prepend-path       MANPATH       $topdir/share/man
setenv             MPI_BIN      $topdir/bin
```

⁹Puede que usted note una diferencia en cómo se llama el módulo , ya que es muy importante saber exactamente lo que un módulo está cargando

```

setenv      MPLSYSCONFIG      $topdir/etc
setenv      MPLRUN            $topdir/bin
setenv      MPLINCLUDE        $topdir/include
setenv      MPLLIB            $topdir/lib
setenv      MPLMAN            $topdir/share/man
setenv      MPLCOMPILER       mpicc
setenv      MPLSUFFIX          _openmpi
setenv      MPLHOME           $topdir
setenv      CC                 mpicc
setenv      CXX                mpiCC
setenv      F77                mpifort
setenv      F90                mpifort
setenv      FC                 mpifort
setenv      MPLINTERCONNECT    p4
setenv      MPLVENDOR          openmpi
#####

```

Como se puede ver, este modulefile en particular es un poco largo, sin embargo es muy útil, ya que contiene la mayoría de las variables de entorno que se pueda necesitar en algún momento con OpenMPI. En el futuro es probable que se haga un poco uso de OpenMPI, por lo que estas medidas tienen en la actualidad el objetivo de ahorrar tiempo.

6.6. Configuración de la red

La Raspberry Pi utiliza una dirección IP dinámica, que por desgracia no es como se debe configurar la red para el cluster, ya que cada nodo deberá contar con una ip que no cambie, ya que esta ip se insertará en varios archivos donde se volverá tedioso, el tener que cambiar la Ip, ya que es bastante útil saber en realidad donde esta cada nodo en la red, esto nos lleva a un punto muy interesante sobre la creación de un cluster con Raspberry Pi. Por lo general sólo tienen una interfaz de red estándar para trabajar. La idea básica es que el nodo maestro debe estar conectado a la Internet, ya que muy probablemente será necesario en algún momento, Sin embargo los nodos no tendrán que conectarse a Internet, ya que pueden y recibirán todo lo que necesitan de su nodo principal. Desafortunadamente, esto no es realmente posible a menos que tenga algún tipo de interfaz de red externa y un conmutador de red adicional para crear su red secundaria. Teniendo en cuenta estas limitaciones, entonces no debería ser una sorpresa que se debe conectar todos los nodos al mismo conmutador / enrutador. Con esto en mente, se deberá modificar la configuración de la red.

6.6.1. Creación de una IP estática

En primer lugar, si es posible, comprobar qué la IP del router que se va a tomar, no entre en conflicto o este ocupada por otro dispositivo. En este caso se tomará la IP *192.128.1.130* que es una dirección perfectamente aceptable para

dar al nodo maestro. Por lo tanto, ahora vamos a cambiar la configuración de la red:

```
$ cd /etc/network/  
$ sudo vi interfaces
```

Ahora, una vez que haya abierto este archivo, modificarlo para que se parezca a esto:

```
auto lo  
  
iface lo inet loopback  
iface eth0 inet static  
address 192.168.1.130  
netmask 255.255.255.0  
gateway 192.168.1.1  
  
allow-hotplug wlan0  
iface wlan0 inet manual  
wpa-roam /etc/wpa-supPLICant/wpa-supPLICant.conf  
iface default inet dhcp
```

6.7. Modificación del archivo */etc/hosts*

Ahora, otro aspecto importante del cluster es el hecho de que estas máquinas deben ser capaces de resolver un nombre de host sencillo con el fin de simplificar la conexión y configuración de archivos de la máquina MPI, esto, también, tiene la ventaja de hacer que sea más fácil para nosotros saber exactamente dónde estamos sin conexión, y también elimina una gran parte del juego de adivinanzas que vendría con sólo usar direcciones IP. Con este fin se debe modificar el archivo */etc/hosts* que añade automáticamente cada nueva máquina de cada nodo al archivo *hosts*, lo que simplifica en gran medida la administración del cluster. Ahora, con el fin de hacer esto, primero se debe decidir dónde se va a colocar los próximos nodos en la red, y ¿Cómo se hará el esquema de nombres. En este caso, se va a seguir la convención estándar de *compute-X-Y*, donde X indica el estante o rack donde está el nodo e Y indica la posición en el propio rack. Además, se va a empezar a asignar las IP del nuevo nodo de *192.168.1.131* en adelante. Por lo tanto, se modificará el */etc/hosts* del nodo maestro

```
$ cd /etc/  
$ sudo vi hosts
```

Una vez que abierto, seguir adelante y añadir la siguiente línea al final del archivo:

```
192.168.1.131    compute-0-0
```

Dirección del segundo nodo. Ahora no tiene que saber la dirección IP de cada nodo de memoria. Como puede probablemente imaginar, esto va a simplificar

en gran medida el uso de ssh y el relleno de los equipos que desea ejecutar programas paralelos.

6.8. Pasos finales de la configuración básica

Ahora que ha terminado de hacer todas estas largas y arduas tareas es hora de hacer una limpieza básica en todos los lugares en los que se ha trabajado, y algunos básicos de re-verificación de que todo el software esta funcionando correctamente, ya que se pretende que los módulos funcionen correctamente. Una forma fácil de hacer esto es comprobar qué aspecto tienen las variables de entorno después de cargar el módulo OpenMPI, como debe ser, en teoría, estar lleno de los caminos que ha establecido para ellos en las secuencias de comandos.

```
$ module load openmpi/1.10.2_gcc-4.8.5
$ env | less
```

Una vez que este satisfecho con lo que ve allí, entonces es ahora de reiniciar el dispositivo para que los cambios que se han realizado en la configuración de red se hagan permanentes.

```
$ sudo reboot
```

Después de esto ahora se puede afirmar con confianza que tiene un nodo maestro funcional. Por desgracia, no se tiene realmente las aplicaciones frescas que se ejecutan en él y que ponen a prueba la funcionalidad MPI. Para solucionar este problema en primer lugar se debe crear un segundo nodo ¹⁰. Es hora de configurar algunas cosas increíbles para ver cuán poderoso y versátil es el cluster.

7. Creación de un nodo secundario

Ahora que la configuración básica de nuestro nodo principal está hecho, ahora es el momento para empezar a preparar el primer nodo de cómputo, por una razón muy importante, poner a prueba las aplicaciones paralelas, y por lo tanto es muy útil tenerlo listo. Afortunadamente no es un proceso muy difícil, ya que, a diferencia de lo que hemos hecho antes, no va a necesitar un montón de cosas con el fin de preparar el nodo, ya que, de ahora en adelante, van a obtener todas las cosas que necesitan del almacenamiento compartido. Por tanto, es aconsejable no hacer demasiado para ellos con el fin de no saturar con cosas innecesarias.

¹⁰Con el tiempo usted comenzará a pensar en un montón de maneras en las que usted puede emplear nuevos nodos. Por ejemplo, usted podría tener un nodo como una especie de cortafuegos para su nodo principal, otro como su nodo de almacenamiento, otro como un nodo de autenticación y así sucesivamente. En verdad, los únicos límites a esto son su imaginación y su presupuesto.

7.1. Obtención de las últimas versiones de los paquetes predeterminados

De manera similar a lo que se hizo en el nodo maestro, usted debe tener una imagen limpia de raspbian montada en esta Raspberry Pi, una vez que se haya esto, se procede a iniciar la sesión utilizando el mismo usuario y contraseña por defecto (usuario=*pi*, password=*raspberrypi*), ejecute los siguientes comandos para actualizar todos los paquetes base y predeterminados:

```
$ sudo apt-get update
$ sudo apt-get upgrade
```

Una vez finalice los comandos, se procede a continuar con el establecimiento del nodo.

7.2. Ejecutar el script de configuración

Al igual que antes, tendrá que ejecutar el script de configuración para activar el servicio SSH. Además de esto, también se tendrá que cambiar el nombre del host de esta Raspberry Pi, con el fin de evitar confusiones y problemas. Para hacer esto, al igual que antes se ejecuta la secuencia de comandos:

```
$ sudo raspi-config
```

Una vez allí, bajo las opciones avanzadas, encontrará las opciones que necesita para hacer los pasos antes mencionados. Seguir adelante y continuar con el reinicio del dispositivo después de hacerlo.

7.3. Instalación de los servicios necesarios

Por ahora, todo lo que se va a necesitar en este nodo es la capacidad de ver los archivos que el nodo maestro tiene en su unidad compartida. Para esto serán necesarios algunos servicios públicos, a pesar de que son muy fáciles de instalar. Para hacer esto ejecutar el siguiente comando:

```
$ sudo apt-get install rpcbind nfs-common
```

Después de que se instale con éxito, se deberá habilitar un servicio en el sistema, ya que, sin ella, no se podrá montar el nuevo sistema de archivos.

```
$ sudo service rpcbind start
$ sudo update-rc.d rpcbind enable
```

En este punto se está en condiciones de montar el sistema de archivos NFS.

7.4. Compartir carpetas por medio de NFS

A estas alturas ya se habrá dado cuenta de que realmente no se necesitan demasiadas cosas en el nuevo nodo y en su lugar sólo se busca, montar la carpeta */share/* del nodo principal en el sistema del nodo. En primer lugar, crear una carpeta vacía en la que montará el recurso compartido remoto:

```
$ cd /
$ sudo mkdir share
```

Una vez hecho esto, ir a la carpeta */etc/* y editar el archivo */etc/fstab*:

```
$ cd /etc/
$ sudo vi fstab
```

Al final del archivo añadir la siguiente línea:

```
192.168.1.130:/share/ /share      nfs          0            0
```

A partir de ahora, se montará automáticamente el sistema de archivos en el arranque, pero, por ahora, se hará de forma manual, así que no se debe reiniciar el dispositivo ¹¹, así que con el fin de hacer esto, introduzca el siguiente comando:

```
$ sudo mount -t nfs 192.168.1.130:/share /share
```

Ahora solo falta configurar el nodo master para compartir la carpeta con el nodo. Para esto se debe modificar el archivo */etc/exports*

```
$ sudo vim /etc/exports
```

Donde se insertará la carpeta a compartir, la IP del nodo y los permisos que tendrá esta, por medio de la siguiente línea

```
/share/ 192.168.1.131(rw,sync,no_subtree_check)
```

Para finalizar se reiniciará los servicios del nodo master por medio de los comandos:

```
$ sudo /etc/init.d/nfs-kernel-server restart
$ sudo exportfs -a
```

Ahora, seguir adelante y entrar en la carpeta */share/* del nodo y verificar que la carpeta contiene las aplicaciones ya compiladas, por medio de los comandos:

```
$ cd /share
$ ls
```

Si ambos pueden acceder y leer los archivos que ahora se ha montado sobre la carpeta */share*, entonces todo ha ido bien, y ya está listo para proceder.

7.5. Preparando los módulos

Ahora que se puede acceder a los archivos del nodo maestro, es el momento de añadir algunas de las funcionalidades que se encuentra en el servidor principal en el nuevo nodo, sobre todo, es posible que, en algún momento, se requerirá módulos en él, por lo que es una gran idea preparar este equipo para utilizarlos en demanda. Afortunadamente, ahora que podemos acceder a él de forma

¹¹No se preocupe, esto de ninguna manera va a romper lo que se ha hecho con el archivo */etc/fstab*, simplemente nos permitirá comenzar trabajando a partir de ahora con nuestro sistema de archivos compartidos.

remota, no es necesario compilar los programas como ha sido ya hecho para el nodo master, esto contiene la ventaja añadida de realizar algunos pasos de configuración simples y trabajar mucho más rápido que antes. Para alcanzar esto, hay que ejecutar algunas órdenes sencillas:

```
$ sudo cp /share/apps/modules/3.2.10/3.2.10/init/\
share /etc/profile.d/modules.sh
$ source /etc/profile.d/modules.sh
```

El comando *source* permite ejecutar un script en la consola actual (mismo proceso) y una vez que retorne, los cambios que se hicieron a las variables de entorno se mantendrán.

Esto ahora permitirá hacer un uso completo de los módulos, mientras se puede probar su correcta implementación, haciendo lo siguiente:

```
$ module avail
```

Este comando debe ahora permitirle ver todos los módulos que ahora se tiene a disposición.

7.6. Configurando la red

El siguiente, y muy importante paso a tomar, es corregir la configuración de la red de la pi, tal como se había hecho con el nodo maestro. La razón de esto es simple, y es que para trabajar con máquinas que tienen una dirección IP fija, será más fácil majerarlos y encontrarlos en la red. Para esto de deberá modificar el archivo */etc/network/interfaces*, además del archivo */etc/hosts*, sin embargo, al igual que antes, esto no va a ser complicado.

7.7. Creación de dirección de IP estática

Tal como se había hecho antes, se deben hacer algunas modificaciones en el archivo */etc/network/interfaces*, de la misma manera que se hizo con el nodo maestro. Para ello, siga los siguientes pasos:

```
$ cd /etc/network
$ sudo vi interfaces
```

Editar el archivo para que quede como sigue:

```
auto lo

iface lo inet loopback
iface eth0 inet static
address 192.168.1.131
netmask 255.255.255.0
gateway 192.168.1.1

allow-hotplug wlan0
iface wlan0 inet manual
```

```
wpa-roam /etc/wpa-supPLICant/wpa-supPLICant.conf  
iface default inet dhcp
```

Una vez que se haya hecho esto, ahora es el momento de echar un vistazo al archivo */etc/hosts*.

7.7.1. Modificando el archivo */etc/hosts*

Como realizado anteriormente se debe trasladar a la carpeta */etc* y abrir el *hosts* archivo:

```
$ cd /etc/  
$ sudo vi hosts
```

A continuación inserte la siguiente línea en el archivo:

```
192.168.1.130    raspberrypi
```

Ip del nodo maestro; una vez hecho esto, el nodo esta ahora, en teoría, listo para ser utilizado.

7.8. Pasos finales de la configuración básica

En realidad no hay mucho que hacer para el nodo ahora que se ha puesto en marcha su funcionamiento. Ahora es el momento de reiniciar el sistema para aplicar la nueva configuración de la red y poner a prueba las configuraciones de inicialización.

```
$ sudo reboot
```

Una vez se reinicie deberá ver que el recurso compartido NFS se ha montado con éxito, y que ahora tiene una dirección IP estática. A partir de ahora se probará las capacidades del cluster, y asegurarse de que en realidad funciona como se pretende.

8. MPI, estabilidad y pruebas de rendimiento

Ahora que tenemos todas las funcionalidades básicas, es hora de comenzar realmente la recopilación y el uso de las aplicaciones de gran alcance que nos permitirán ver lo frutífero del trabajo. En esta ocasión, nos centraremos en dos pruebas bastante prácticas e interesantes. El primero de estos es una sencilla aplicación cuyo único trabajo es calcular el número Pi ¹² utilizando métodos paralelos; Mientras tanto, el segundo se centra en dar una lectura precisa del rendimiento máximo de la agrupación como un conjunto, y por lo tanto es muy interesante para poner en movimiento. Lo ideal es que va a hacer este procedimiento en el nodo maestro.

¹²Lo cual, en cierto modo, es muy apropiado para este proyecto

8.1. MPI ejemplo de cálculo - C Versión

Este es el mencionado programa que calcula el valor de Pi utilizando métodos paralelos para resolverlo. Es muy interesante en el sentido de que ejemplifica exactamente lo que un programa paralelo debe hacer, y también es bastante simple, por tanto, permite medir de inmediato la cadena de herramientas para que todo se haga correctamente. Como esta primera prueba es de inmenso valor, ya que será el punto de partida para la utilización de la agrupación.

8.1.1. Visión general de lo que se hará

Para utilizar este código, es necesario compilar el entorno OpenMPI, por lo que tiene dentro de ella el soporte adecuado MPI y las bibliotecas vinculadas. Se obtendrá el software de su fuente y luego hacer una compilación rápida. Después se hará la primera prueba adecuada de el programa en las dos máquinas al mismo tiempo, es de esperar, sin problemas, lo que nos indica que se ha hecho un buen trabajo de todo el establecimiento, y si por alguna razón no salió como se esperaba, probablemente será capaz de averiguar donde se encuentra el error.

8.1.2. Obtener el software

El software en cuestión es parte de los ejemplos de una introducción a la computación paralela, el primer paso será descargar el software y colocarlo en una carpeta adecuada.

```
$ cd /share/apps/  
$ sudo mkdir -p piexample/noversion/gcc/4.8.5/openmpi/1.10.2  
$ cd piexample/noversion/gcc/4.8.5/openmpi/  
$ sudo chown pi:pi 1.10.2  
$ cd 1.10.2  
$ wget computing.llnl.gov/tutorials/mpi/samples/C/mpi_pi_reduce.c
```

8.1.3. Preparando para compilar

Para compilar correctamente este programa tendrá que cargar el módulo OpenMPI, ya que contiene todo lo necesario, incluye rutas de acceso y rutas de bibliotecas para hacer este trabajo. Para ello basta con ejecutar:

```
$ module load openmpi/1.10.2_gcc-4.8.5
```

Después de hacer esto, usted debe estar más que listo para compilar el programa!

8.2. Compilando el software

La elaboración de este programa es una tarea sumamente sencilla, y, además, le permite familiarizarse con el uso de MPI. Para hacer la compilación, todo lo que tiene que hacer es ejecutar el siguiente comando:

```
$ mpicc mpi_pi_reduce.c -o mpi_pi_reduce
```

Esto le dará un archivo llamado *mpi_pi_reduce* (Sin la extensión *.c*), que es el programa que con el tiempo se deberá correr.

8.3. Probando el software

Aquí es donde finalmente serán capaces de obtener la satisfacción de que realmente el software se ejecuta en varias máquinas, y aquí es donde todo va bien ir muy bien o muy mal. Se empezará por la creación de un (*machinefile*), que es un archivo que OpenMPI utilizará para saber lo que las máquinas necesitarán para funcionar correctamente. Como tal, la primera tarea es crear un archivo y abrirlo:

```
$ touch machinefile
$ vi machinefile
```

Ahora, con la apertura de archivo, debe poner en él:

```
localhost
compute-0-0
```

En esencia, se está indicando que se desea ejecutar OpenMPI en dos procesos, uno en esta máquina (nodo maestro) y uno en el *compute-0-0*. Ahora, una vez hecho esto, todo lo que realmente sigue es ejecutar el programa. Para ello, seguir adelante y ejecutar el siguiente comando ¹³:

```
$ mpirun -np 2 -machinefile machinefile /share/apps/\
piexample/noversion/gcc/4.8.5/openmpi/1.10.2/mpi_pi_reduce
```

Si al compilar el comando mpirun sale un error con el orted agregar el PATH y LD_LIBRARY_PATH en el .bashrc antes de la línea de comandos de inicio Después de hacer esto, deberá ver el lanzamiento del programa y le pedirá una contraseña, esto, por suerte es algo normal y algo que se eliminará pronto, pero, por ahora, sólo se debe introducir: *raspberry* que es la contraseña por defecto de la Raspberry Pi. *pi* usuario. Una vez hecho esto, el programa continuará y verá la salida del programa de la aproximación del número Pi sin problema. Si lo hace, entonces todo salió realmente bien.

8.4. HPL Performance Benchmark

The HPL Benchmark es una herramienta que permitirá que se haga con facilidad y rapidez, lo que nos permite ver lo bien que el sistema lleva a cabo bajo carga ¹⁴. Esta referencia también permitirá obtener una mejor sensación en cuanto a lo que será la configuración, compilación y ejecución de un

¹³Si experimenta problemas con esto, sustituya *mpirun* con */share/apps/openmpi/1.10.2/gcc/4.8.5/bin/mpirun* y *machinefile* con */share/apps/piexample/noversion/gcc/4.8.5/openmpi/1.10.2/machinefile*

¹⁴Ahora, entienden que hay varios otros puntos de referencia (benchmarks) por ahí, que, en teoría, dan una representación mucho mejor y más precisa de la eficacia que un sistema llevará a cabo, es decir, HPC, sin embargo, como el ranking TOP500 utiliza HPL, parece obvio y práctico elegir este.

software científico y distribuido, ya que tienden a ser en ocasiones bastante difícil de manejar y difícil de poner en buen estado de funcionamiento.

8.4.1. Visión general de lo que se hará

HPL es un programa muy interesante de configurar, ya que ha sido preparado para ser esencialmente tan fuerte y bien optimizado como sea posible. Ahora, con el fin de lograr que se ejecuta, se debe tener en cuenta que el programa en sí tiene una verdadera dependencia, que tiene que ser construido para ella, que es de alguna forma de una biblioteca de álgebra lineal. En este caso se tiene varias opciones para elegir, incluyendo uno de la aplicación BLAS o la biblioteca VSIPL, sin embargo, para este caso particular, se utilizará OpenBLAS, que, con la forma en que se ha probado, ha mostrado el mejor rendimiento y comportamiento. La versión específica que se utilizará son 2.1 para HPL y 0.2.15 para OpenBLAS, ya que estas versiones han demostrado ser muy estables. Ahora, con esto en mente, se mostrará brevemente los pasos que se deben seguir para instalar y ejecutar HPL.

- Descargar y obtener el software desde la página fuente para ambos, OpenBLAS y HPL.
- Configurar, compilar e instalar OpenBLAS 0.2.15.
- Configurar, compilar e instalar HPL 2.1.
- Ajustar y configurar los parámetros del archivo de HPL.
- Por último, hay que ejecutar la aplicación.

8.4.2. Obtener el software

En esta ocasión, se debe descargar y extraer dos piezas de software, para eso prosiga hacer lo siguiente:

```
$ cd
$ wget http://github.com/xianyi/OpenBLAS/archive/v0.2.15.tar.gz
$ wget http://www.netlib.org/benchmark/hpl/hpl-2.1.tar.gz
$ tar xzvf v0.2.15.tar.gz
$ tar xzvf hpl-2.1.tar.gz
```

Ahora que ha terminado de obtener el software, siga adelante e ingrese en la carpeta OpenBLAS:

```
$ cd OpenBLAS-0.2.15
```

Una vez hecho esto, ahora así comenzar a compilar.

8.4.3. Compilar e instalar OpenBLAS

OpenBLAS, a diferencia de la mayoría de las otras aplicaciones que se ha instalado hasta el momento, tiene una estructura y proceso de instalación bastante simple. Esto es debido al hecho de que OpenBLAS se ha construido con el objetivo de ser portátil, y, por lo tanto, tiene una gran capacidad de auto-detección para que sea fácilmente armable en la mayoría de ambientes. Comenzar el proceso de compilación con un comando muy simple:

```
$ make
```

Una vez que finalice, deberá crear la estructura de carpetas donde se va a instalar, y usted tendrá que tener privilegios de escritura. Así que, para hacerlo, haga lo siguiente:

```
$ sudo mkdir -p /share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/
$ sudo chown pi:pi /share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/
```

Se configurará los permisos hacia atrás, pero por ahora, hacer lo siguiente:

```
$ make install PREFIX=/share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/
```

Una vez hecho esto, ahora es el momento de cambiar permisos de la carpetas creadas *root*

```
$ sudo chown root:root /share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/
```

Con este hecho ahora, se tiene las librerías necesarias para poder compilar HPL, de esta manera podemos dirigir la segunda y última prueba.

8.4.4. Compilar e instalar HPL

HPL, al igual que con la mayoría del software utilizado en los clusters es de computación de alto rendimiento, es algo difícil conseguir que funcione, pero después de familiarizarse con esto, se convierte en algo bastante sencillo. Con el fin de configurar correctamente, seguir adelante y pasar a la carpeta:

```
$ cd ..
$ cd hpl-2.1
```

Una vez allí, tendrá que tomar uno de los archivos de instalación pre-hechos y modificarlo para que sea compatible con el sistema. Para hacer esto, copiar más de uno de los archivos de la carpeta *setup*. Después de eso debe abrirlo y modificarlo para adaptarse a lo que se va a utilizar.

```
$ cp setup/Make.Linux_PII_CBLAS .
$ mv Make.Linux_PII_CBLAS Make.PI
$ vi Make.PI
```

Ahora una vez abierto el archivo, seguir paso a paso los cambios que tienen que realizarse. La línea 64 es la siguiente:

```
ARCH      = PI
```

Líneas 70 hasta 75 deberán ser:

```
TOPdir  = /home/pi/hpl-2.1/
INCdir  = $(TOPdir)/include
BINDir  = $(TOPdir)/bin/$(ARCH)
LIBdir  = $(TOPdir)/lib/$(ARCH)
#
HPLlib  = $(LIBdir)/hpl.a
```

Líneas 84 hasta 86 deberán ser:

```
MPdir   = /share/apps/openmpi/1.10.2/gcc/4.8.5/
MPinc   = -I$(MPdir)/include
MPLib   = $(MPdir)/lib/libmpi.so
```

Líneas 95 hasta 97 deberán ser:

```
LAdir   = /share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/lib/
LAinc   =
LAlib   = $(LAdir)/libopenblas.a
```

Líneas 169 hasta 171 deberán ser:

```
CC       = /share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/bin/mpicc
CCNOOPT  = $(HPL_DEFS)
CCFLAGS  = $(HPL_DEFS) -fomit-frame-pointer -O3 -funroll-loops
```

Y por último línea 176 deberá ser:

```
LINKER   = /share/apps/openblas/0.2.15/gcc/4.8.5/openmpi/1.10.2/bin/mpicc
```

Asegurese de guardar el archivo. Una vez hecho esto, entonces usted estará listo para construir el programa en sí. Esto se puede hacer con los siguientes comandos:

```
$ make arch=PI
```

Una vez que haya terminado con esto, usted será capaz de encontrar y ejecutar dentro de una carpeta que se creará. Este mismo archivo que va a ser el que nos permitirá ejecutar todas las pruebas deseadas. Por favor verificar que un ejecutable llamado *xhpl* está presente en la carpeta *bin/PI/*. Para ello, haga lo siguiente:

```
$ ls bin/PI/
```

Debería poder ver, dentro de la carpeta, el archivo mencionado anteriormente. Además tendrá que buscar un archivo llamado *HPL.dat*, que será el archivo que se va a utilizar para hacer la configuración de ejecución para *HPL*. Si con éxito, encuentran los archivos, ahora es el momento de realizar la instalación. Hacer los siguientes comandos:

```
$ sudo mkdir -p /share/apps/hpl/2.1/gcc/4.8.5/openmpi/1.10.2/
$ sudo chown pi:pi /share/apps/hpl/2.1/gcc/4.8.5/openmpi/1.10.2/
$ mv bin/PI/{xhpl,HPL.dat} /share/apps/hpl/2.1/gcc/4.8.5/openmpi/1.10.2/
$ sudo chown -R root:root /share/apps/hpl/2.1/gcc/4.8.5/openmpi/1.10.2/
```

Una vez hecho esto, usted ahora tiene una implementación funcional de *HPL* que se puede ejecutar.

8.4.5. Probando el software

Ahora que todo, en términos de construcción e instalación, ha finalizado, es hora de probar el software con el fin de garantizar que el sistema es ahora estable en condiciones de funcionamiento. Para lograr este objetivo, seguir con unos pasos bastante simples. Para empezar, hay que editar el archivo *HPL.dat*, así adelante y abralo:

```
$ cd /share/apps/hpl/2.1/gcc/4.8.5/openmpi/1.10.2/
$ sudo vi HPL.dat
```

Una vez abierto, se dará cuenta de que este archivo está lleno de configuraciones. La mayoría de parámetros, que en teoría pueden dejarse como están, aunque con, algunos, especialmente el tamaño *N*, que determina la cantidad de RAM que puede utilizar a la vez, y el *NB*, que determina cómo se dividirá la matriz, es perfecto para hacer los cambio permitentes en dichos parámetros que necesitan ser cambiados. Además de estos, las líneas *P* y *Q*, ya que estos indican cómo se presenta el cluster. Y, ya que se esta utilizando sólo 2 máquinas con un núcleo cada una, que sólo puede realmente ejecutar la instalación como se muestra en el archivo. Con todo esto en mente, se proporcionará la configuración óptima para estas dos máquinas:

```
HPLinpack benchmark input file
Innovative Computing Laboratory, University of Tennessee
HPL.out          output file name (if any)
6                device out (6=stdout,7=stderr,file)
1                # of problems sizes (N)
9984             Ns
1                # of NBs
192             NBs
0                PMAP process mapping (0=Row-,1=Column-major)
1                # of process grids (P x Q)
1                Ps
2                Qs
16.0            threshold
1                # of panel fact
2                PFACTs (0=left, 1=Crout, 2=Right)
1                # of recursive stopping criterium
4                NBMINs (>= 1)
1                # of panels in recursion
2                NDIVs
1                # of recursive panel fact.
1                RFACTs (0=left, 1=Crout, 2=Right)
1                # of broadcast
1                BCASTs (0=1rg,1=1rM,2=2rg,3=2rM,4=Lng,5=LnM)
```

```

1          # of lookahead depth
1          DEPTHs (>=0)
2          SWAP (0=bin-exch,1=long,2=mix)
64         swapping threshold
0          L1 in (0=transposed,1=no-transposed) form
0          U  in (0=transposed,1=no-transposed) form
1          Equilibration (0=no,1=yes)
8          memory alignment in double (> 0)

```

Ahora, con los cambios pertinentes hechos, se esta sólo a un paso para realmente ser capaz de ejecutar la prueba, y, sin más preámbulos, aquí está lo que se necesita hacer para ejecutarlo:

```

$ sudo cp /share/apps/piexample/noversion/gcc/4.8.5\
/openmpi/1.10.2/machinefile .
$ mpirun -np 2 -machinefile machinefile xhpl

```

Y con eso, la prueba debe estar en marcha y obtendrá un informe de ejecución del mundo real cuando se termine. Con suerte podrá ver que el cluster funciona bastante bien, y ofrece muy buen rendimiento. Con esto en mente, no olvide guardar esa configuración HPL.dat, ya que puede volver a ajustar los cambios para aprender más acerca de la optimización de aplicaciones paralelas.

9. Jhon The Ripper

John the Ripper es una excelente aplicación para cualquier persona interesada en el mundo de la seguridad informática, ya que es un programa potente y flexible para el craqueo de hashes y recuperar las contraseñas y códigos de acceso que se encontraban dentro. Sin embargo, el rendimiento de la máquina siempre ha sido un factor decisivo en cuanto a la eficacia de John the Ripper es el agrietamiento en los hashes. Con esto en mente, no debería ser ninguna sorpresa que tal vez si dos máquinas trabajan en paralelo, seguramente podrían crackear las contraseñas más fácil que si no lo están. Ahora, John the Ripper es lanzado en tres versiones, una versión "Pro", la cual hay que pagar, sin embargo es muy estable, la versión "Free", y la que es más interesante de todas las versiones, apoyado fuertemente por la comunidad "jumbo", donde los miembros de la comunidad se toman el tiempo para agregar muchas de las características que faltan con la versión oficial de Jhon The Ripper. Esa versión, será la que se utilizará, ya que brindará mayor flexibilidad y mayor cantidad de características. Además, es más fácil de instalar y desplegar en un sistema que es gratis (y oficial ¹⁵). Con todo esto ahora absorbido, proceder con la configuración.

¹⁵No obstante, que apoyen y proporcionen en el sitio web de John Th Ripper, y como tal, realmente parece ser la mejor alternativa para utilizar

9.1. Visión general de lo que se ha hecho y lo que se hará

En primer lugar, la última versión de John the Ripper "jumbo" contiene un gran cuerpo de trabajo realizado por los programadores de la comunidad que se decantó algunas de las nuevas características que se introdujeron para C++ 14, y por desgracia, los compiladores no son compatibles con esta norma. Muchas de las cosas que se han hecho, simplemente son la manera en que en realidad podemos usar John the Ripper, por ejemplo, con el fin de tener a John The Ripper distribuido, es necesario tener los archivos de hash en algún lugar en el que se pueden encontrar fácilmente y ser leídos por todos los sistemas, intrínsecamente usando un sistema NFS. Esto, básicamente, es lo que ha permitido llegar a este punto, donde se está casi listo para ejecutar la aplicación que es el verdadero foco de este trabajo, una versión distribuida de John the Ripper. Ahora para construir y preparar la aplicación para su uso, se deberá que seguir estos pasos:

- Descargar la última versión estable de John the Ripper "jumbo"
- A continuación, se deberá extraer el software a un lugar adecuado.
- Una vez hecho esto, se procede a configurar el software, de modo que pueda ser compilado de acuerdo a las necesidades específicas.
- Compilar el paquete de software y posteriormente instalarlo.
- Preparar un conjunto adecuado de los hashes a ser crackeados.

Después de todo esto, finalmente tendrá un cluster, con una casi infinita cantidad de posibilidades que tiene por delante, a la espera de ser llevado a cabo.

9.2. Obtener el software

La obtención del software es bastante simple, y realmente, dado el hecho de que hay varias opciones, y cualquiera de ellas son perfectamente bien recibidas. Mediante el enlace de descarga oficial de Openwall, se podrá obtener el software, Una vez se tenga, seguir adelante y extraerlo. Para obtenerlo ejecute el siguiente comando:

```
$ cd
$ wget http://www.openwall.com/john/j/john-1.8.0-jumbo-1.tar.gz
$ tar xzvf john-1.8.0-jumbo-1.tar.gz
$ cd john-1.8.0-jumbo-1/src
```

Una vez que esté allí, deberá ver un gran número de archivos, de todo tipo, pero, más importante que cualquier otra cosa, usted debe estar seguro de que hay un archivo de configuración de trabajo.

9.3. Configurando el software

John the Ripper, por defecto no incluye un *configure* incorporado, por lo que su proceso de compilación es más difícil y complicado de lo que en realidad

debería ser (aunque es simple). Afortunadamente se tiene la versión "jumbo" que tiene un robusto y bien diseñado *script de configuración* que hace que el proceso de construcción bastante sencilla. Sólo asegúrese de que usted tiene su módulo OpenMPI cargado. Para configurar este programa, ejecute el siguiente comando:

```
$ ./configure --prefix=/share/apps/johntheripper/\
1.8.0-jumbo-1/gcc/4.8.5/openmpi/1.10.2/ --enable-mpi\
--disable-cuda --disable-openmp --build=arm-linux-gnueabi
```

Una vez finalice, se debe crear la estructura de carpetas, para ello dejar listo la instalación con los siguientes comandos:

```
$ sudo mkdir -p /share/apps/johntheripper/1.8.0-jumbo-1\
/gcc/4.8.5/openmpi/1.10.2
$ sudo chown pi:pi /share/apps/johntheripper/1.8.0-jumbo-1\
/gcc/4.8.5/openmpi/1.10.2
```

Ahora, con eso hecho, es el momento de empezar a compilar el software.

9.4. Compilar e instalar el software

La compilación de John the Ripper, por suerte, es bastante sencillo, y por lo tanto realmente no requiere mucha presentación. Para empezar a compilar, ejecutar estos comandos:

```
$ make
$ make install
```

En caso de presentar error con la función `is_sha512`, mirar la guía:

<http://www.openwall.com/lists/john-dev/2014/12/18/21>

Si después de compilar la carpeta definida en el prefix, se encuentra vacía, copiar todo lo que John The Ripper crea en la carpeta `run` a esa carpeta manualmente. Después de esto, en teoría, John the Ripper esta instalado y listo para usar.

9.5. Prerando el archivo `modulefile` para John The Ripper

Este es un módulo que probablemente permitirá trabajar con John bastante bien. En primer lugar, proceder con la creación y edición del archivo `modulefile`:

```
$ cd /share/apps/modules/3.2.10/modulefiles/
$ sudo mkdir johntheripper
$ sudo vi johntheripper/1.8.0-jumbo-1-gcc-4.8.5-openmpi-1.10.2
```

Ahora que está abierta, seguir adelante y editar la siguiente información:

```
## Modulefile for John the Ripper 1.8.0-jumbo-1
##
##
```

```
##
proc ModulesHelp { } {
    global version modroot

    puts stderr "Prepares the environment for John the Ripper"
}

module-whatIs "Modulefile for John the Ripper 1.8.0-jumbo-1"

module load openmpi/1.10.2_gcc-4.8.5

set      topdir    /share/apps/johntheripper/1.8.0-jumbo-1/gcc/4.8.5/openmpi/1.1
set      version  1.8.0-jumbo-1

prepend-path PATH $topdir/run
```

Esto debería ser todo lo que necesita para el funcionamiento. En este punto se está listo para comenzar las pruebas de John the Ripper.

9.6. Preparación de un conjunto de valores hash para las pruebas

Ahora que ha producido un ejecutable de trabajo, es el momento de comenzar el montaje de un conjunto adecuado de los hashes para crackear. Como sugerencia se crearán 3 usuarios y contraseñas bajo recuento de caracteres. por lo que se puede decir que el software funciona, sin comprometerse a varias horas de trabajo tratando de crackear las contraseñas que se han creado. Para ello, ejecute estos comandos para generar los usuarios:

```
$ sudo useradd user1 -p pw1
$ sudo useradd user2 -p pw2
$ sudo useradd user3 -p pw3
```

Ahora que se ha hecho esto, el sistema tiene 3 nuevos usuarios que están listos para crackearse. Por desgracia, no se pueden crackear todavía. Los modernos sistemas Linux tienen la información del usuario cuidadosamente dividida entre dos archivos, */etc/passwd* y */etc/shadow*, y, con el fin de ser capaz de descifrarlo, debe utilizar una utilidad proporcionada por John the Ripper nombrado *unshadow*, combinarlos en un único archivo. Con el fin de hacer esto, seguir adelante y ejecutar estos comandos:

```
$ sudo su
# module load johntheripper/1.8.0-jumbo-1_gcc-4.8.5_openmpi-1.10.2
# unshadow /etc/passwd /etc/shadow > passdb.txt
# chown pi:pi passdb.txt
# mkdir /share/files
# mv passdb.txt /share/files/.
# cp /share/apps/piexample/noversion/gcc/4.8.5\
```

```
/openmpi/1.10.2/machinefile /share/files /.  
# logout  
$ cd /share/files  
$ module johntheripper/1.8.0-jumbo-1_gcc-4.8.5_openmpi-1.10.2
```

En caso de error, intentar cambiar el primer comando por

```
$ sudo su -
```

Ahora, por fin, se tiene un archivo que se puede utilizar realmente, y ahora, por último, es el momento de poner a funcionar el cluster.

9.7. Crackear contraseñas

Después de todo, se está a un comando de completar los objetivos que se han fijado desde un principio; ahora a ejecutar este comando:

```
$ mpirun -np 2 -machinefile machinefile john passdb.txt
```

Una vez que comienza a crackearse las contraseñas, que va en la demostración de que ha decifrado con éxito tres contraseñas, y como tal, es decir, acoplados a las pruebas anteriores, se tiene como prueba absoluta de que se ha construido un pequeño y fantástico cluster.

9.8. Desempeño

Durante las pruebas se ha encontrado que el rendimiento real de John The Ripper era altamente dependiente de lo bien que la propia red funciona. Sin embargo, si se utiliza en un conmutador de red con velocidades razonables, hubo una ligera mejora en tiempo de ejecución de craqueo hasta tres contraseñas a la vez. Esto fue en el orden de alrededor de un par de minutos. Ciertamente hay una gran cantidad de trabajo por hacer todavía en cuanto a experimentar e investigar, así que esto sigue siendo una interesante pregunta abierta.

10. Pensamientos para una futura expansión

Aquí, al final de este viaje llega un momento en que ahora debe decidir lo que desea utilizar en este cluster, en este caso, sería tener una potencia pequeña y casi no consume utilidad para comprobar si todos los usuarios de un grupo determinado utilizan contraseñas seguras. Pero, como sugerencia las posibilidades son realmente infinitas, y por lo tanto la responsabilidad ahora está en usted para demostrar sus habilidades al demostrar esta más que preparado para el desafío de continuar con este cluster. Por lo menos, esta es una fantástica herramienta educativa que se puede utilizar para enseñar a una nueva generación, los administradores de sistemas altamente competentes.

11. Conclusiones

Habiendo dicho casi todo, ahora viene el tiempo para reflexionar sobre lo que realmente se ha hecho. aunque a veces puede sentir que este proyecto te deja vagar en la oscuridad, una abrumadora sensación de propósito se supera una vez se da los primeros éxitos, y ese es el verdadero combustible y el poder que tiene esta herramienta. La motivación de crear poco a poco algo nuevo a partir de lo que parece algo completamente separado al mundo de la computación de alto rendimiento, es quizás el más potente que se puede dar a un estudiante. El propósito de este documento no es otro que el de servir una pequeña luz que guía en el mundo de la informática de alto rendimiento, y, con suerte, se ha estado a la altura de ser lo que se propuso ser.

12. References

- D'Ámore, M., Baggio, R., & Valdani, E. (2014). A Practical Approach to Big Data in Tourism: A Low Cost Raspberry Pi Cluster. *Information and Communication Technologies in Tourism 2015*, 169-181.
- Turne K. (1994). Exploiting the m4 Macro Language, from <http://www.cs.stir.ac.uk/~kjt/research/pdf/m4.pdf>
- Furlani, J. L. (1991). Modules: Providing a Flexible User Environment. *Proceedings of the Fifth Large Installation Systems Administration Conference (LISA V)*, 141-152.
- GNU MP 6.1.0. (n.d.). Retrieved February 04, 2016, from <https://gmplib.org/manual/index.html>
- GNU MPFR 3.1.3. (n.d.). Retrieved February 04, 2016, from <http://www.mpfr.org/mpfr-current/mpfr.html>
- Gite, V. (2007, December 18). Ubuntu Linux NFS Server installation and Configuration. Retrieved February 04, 2016, from <http://www.cyberciti.biz/faq/how-to-ubuntu-nfs-server-configuration-howto/>
- Gowtham S. (2007, July 2). HPL Benchmark For Single Processor Machines. Retrieved February 04, 2016, from <http://sgowtham.com/journal/hpl-benchmark-for-single-processor-machines/>
- Gowtham S. (2012, February 27). Rocks 5.4.2 – HPL 2.0 benchmark with GCC 4.1.2. Retrieved February 04, 2016, from <http://sgowtham.com/journal/hpl-2-0-benchmark-with-gcc-4-1-2-on-rocks-5-4-2/>
- Html Documentation GNU MPC 1.0.3. (n.d.). Retrieved February 04, 2016, from <http://www.multiprecision.org/index.php?prog=mpc>
- Installing GCC: Configuration. (n.d.). Retrieved February 04, 2016, from <https://gcc.gnu.org/install/configure.html>
- Kiepert, J. (2013, May 22). Creating a Raspberry Pi-Based Beowulf Cluster. Retrieved February 4, 2016, from http://coen.boisestate.edu/ece/files/2013/05-/Creating.a.Raspberry.Pi-Based.Beowulf.Cluster_v2.pdf
- Leonard, P. (2012, June 18). Parallel Processing on the Pi (Bramble). Retrieved February 04, 2016, from <http://westcoastlabs.blogspot.co.uk/2012/06/parallel-processing-on-pi-bramble.html>
- Modules Software Environment. (n.d.). Retrieved February 04, 2016, from <https://www.nersc.gov/users/software/nersc-user-environment/modules/>
- Network configuration. (n.d.). Retrieved February 04, 2016, from https://wiki.archlinux.org/index.php/Network_configuration

- OpenBLAS User Manual. (n.d.). Retrieved February 04, 2016, from <https://github.com/xianyi/OpenBLAS> Manual
- OpenMPI Project. (n.d.). FAQ: Building Open MPI. Retrieved February 04, 2016, from <https://www.open-mpi.org/faq/?category=building>
- Parallel and distributed processing with John the Ripper. (n.d.). Retrieved February 04, 2016, from <http://openwall.info/wiki/john/parallelization>
- RPi Distributions. (n.d.). Retrieved February 04, 2016, from http://elinux.org/RPi_Distributions
- Raspberry Pi Foundation (n.d.). What is a Raspberry Pi? Retrieved February 4, 2016, from <https://www.raspberrypi.org/help/what-is-a-raspberry-pi/>
- Raspbian Team. (n.d.). Raspbian FAQ. Retrieved February 04, 2016, from <https://www.raspbian.org/RaspbianFAQ>
- Redmond, E. (2012). Building a Riak Cluster on Raspberry Pi. Retrieved February 04, 2016, from <http://basho.com/posts/technical/building-a-riak-cluster-on-raspberry-pi/>
- Whitney, E., & Sprague, M. (2001). Drag your design environment kicking and screaming into the '90s with Modules! Synopsys Users'Group. Retrieved February 4, 2016, from http://modules.sourceforge.net/docs/MC2-whitney_paper.pdf