

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

semonsalve@gmail.com

Asesor:

Juan Guillermo Lalinde Pulido

jlalinde@eafit.edu.co

2619500 ext 9588

25 de septiembre 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. Apolo	3
2.2. Grafos	3
2.3. MPI	3
2.4. Boost	3
2.5. Metaprogramación	3
2.5.1. Metaprogramación en C++	3
3. Instalación	4
3.1. Virtualización de un cluster	4
3.2. MPI	4
3.2.1. OpenMPI	4
3.2.2. Mpich	4
3.3. Boost Graph Library	4
3.3.1. Parallel Boost Graph Library (Parallel BGL)	5
3.3.2. Instalacion de Boost en Apolo	5
3.3.3. makefile	5
4. Implementación	6
5. Resultados	7
5.1. Sin paralelizar	7
5.2. 10 Nodos	7
5.3. 100 Nodos	7
5.4. Trabajo Futuro	7
6. Glosario	8
A. Repositorio del código	10

Resumen

El crecimiento de la capacidad computacional abre nuevos campos de aplicación.

La teoría de grafos es una de las herramientas más importantes que ofrece las ciencias de la computación para la resolución de diferentes tipos de problemas.

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 en diversos temas.

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. Apolo

Super Computadores

2.2. Grafos

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. Metaprogramación

2.5.1. Metaprogramación en C++

Capítulo 3

Instalación

3.1. Virtualización de un cluster

Apolo

Hibrido

Nodos

Mira home ÑOÑA!!!

imagen de apolo. Definir los nodos.

3.2. MPI

3.2.1. OpenMPI

3.2.2. Mpich

3.3. Boost Graph Library

build mpi build GraphParallel build serilization

Boost and Eclipse configuración

3.3.1. Parallel Boost Graph Library (Parallel BGL)

Serialization

Regex

Parallel

MPI

3.3.2. Instalacion de Boost en Apolo

3.3.3. makefile

Cómo ejecutar el script de instalación

que se debe modificar en el makefile[8]

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

5.4. Trabajo Futuro

Capítulo 6

Glosario

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

Data Center:

Bibliografía

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

aca [4] [1] [7] [2] [8] [3] [5] [6]

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.