Programa del curso

Semestre 2018-20

Nombre del curso:	Herramientas de programación de alto rendimiento
Créditos:	4
Profesor:	Pablo Figueroa (pfiguero@uniandes.edu.co)
Versión PDF	Click Aquí

Objetivos

- Usar herramientas de desarrollo y programación de alto rendimiento, tanto en ejemplos de laboratorio como en proyectos reales.
- Usar herramientas de comprensión de código de proyectos de interés.
- Usar nuevas herramientas de programación en paralelo, basadas en las tecnologías multicore de CPUs y GPUs.
- Conocer algoritmos que hacen uso intensivo de la GPU.

Metodología

El contenido del curso se divide en temas teóricos y temas de tecnología. La teoría viene del libro guía de la materia y se cubren 4 tecnologías: Programación en línea de comandos, CUDA, OpenMPI y Python. Los trabajos individuales de los estudiantes cubren otras tecnologías y ejercicios de práctica.

Evaluación

- 40%: Descripción de los ejemplos (4)
- 20%: Análisis de otra tecnología no cubierta en el curso (en grupo)
- 20%: Competencia en rendimiento
- 20%: Trabajo final (en grupo)

Descripción de ejemplos

La descripción de un ejemplo es un documento que muestra el conocimiento adquirido sobre el lenguaje, sobre el ejemplo en cuestión y sobre su ejecución con base en los conceptos básicos. El documento tiene la siguiente tabla de contenido:

- Clasificación, de acuerdo a la taxonomía de Flynn
- Descripción de tareas, unidades de ejecución, elementos de procesamiento, balanceo de carga, sincronización,
- Análisis de tiempos
- Análisis de comunicación
- Patrones aplicados (hasta los vistos en el curso al momento de la evaluación)
- Análisis de líneas de código (Diigo)
- Bitácora de pruebas y ejemplos adicionales
- Posibles mejoras (idealmente acompañado con un patch)

Análisis de otra tecnología

Hay muchas tecnologías para desarrollo de aplicaciones de alto rendimiento, y el objetivo en este caso es analizar otras tecnologías de manera similar a la forma en la cual se analiza en el curso. El análisis debe incluir

- Una breve introducción de la tecnología y su historia
- Descripción de la tecnología (extensión a un lenguaje, lenguaje, framework,)
- Descripción de las herramientas de trabajo y análisis de resultados
- Un ejemplo detallado
- Bitácora del análisis de la tecnología
- Videos cortos (no mas de 10 minutos) con la presentación de dicha tecnología

Esta es una lista inicial de tecnologías. Uds pueden proponer otras.

- C++AMP
- PowerShell (Windows)
- OpenACC
- OpenMP
- Perl, en el contexto de problemas de bioinformática
- Paralelismo y concurrencia en Java
- OpenCL
- Aparapi (ATI)
- Direct Compute (Windows)

Puede verse también el problema desde el punto de vista de software existente de alto rendimiento, implementado en otras tecnologías. Ejemplos son:

- Simuladores de clima
- Genomics, Proteomics
- SETI at Home
- Boinc
- Fold it
- Ejemplos específicos de GPGPU

Competencia en rendimiento

Vamos a competir en mejorar el rendimiento del código ejemplo de GPU, el simulador desarrollado por Diego Rodriguez. el trabajo consiste en hacer modificaciones al código de Diego, medir el rendimiento de los cambios y entregar al final un informe que contenga:

El acceso a Gambita es restringido, por lo que se espera que uds. tomen turnos para su uso.

Trabajo final

Se propone la solución a un problema de alto rendimiento, en cualquier dominio y con cualquier herramienta o tecnología. Algunos problemas sugeridos son:

- Simulación distribuida de pilotaje de aviones
- Simulación distribuida de tráfico urbano
- Visualización en astrofísica
- (Omar Lopez) Mejora de desempeño de un código de Smoothed Particle Hydrodynamics (SPH) en GPU
- (Omar Lopez) Visualización paralela de un vuelo virtual

Bibliografía

- Timothy G. Mattson, Beverly A. Sanders, Berna L. Massingill. Patterns for Parallel Programming. Addison-Wesley Professional; 1 edition (September 25, 2004). Libro Guia
- UMunshi, Gaster, Mattson, Fung, Ginsburg. OpenCL Programming Guide. Addison-Wesley. 2012
- Sanders, Kandrot. CUDA By Example. Addison-Wesley. 2011