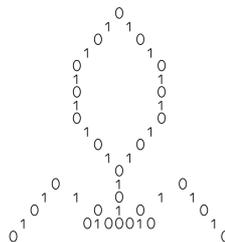




Crear... Crear... Crecer...

UNIVERSIDAD NACIONAL DE RÍO CUARTO
FAC. DE CS. EXACTAS, FCO-QCAS Y NATURALES
DEPARTAMENTO DE COMPUTACIÓN



FUDEPAN
FUNDACIÓN PARA EL DESARROLLO DE LA
PROGRAMACIÓN EN ÁCIDOS NUCLEICOS

TRABAJO FINAL
LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

Manual de compilación e instalación de FuD-BOINC

Autores:
Lucas BESSO
Raúl STRIGLIO

Director:
Lic. Laura TARDIVO

Co-Director:
Daniel GUTSON

Última actualización:
18 de diciembre de 2011

Índice general

I	Manual de instalación y compilación de FuD-BOINC	1
1.	Introducción	2
2.	BOINC	3
2.1.	Dependencias requeridas	3
2.1.1.	Paquetes requeridos por cliente y servidor	3
2.1.2.	Paquetes requeridos por el servidor	4
2.1.3.	Paquetes requeridos por el cliente	4
2.2.	Configuración de MySQL Server	5
2.3.	Descarga del código fuente	5
2.4.	Compilación	6
2.5.	Proyecto BOINC	6
2.5.1.	Dependencias requeridas	6
2.5.2.	Crear un proyecto	6
2.5.3.	Configuración de un proyecto	6
3.	FuD-BOINC	7
3.1.	Dependencias requeridas	7
3.2.	Descarga del código fuente	8
3.3.	Compilación e instalación	8
4.	Aplicación de prueba Parallel-clusterer	10
4.1.	Dependencias requeridas	10
4.2.	Descarga del código fuente	11
4.3.	Compilación e instalación	11
4.4.	Ejecución	12

Parte I

Manual de instalación y compilación de FuD-BOINC

Capítulo 1

Introducción

Este documento describe los pasos que se deben seguir para poder compilar la librería FuD con la capa de distribución FuD-BOINC. Para lograr ésto, se explica cómo descargar BOINC, cómo compilarlo y cómo crear un proyecto de computación voluntaria en donde se puedan correr las aplicaciones desarrolladas con FuD-BOINC. Por último, se explican los pasos a seguir para descargar y compilar FuD con las librerías BOINC ya compiladas.

El manual pretende ser un medio simple en donde se integren los pasos necesarios para comenzar a utilizar FuD-BOINC por lo que si se desea extender algunos conceptos y/o instrucciones aquí detalladas recomendamos consultar la información oficial que será incluida con cada sección.

Capítulo 2

BOINC

2.1. Dependencias requeridas

Para poder compilar las librerías de BOINC es necesario resolver los requisitos previos del framework.

La información de esta sección está basada de la wiki oficial de BOINC¹.

A continuación se especifican los paquetes necesarios para compilar BOINC basado en sistemas Unix/Linux.

2.1.1. Paquetes requeridos por cliente y servidor

- m4
- make
- autoconf
- automake1.9
- gcc-4.1
- gcc
- g++-4.1
- pkg-config
- libtool
- subversion
- vim

¹<http://boinc.berkeley.edu/trac/wiki/ServerIntro#cookbook-debian40>

2.1.2. Paquetes requeridos por el servidor

- apache2-mpm-prefork
- libapache2-mod-php5
- mysql-client-5.0
- mysql-server-5.0
- php5-mysql
- php5-cli
- php5-gd
- phpmyadmin
- python-mysqldb
- libmysql++-dev
- libssl-dev

2.1.3. Paquetes requeridos por el cliente

- libssl-dev
- libglut3-dev
- glut3-dev
- libglui-dev
- libglitz-glx1-dev
- libsdl1.2-dev
- libcurl3-dev
- freeglut3
- freeglut3-dev
- libsm-dev
- libice-dev
- libxmu-dev

- libxi-dev
- libx11-dev
- libjpeg62-dev
- libgtk2.0-0
- libgtk2.0-0-dev

2.2. Configuración de MySQL Server

Para definir una nueva contraseña del usuario root hacer lo siguiente:

```
$ mysqladmin -h localhost -u root password mysqlrootpw {or own}
```

Para crear un nuevo usuario en la base de datos hacer lo siguiente:

```
$ mysql -h localhost -u root -p  
> GRANT ALL ON *.* TO 'boincadm'@'localhost';  
> SET PASSWORD FOR 'boincadm'@'localhost'='';
```

Los permisos deberían ser limitados a la base de datos del proyecto después. Aquí, la definición de una contraseña vacía simplifica el proceso de instalación la cual luego puede ser modificada.

2.3. Descarga del código fuente

El código fuente de BOINC se encuentra almacenado en un repositorio de Subversion (SVN). Correr el siguiente comando para obtener la última versión estable:

```
svn co http://boinc.berkeley.edu/svn/trunk/boinc
```

Para más información consultar la wiki oficial de BOINC² que menciona este tema.

²<http://boinc.berkeley.edu/trac/wiki/SourceCode>

2.4. Compilación

2.5. Proyecto BOINC

2.5.1. Dependencias requeridas

2.5.2. Crear un proyecto

2.5.3. Configuración de un proyecto

Capítulo 3

FuD-BOINC

3.1. Dependencias requeridas

Para poder compilar la librería FuD-BOINC es necesario resolver los requisitos de la misma. A continuación se especifican los paquetes necesarios para compilar la librería sobre un sistemas Unix/Linux.

Dependencias de FuD:

- **librería Mili:**

El código fuente de Mili se encuentra almacenado en un repositorio de Subversion (SVN). Correr el siguiente comando para obtener la última versión estable:

```
http://code.google.com/p/mili/downloads/list
```

Para conocer los pasos de instalación, consultar el archivo README

- **librería Boost 1.42 o superior:**

Descargar e instalar los paquetes:

- libboost-system
- libboost-thread

Dependencias de FuD-BOINC:

Descargar e instalar los paquetes:

- glibc
- libssl-dev

- libmysqlclient-dev

Se deben tener correctamente instaladas las siguientes librerías de BOINC:

- librerías de BOINC
- libboinc
- libboinc_api
- libsched
- libboinc_crypt

3.2. Descarga del código fuente

El código fuente de FuD-BOINC se encuentra almacenado en un repositorio de Subversion (SVN). Correr el siguiente comando para obtener la última versión estable:

```
svn checkout https://fud.googlecode.com/svn/branches/boinc FuD-BOINC
```

Si bien el código fuente de FuD-BOINC es descargado desde un branch, éste en un futuro va a estar disponible en la versión trunk de FuD.

3.3. Compilación e instalación

Una vez descargado el código fuente, se debe proceder con los siguientes pasos para la compilación e instalación de la librería FuD-BOINC:

1. crear un directorio donde se compilará el código fuente.
2. ingresar al nuevo directorio.
3. ejecutar el comando:

```
cmake -Dmiddleware=boinc [options] PATH_TO_FuD-BOINC_Source
```

Debemos destacar que la opción **-Dmiddleware=boinc** debe ser utilizada para compilar con la capa de distribución implementada con BOINC. Si ésta opción no se la especifica, se compilará con la implementación por defecto(ASIO).

A continuación se destacan las opciones de instalación de FuD-BOINC:

- **-Dboinc_source=PATH:** utilizar ésta opción para especificar el PATH del directorio de código fuente de BOINC. Por defecto PATH = “home/boinc”
- **-DCMAKE_BUILD_TYPE=Debug:** especificar ésta opción si se desea compilar con el flag para dar soporte a la depuración.
- **-DCMAKE_COVER_ON=on:** especificar ésta opción si se desea compilar con los flags para dar soporte a la cobertura de código.

Capítulo 4

Aplicación de prueba Parallel-clusterer

4.1. Dependencias requeridas

Para poder compilar la aplicación Parallel-clusterer es necesario resolver los requisitos de la misma. A continuación se especifican los paquetes necesarios para compilar la aplicación sobre un sistemas Unix/Linux.

- **Biopp:**

El código fuente de Biopp se encuentra almacenado en un repositorio de Mercurial (hg). Correr el siguiente comando para obtener la última versión estable:

```
hg clone https://code.google.com/p/biopp/
```

Luego reemplazar el archivo Makefile del directorio descargado por el que se provee en el directorio CD-Tesis/Makefiles/Biopp.

Para su instalación: ejecutar el comando “make” y luego “make install”.

- **Prot-filer:** El código fuente de Prot-filer se encuentra almacenado en un repositorio de Mercurial (hg). Correr el siguiente comando para obtener la última versión estable:

```
hg clone https://code.google.com/p/prot-filer/
```

Luego reemplazar el archivo Makefile del directorio descargado por el que se provee en el directorio CD-Tesis/Makefiles/Prot-filer.

Para su instalación: ejecutar el comando “make” y luego “make install”.

- **Getopt_pp:** El código fuente de Getopt_pp se encuentra almacenado en un repositorio de Mercurial (hg). Correr el siguiente comando para obtener la última versión estable:

```
hg clone https://code.google.com/p/getoptpp/
```

Para su instalación: seguir los pasos indicados en el archivo README. Debemos destacar que se deben compilar librerías estáticas.

- **xdrfile:** Descargar la librería desde la siguiente dirección: <http://download.fedora.redhat.com/pub/fedora/linux/releases/15/Everything/source/SRPMs/xdrfile-1.1-4.fc15.src.rpm>.

Para su instalación: seguir los pasos indicados en el archivo INSTALL.

- **feca:**

El código fuente de feca se encuentra almacenado en un repositorio de Mercurial (hg). Correr el siguiente comando para obtener la última versión estable:

```
hg clone https://code.google.com/p/feca/
```

Luego reemplazar el archivo Makefile del directorio descargado por el que se provee en el directorio CD-Tesis/Makefiles/feca.

Para su instalación: ejecutar el comando “make” y luego “make install”.

4.2. Descarga del código fuente

El código fuente de la aplicación Parallel-clusterer se encuentra almacenado en un repositorio de Subversion (SVN). Correr el siguiente comando para obtener la última versión estable:

```
svn checkout http://parallel-clusterer.googlecode.com/svn/trunk/  
parallel-clusterer
```

4.3. Compilación e instalación

Para compilar la aplicación, primero se debe reemplazar el archivo “Makefile” original por el que se provee en el directorio CD-Tesis/Makefiles/Parallel-Clusterer perteneciente al CD de éste proyecto. Luego se debe ejecutar el comandos “make” para compilar la aplicación y “make install” para instalar la aplicación

4.4. Ejecución

Para ejecutar la aplicación compilada con FuD-BOINC, primero se debe agregar la aplicación al proyecto BOINC, siguiendo los pasos especificados en la sección [2.5.3](#). Luego, desde el directorio del proyecto, ejecutar el siguiente comando:

```
bin/clusterer -i [input\_file] -f compressed -s [output\_file] -a full\ cache.
```

input_file es el archivo que se provee en el directorio CD-Tesis/Parallel-clusterer/input_files/.

output_file es el nombre del archivo de salida donde la aplicación escribirá las estadísticas resultantes.