

Proyecto de Grado

Instalación, Implementación y Uso de las librerías Boost y su aplicación en recorrido de grafos para la supercomputadora Apolo

Autor:

Sergio Andrés Monsalve Castañeda

Código: 200410061010

smonsal3@eafit.edu.co

Asesor:

Juan Guillermo Lalinde Pulido

jlalinde@eafit.edu.co

2619500 ext 9588

13 de noviembre de 2013



Índice general

1. Introducción	2
1.1. Objetivos	2
1.2. Alcance y productos	2
1.3. Usabilidad	2
2. Marco de Referencia	3
2.1. Computación Paralela	3
2.1.1. Apolo	3
2.2. C++	3
2.2.1. Metaprogramación en C++	3
2.3. MPI	3
2.4. Boost	3
2.5. Grafos	3
2.5.1. PBGL	3
3. Instalación	4
3.1. Virtualización de un cluster	4
3.1.1. Configuración del Nodo Maestro	4
3.1.2. Configuración de los Nodos Esclavos	6
3.2. Preguntas Frecuentes:	7
3.3. MPI	7
3.3.1. OpenMPI	7
3.3.2. Mpich	7
3.4. Boost Graph Library	7
3.4.1. Parallel Boost Graph Library (Parallel BGL)	7
3.4.2. Instalacion de Boost en Apolo	7
3.4.3. makefile	7
4. Implementación	8
5. Resultados	9
5.1. Sin paralelizar	9
5.2. 10 Nodos	9
5.3. 100 Nodos	9
6. Trabajo Futuro	10

7. Glosario	11
A. Repositorio del código	13

Resumen

Existe una gran cantidad de problemas que debido a su complejidad no es posible resolver, sea porque requieren mucho tiempo o la complejidad de estos es tal que tardarían tanto tiempo que su estudio no es posible. Adicionalmente cada vez surgen nuevos problemas, aun mas complejos. Con el crecimiento de la capacidad computacional ya es posible abordar algunos de ellos que antes hubieran tomado mucho tiempo o simplemente no hubiese sido posible resolver.

Dentro de esta amplia gama de problemas, existen unos particulares que pueden ser modelados mediante la teoría de grafos, una gran herramienta que ofrece las ciencias de la computación.

Aunque los algoritmos para grafos tienen una complejidad alta en general, existen algoritmos paralelos. En este proyecto se pretende instalar en Apolo, el supercomputador de EAFIT, la infraestructura básica para que un programador, utilizando las librerías BOOST para C++, pueda desarrollar rápidamente aplicaciones que aprovechen su capacidad computacional.

De esta manera se pretende beneficiar a la Comunidad Científica, Comunidad académica, Estudiantes de Ingeniería de Sistemas, Estudiantes de Ingeniería Matemática y en General a quien necesite realizar cómputo de grafos en paralelo utilizando Boost, o requiera de una introducción a este tema.

Agradecimientos

En primer lugar quiero agradecer a Juan Guillermo Lalinde quien desde el inicio de me carrera me inspiro y apoyo en grandes retos académicos, sin su ayuda hoy sería un cocinero. A Juan David Pineda por toda su paciencia para explicarme el funcionamiento de Apolo. Por ultimo agradezco a John Jairo Silva, Alejandro Gómez, Mateo Gómez, Jaime Pérez y John Mario Gutiérrez por sus multiples colaboraciones.

Capítulo 1

Introducción

1.1. Objetivos

- Construir un manual para la instalación de las librerías de Boost.
- Generar Documentación sobre el manejo de las librerías de Boost.
- Crear códigos de ejemplo para la ejecución de las librerías de Boost.

1.2. Alcance y productos

- Descripción de las Librerías de Boost.
- Librerías de Utilización de Grafos Instaladas en Apolo.
- Manual de Instalacion de librerias Para Boost.
- Ejemplo en código documentado y manual de como hacer un programa utilizando Boost para grafos.
- Introducción al manejo de Grafos con Boost (representación conceptual).

1.3. Usabilidad

Utilizar los GPS de los taxis para hacer análisis de la malla vial de la ciudad. Movilización y Rutas óptimas para el recorrido de Bomberos, Ambulancias, Policía para la atención de emergencias y desastres

Movilidad, Seguridad, Educación, Tecnología.

Capítulo 2

Marco de Referencia

2.1. Computación Paralela

[2] [3]

2.1.1. Apolo

2.2. C++

2.2.1. Metaprogramación en C++

2.3. MPI

2.4. Boost

The Parallel Boost Graph Library is an extension to the Boost Graph Library (BGL) for parallel and distributed computing. It offers distributed graphs and graph algorithms to exploit coarse-grained parallelism along with parallel algorithms that exploit fine-grained parallelism, while retaining the same interfaces as the (sequential) BGL. Code written using the sequential BGL should be easy to parallelize with the parallel BGL. Visitors new to the Parallel BGL should read our architectural overview.[1]

2.5. Grafos

2.5.1. PBGL

Capítulo 3

Instalación

3.1. Virtualización de un cluster

La siguiente es la descripción de los pasos a seguir para la creación de un ambiente de pruebas paralelo virtualizado que simula el cluster de Apolo.¹

3.1.1. Configuración del Nodo Maestro

1. Descargue e Instale VirtualBox Manager.²
2. Descargar la imagen del Master.³
3. Descargue el “Extension Pack”.⁴
4. Una vez instalado VirtualBox en su computador, proceda a instalar el Extensión Pack:
 - En VirtualBox acceda al menú Archivo → Preferencias → Sección Extensiones → Proceda a instalar el “Extension Pack”.
 - Vaya a la sección Red → Adicione una red Sólo Anfitrión. Esta debe ser la primera interfaz de red de Sólo Anfitrión que se crea, si ya está creada por favor no cree otra adicional.
 - Haga click en aceptar para finalizar la operación.
5. En VirtualBox, importe desde el Menú → Importar Appliance. Deje la configuración por defecto y **no** reinicialice la dirección MAC.
6. Señale la máquina virtual llamada “Master” y vaya a la configuración y revise la siguiente configuración en esta:

¹Estas instrucciones fueron probadas en un ambiente de Linux Fedora 18.

²<https://www.virtualbox.org/wiki/Downloads>

³<http://goo.gl/8eTJOr>

⁴<https://www.virtualbox.org/wiki/Downloads>

- En la sección Sistema, pestaña Procesador, debe tener dos procesadores y habilitado PAE/NX.
 - En la sección Red, pestaña Adaptador 1, deberá estar configurado como NAT.
 - En la pestaña Adaptador 2 deberá estar en Adaptador Sólo-Anfitrión y el nombre deberá ser `vboxnet0`⁵.
 - Acepte todos los cambios.
7. Señale la máquina Master e iníciela.
8. Una vez iniciada la máquina virtual del Master proceda a crear una nueva máquina virtual como Nodo Trabajador a partir de los siguientes pasos:
- Haga click en Crear Nueva Máquina.
 - El nombre será *compute-0-0*.
 - El tipo será Linux.
 - La versión será Red Hat de 64 bits.
 - Click en siguiente.
 - Asigne 1024 Megabytes de Memoria RAM⁶.
 - Cree el disco duro con 30 gigas de espacio, recuerde que esto se asignará dinámicamente. El tipo de disco duro será VDI y dinámicamente asignado.
9. Una vez creado el nodo trabajador se procede a configurarlo a partir de los siguientes pasos:
- Señalar la máquina *compute-0-0* y dar click en Configurar.
 - En la pestaña Sistema, asegúrese de que tenga la cantidad de memoria RAM correcta
 - Deshabilite el *floppy disk*
 - habilite la red como dispositivo de *booteo* y además súbalo como primer dispositivo, sólo deberá quedar la tarjeta de red como primera opción y como segunda el disco duro de la máquina virtual, de esta manera nos aseguramos que esta máquina virtual pueda instalarse automáticamente de manera desatendida, por esta razón el clúster es escalable.
 - En la pestaña Procesador asegúrese de que hay dos procesadores y está habilitado PAE/NX.
 - En la sección Red deberá configurar sólo la primera interfaz de red y deberá estar configurada como Sólo-Anfitrión y deberá tener `vboxnet0`.

⁵Tenga en cuenta que este adaptador se llama `vboxnet0` en Linux, en Windows tendrá otro nombre, lo más importante es que sea la primera interfaz de red de Sólo Anfitrión, ya que sino dará lugar al problema de reasignación de interfaces de red dentro del nodo Master, en otras palabras, asignará las interfaces `eth2` y `eth3` en vez de asignar `eth0` y `eth1` a la NAT y a la Sólo Anfitrión respectivamente.

⁶La cantidad de memoria RAM asignada será proporcional a la que tenga el sistema en el cual están las máquinas virtuales. Igual con el disco duro y los procesadores, sin embargo, para efectos de ver la paralelización en memoria compartida se recomienda tener 2 procesadores por máquina

- Acepte los cambios.
10. Repita los pasos para crear otro nodo si lo considera necesario para crear el *compute-0-1*, siempre y cuando la computadora que usted tiene soporte una tercera máquina virtual ejecutandose al tiempo que el Nodo Master y el *compute-0-0*
 11. Ingrese en el Nodo Master como usuario `root` y la contraseña es `apolito123!`
 12. Abra una consola
 13. Ingrese el comando `insert-ethers`
 14. Escoja **Compute**. Aparecerá una interfaz de consola mostrándole los nodos agregados en la medida que se les vaya asignando IP por medio de DHCP y una imagen de Linux para instalar con PXE.
 15. Encienda de uno en uno los Nodos trabajadores que haya creado, empezando por el *compute-0-0* y así sucesivamente. Observará que aparece en la interfaz `insert-ethers` la dirección MAC del Nodo Trabajador, se le asignará el nombre `compute-0-0` y si aparece un símbolo de asterisco es porque recibió exitosamente la imagen para la instalación de Linux.
 16. Una vez de que los nodos se termine de instalar automáticamente salga de la interfaz de `insert-ethers` en el Master con la tecla F8.
 17. Ingrese al directorio especificado con el siguiente comando: `cd /export/apps/installers`
 18. Descargue el instalador del comando `htop` con la siguiente instrucción: `wget http://goo.gl/TDWExw`
 19. Instale `htop` en el nodo Master con la siguiente instrucción: `rpm -ivh htop*.rpm`
 20. Ahora instale masivamente `htop` en el resto del clúster con el siguiente comando: `rocks run host 'rpm -ivh /share/apps/installers/htop*.rpm'`
 21. El cluster está completo.

3.1.2. Configuración de los Nodos Esclavos

1. Una vez que se tiene el nodo Master funcionando y por lo menos un nodo trabajador como el *compute-0-0* se procede a realizar los siguientes pasos de configuración:
 - Adicione un usuario sin privilegios con los siguientes comandos⁷:
 - `adduser jdpinedac` Para crear un usuario sin privilegios.
 - `passwd jdpinedac` Para cambiar la contraseña del usuario.
 - `rocks sync users` Para sincronizar el usuario creado en todo el cluster, este usuario estará creado tanto en el nodo master como en los nodos trabajadores.
 - `su - jdpinedac` Para que el usuario `root` se convierta en el usuario sin privilegios. Se le harán algunas preguntas de contraseñas, déjelas vacías presionando la tecla *enter* varias veces.

⁷En adelante el usuario de ejemplo será `jdpinedac`, pero usted podrá asignar el nombre de usuario que desee

3.2. Preguntas Frecuentes:

[Posibles Problemas]

- Los nodos de trabajo no inician Correctamente: Es posible que estos hayan sido apagados de manera incorrecta. siga el procedimiento desde el numeral X hasta el Y de “ASDF”
- Verifique que su computador puede virtualizar: Ingrese a la BIOS y verifique que esta activada la opcion de Virtualización de Hardware.
- Si su caso no se encuentra aqui listado contacte al autor

Imagen de apolo. Definir los nodos.

3.3. MPI

3.3.1. OpenMPI

3.3.2. Mpich

3.4. Boost Graph Library

build mpi build GraphParallel build serilization

Boost and Eclipse configuración

3.4.1. Parallel Boost Graph Library (Parallel BGL)

Seriallization

Regex

Parallel

MPI

3.4.2. Instalacion de Boost en Apolo

3.4.3. makefile

Cómo ejecutar el script de instalación

que se debe modificar en el makefile[10]

instalar rpm de Boost: Cuando se formatea o reinicia máquina Cuando se instala normalmente.

Capítulo 4

Implementación

Ejemplo Programa Corriendo en Apolo

Programa Normal.

Ejecturar

Capítulo 5

Resultados

5.1. Sin paralelizar

5.2. 10 Nodos

5.3. 100 Nodos

Capítulo 6

Trabajo Futuro

Capítulo 7

Glosario

HPC: (High Performance Computing) Computación de alto desempeño.

Data Center:

Bibliografía

- [1] Boost c++ libraries. <http://www.boost.org/>.
- [2] Introduction to Parallel Computing.
- [3] Software Carpentry: Lessons (Version 4.0).
- [4] Andrei Alexandrescu. *Modern C++ Design: Generic Programming and Design Patterns Applied*. C++ in-depth series. Addison-Wesley, 2001.
- [5] The Boost and Graph Library. *The Boost Graph Library*.
- [6] K. Czarnecki and U. Eisenecker. *Generative programming: methods, tools, and applications*. Addison Wesley, 2000.
- [7] George Em Karniadakis and Robert M Kirby II. in C ++ and MPI A seamless approach to parallel algorithms and their implementation.
- [8] BW Kernighan, DM Ritchie, and P Ekelint. *The C programming language*, volume 8. 1988.
- [9] B. Stroustrup. *The C++ Programming Language*. Pearson Education, 2013.
- [10] Kurt Wall. *Programación en Linux con ejemplos*. Prentice-Hall, 2000.

aca [6] [1] [9] [4] [10] [5] [7] [8]

Apéndice A

Repositorio del código

Para una copia del código utilizado dirigirse a: <https://github.com/smonsalve/tesis.git>

OVA de la maquina virtual con Rocks instalado y dos nodos.

Input data para Grafo (X Vertices y Y Aristas).

Makefile de Instalación.