Grupo de Física y Astrofísica Computacional
Instituto de Física - Universidad de Antioquia

UNIVERSIDAD DE ANTIOQUIA
1803

Apéndice B
Instalación de Gaussian en un Cluster Rocks

Jorge Zuluaga – Instituto de Física – U. de A.

Última actualización: 5 de diciembre de 2006

Este apéndice se escribe con el propósito de ofrecer una guía de instalación rápida de Gaussian® y GaussView® en un Cluster Rocks. Gaussian es un robusto paquete para el cálculo de estructura electrónica ampliamente utilizado en problemas de química computacional y estado sólido. Aunque en su versión más sencilla el paquete corre secuencialmente puede ser utilizado para realizar tareas de computo más complejas que se valen de computación distribuida y paralela. GaussView es una herramienta relacionada con Gaussian y que permite entre otras cosas diseñar gráficamente la estructura electrónica de una molécula, crear archivos de configuración de Gaussian para la molécula, enviar y monitorear trabajos de computo con Gaussian.

ATENCIÓN. *Gaussian y GaussView son herramientas propietario, es decir que para usarlas es necesario adquirir una licencia o un número de licencias correspondientes al número de máquinas en los que se va a utilizar. Antes de instalarla y ejecutarla sobre el cluster asegúrese de que cuenta con los permisos apropiados para hacerlo. Para ello puede dirigirse a la página oficial del producto http://www.gaussian.com*

1. Instalación de los binarios

En su versión más económica Gaussian y GaussView vienen normalmente pre compilados y empacados en un tarball que llamaremos genéricamente aquí 'gaussian03.tar.gz'. El primer paso en la instalación del paquete es la creación del directorio raíz del paquete y la descompresión del tarball en ese directorio. Como se explico en la Guía 3, la instalación de tarballs que incluyen (como en el caso de Gaussian y GaussView bibliotecas que deben ser enlazadas en tiempo de ejecución se hace directamente sobre el sistema de archivos /export/apps:

Comandos 2,1:

```
# mkdir /export/apps/opt/gaussian03
# tar zxvf gaussian03.tar.gz -C /export/apps/opt/gaussian03
```

El procedimiento de descompresión anterior crea 2 directorios nuevos en /export/apps/opt/gaussian03 (en adelante lo llamaremos $GAUSS_DIR): el directorio g03 con los binarios y librerias de gaussian y el directorio gv que contiene lo propio para

GaussView.

Antes de ejecutar Gaussian es necesario crear un grupo de usuarios que tendrán un acceso irrestringido a los archivos y binarios del paquete. La creación de un grupo nuevo se puede realizar modificando directamente el archivo de grupos /etc/group y asignándole al nuevo grupo un GID apropiado.

Así por ejemplo:
/etc/group:

```
...
users:x:100
```

Una vez creado el grupo es necesario asignar a los usuarios este como su grupo. Esto se puede lograr directamente modificando el archivo /etc/passwd o mediante el comando:

Comandos 2,2:

```
# usermod -g 100 fulano
```

Un comando que debe ser ejecutado para todos los usuarios.

Una vez creado el grupo de usuarios debe cambiarse el propietario y grupo de los archivos de Gaussian y GaussView para que puedan ser ejecutados y revisados por usuarios del grupo 'users':

Comandos 2,3:

```
# chown -R root.users /share/apps/opt/gaussian03
# chmod -R og+rx /share/apps/opt/gaussian03
```

Para terminar la instalación de los binarios es necesario cambiar finalmente una variable en el archivo $GAUSS_DIR/gv/init_gv.bash:

$GAUSS_DIR/gv/init_gv.bash:

```
#!/bin/bash
# Environment Initialization Script for:
#
#    GaussView 3.09
#
#    Copyright 1996-2003 Semichem,Inc.
#
# Change this entry only:
export GV_DIR='/.gaussian03/gv'
```

La variable GV_DIR debe ser fijada en su valor correcto “$GAUSS_DIR/gv”.

2. Configuración de variables de ambiente

Para ejecutar Gaussian y o GaussView es necesario fijar un sistema complejo de variables de ambiente y aliases en el sistema operativo. Para ello normalmente cada usuario debe configurar apropiadamente sus archivos de inicialización de sesión (rc files, p.e. .bashrc,

.profile, etc.) para que esas variables y aliases sean reconocidos. Se propone aquí una sencilla estrategia para simplificar esta sobrecarga de trabajo sobre el usuario.

Para implementar la estrategia propuesta, inicialmente se agregan las variables de ambiente críticas al archivo /etc/profile (común a todos los usuarios):

/etc/profile:

```
...
if [ -z "$INPUTRC" -a ! -f "$HOME/.inputrc" ]; then
    INPUTRC=/etc/inputrc
fi

#GAUSSIAN LINES
GAUSS_DIR=/share/apps/opt/gaussian03
PATH=$PATH:$GAUSS_DIR/g03:$GAUSS_DIR/gv/lib:$GAUSS_DIR/gv
GAUSS_SCRDIR=/state/partition1/scratch/$LOGNAME.scratch
GAUSS_EXEDIR=$GAUSS_DIR/g03
GV_DIR=$GAUSS_DIR/gv
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$GAUSS_DIR/g03:$GAUSS_DIR/gv/lib
export LD_LIBRARY_PATH GAUSS_DIR GAUSS_SCRDIR GAUSS_EXEDIR GV_DIR GAUSS_LOGIN
...
```

Además de /etc/profile, donde se incluyen gran variedad de definiciones y comandos, Linux incluye también el archivo /etc/bashrc creado con un propósito similar. Las líneas adicionales en bashrc son, además de las que se incluyeron en /etc/profile:

/etc/bashrc:

```
...
export GAUSS_DIR=/share/apps/opt/gaussian03
export GAUSS_LOGIN=$HOME/.g03login
if [ ! -e $GAUSS_LOGIN ];then GAUSS_LOGIN=/etc/g03login;fi
source $GAUSS_LOGIN
```

Como puede verse arriba hay dos aspectos de especial relavancia en la configuración de las variables de ambiente definidas en profile y bashrc. Primero la existencia de directorios de Scratch para la información temporal creada por la ejecución del programa. En la caso del trabajo en el cluster la recomendación es crear archivos de Scratch en los disco duros locales de cada nodo. Evidentemente no se puede sobrecargar la red del cluster mediante el montaje por NFS de un sistema de archivos que almacene el scratch de todos los usuarios. Para garantizar esto se ha incluido en el código del bashrc un comando que permite crear el directorio de Scratch en cada nodo, en caso de que no se haya creado previamente. Antes de habilitar la creación automática del directorio de scratch del usuario es necesario asegurarse de que el directorio general haya sido creado en el frontend y todos los nodos del cluster:

Comandos 2,4:

```
# mkdir -p /state/partition1/scratch; chmod a+rwx /state/partition1/scratch
# cluster-fork “mkdir -p /state/partition1/scratch; chmod a+rwx /state/partition1/scratch”
```

El otro aspecto especial e intrínseco en los archivos de configuración profile y bashrc es la invocación del script /etc/g03login que debe ser creado independientemente. El contenido del script se presenta a continuación:

/etc/g03login:

```
export g03root=$GAUSS_DIR
export srcdir=$GAUSS_DIR
export scrdir=/state/partition1/scratch/$LOGNAME.scratch
if [ ! -d $scrdir ];then mkdir -p $scrdir;fi
source $g03root/gv/init_gv.bash
csh $g03root/g03/bsd/g03.login
```

Una vez configurados los archivos de inicio de sesión /etc/profile, /etc/bashrc y /etc/g03login el usuario esta listo para usar Gaussian y GaussView en el frontend. Para usar los mismos paquetes en cualquier nodo basta incluir los tres archivos indicados arriba en el sistema de sincronización de archivos de configuración 411 (Guía 3, sección 3.2)

3. Pruebas

Como paso final en la instalación de Gaussian se deben realizar algunas pruebas para asegurarse que todo funcione correctamente.

En primera instancia debemos intentar conectarnos como usuarios al frontend. En caso de funcionar, la conexión no debería devolvernos ningún error implicando el reconocimiento exitoso de las variables de ambiente y la ejecución exitosa también del script de inicialización.

En una segunda instancia debemos ejecutar el binario principal de gaussian g03 para identificar si existe algún problema con el reconocimiento de todas las variables de ambiente:

Comandos 2,5:

```
$ g03
Entering Gaussian System, Link 0=g03
```

Un error de configuración típico produce el siguiente error:

```
g03: error while loading shared libraries: util.so: cannot open shared object file:
No such file or directory
```

En este caso el problema es debido a una inapropiada configuración de la variable LD_LIBRARY_PATH. El programa se corrige revisando con cuidado la sintaxis de esa variable en el /etc/profile.

Como una prueba más rigurosa podemos pedir a Gaussian que calcule la energía de una configuración electrónica elemental, aquella de la molécula de Hidrógeno. Para ello basta con preparar un archivo con la descripción de la molécula y los modelos usados para los cálculos de energía:

h2.com:

```
# opt hf 3-21g

h2

0 1
H
H 1 1.0
```

(Las líneas en blanco deben respetarse).

Para correr el cálculo se ejecuta:

Comandos 2,6:

```
$  g03 h2.com
```

El programa no tiene ninguna salida en pantalla pero produce un archivo con resultados h2.log:

h2.log:

```
86-Linux-G03RevB.02\State=1-SGG\HF=-1.1229598\RMSD=2.311e-13\RMSF=4.89
8e-05\Dipole=0.,0.,0.\PG=D*H [C*(H1.H1)]\\@


ART, GLORY, FREEDOM FAIL, BUT NATURE STILL IS FAIR.

                                          -- BYRON
Job cpu time:  0 days  0 hours  0 minutes  5.7 seconds.
File lengths (MBytes):  RWF=     11 Int=      0 D2E=      0 Chk=      4 Scr=      1
Normal termination of Gaussian 03 at Sun Dec  3 20:33:48 2006.
```

Como prueba final se puede intentar ejecutar el programa con un Scheduler. Para ello podemos usar por ejemplo SGE. A continuación se presenta un script de SGE apropiado para lanzar un trabajo de cálculo de Gaussian:

launch-gaussian.sh:

```
#!/bin/bash
#$ -S /bin/bash
#$ -o gaussian-h2.out
#$ -j y
#$ -cwd
#$ -N H2

gaussdir=/share/apps/opt/gaussian03
export LD_LIBRARY_PATH=$gaussdir/g03:$gaussdir/gv/lib
export g03root=$gaussdir
export srcdir=$gaussdir
export GAUSS_SCRDIR=/state/partition1/scratch/$LOGNAME.scratch
export GAUSS_EXEDIR=$g03root/g03

$g03root/g03/g03 h2.com
```

Para lanzar el proceso se usa qsub:

Comandos 2,7:

```
$ qsub launch-gaussian.sh
```

El proceso se puede monitorear con qstat como se describió en la Guía 2, sección 2.1. La señal de que el proceso culmino con éxito es la aparición del archivo h2.log con un mensaje de completación del proceso.

Finalmente se puede probar la ejecución de GaussView (se debe disponer de modo gráfico):

Comandos 2,8:

$ gview

```
/share/apps/opt/gaussian03/gv/gview: error while loading shared libraries:
libstdc++-libc6.2-2.so.3: cannot open shared object file: No such file or directory
```

Este error se produce por la ausencia de la biblioteca estándar de C++ en la distribución básica de Rocks (cuando se instala Rocks con el Roll Viz este problema no se presenta). La solución natural a este inconveniente es la instalación de esa biblioteca en el sistema. También es posible instalar solamente el archivo con la biblioteca requerida tomándolo directamente de otra instalación de Linux.

Comandos 2,9:

cp -rf libstdc++-libc6.2-2.so.3 /share/apps/lib

A este punto debe recordarse incluir el directorio /share/apps/lib en el archivo de configuración del enlazador dinámico y recargar esta configuración.

Síntesis de comandos en los apéndices

Comando	Explicación
# ifconfig eth0 | head -n1	Encuentra la Mac Address asociada a una NIC
# grep -B 5 -A 5 00:0A:5E:4C:94:1A /etc/sysconfig/hwconf	Encuentra las propiedades de una NIC con una Mac Address específica
# loadkeys /lib/kbd/keymaps/i386/qwerty/i386/es.map.gz	Carga el mapa de caracteres en español desde la línea de comandos
# echo loadkeys /lib/kbd/keymaps/i386/qwerty/i386/es.map.gz >> /etc/rc.local	Permite cargar el mapa de caracteres en español al momento el boot
# service gpm start	Arranca el servicio de mouse en consola de texto
# service xfs restart	Reinicia el servicio de X Fonts
# system-config-display	Inicia el configurador del modo gráfico
# startx	Inicia el modo gráfico
# insert-ethers	Inicia el aplicativo para la asignación de las Ips a los nodos y la creación de la base de datos de nodos
# rocks-console compute-0-0	Monitorea el avance de la instalación en uno de los nodos
# usermod -g 100 fulano	Cambia el GID de un usuario
$ g03	Invoca a Gaussian en modo interactivo
$ g03 h2.com	Invoca a Gaussian usando el archivo con la descripción de la configuración h2.com
$ gview	Invoca a Gauss View