



Universidad
de Navarra

Programación avanzada (MC2)

Guía docente 2024-25

PRESENTACIÓN

La necesidad de utilizar cada vez más la potencia de los ordenadores para realizar tareas de cálculo y tratamiento de datos, ha dado lugar de forma natural a lo que llamamos computación avanzada ("High performance computing" en inglés).

Dependiendo de la demanda, el presente curso puede tener dos formatos; en una de las variantes, veremos dos de las estrategias más utilizadas para aumentar la potencia y la velocidad de nuestros ordenadores. Más precisamente, trabajaremos la programación en paralelo de programas distribuidos. En la segunda variante del curso, trabajaremos en la resolución de problemas científicos usando Python; dejando para el final una breve introducción a las redes neuronales.

En lo que corresponde a la programación en paralelo, se puede utilizar la tarjeta gráfica de nuestros ordenadores para ejecutar parte de las operaciones que necesitamos realizar ("GPU computing"). La primera parte del curso se centrará sobre cómo podemos utilizar de forma práctica esta tarjeta gráfica para realizar cálculos científicos y análisis de datos.

Por otra parte, se puede sub-dividir un problema numérico en varias partes que cada uno de los procesadores pueden atacar de forma independiente. En el caso de tener un problema distribuido, las comunicaciones entre los procesadores se realizan a través de unas librerías (OpenMP/MPI) que permiten el envío y la recepción de datos entre los distintos procesadores para su ejecución. En esta introducción veremos los elementos que configuran estas librerías y veremos aplicaciones prácticas para su buen uso.

En la variante del curso correspondiente a la resolución de problemas científicos usando Python, avanzaremos afianzando conceptos mediante la discusión sucesiva de problemas que irán aumentando en complejidad, hasta llegar a la construcción de una red neuronal sencilla.

La asignatura es esencialmente práctica y el alumno tendrá que realizar algunos ejercicios de programación con la supervisión de los profesores.

3 ECTS. Módulo 3 (optativa). Segundo semestre. Se imparte en español (bibliografía en inglés).

Es recomendado disponer de un ordenador portátil para instalar los programas necesarios.

Profesor responsable de la asignatura: Reinaldo García García (responsable).

Aula: 1 edificio Los Castaños

Horario: [Calendario del Máster](#)

- **Requerimiento:** conocimiento del Lenguaje de Programación C y/o de Python a nivel elemental.

RESULTADOS DE APRENDIZAJE (Competencias)



	Materia 3.1 Optativas
	Procesamiento de imágenes
	COMPETENCIAS BÁSICAS
CB6	Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo o aplicación de ideas, a menudo en un contexto de investigación.
CB7	Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio.
CB10	Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.
	COMPETENCIAS GENERALES
CG3	Conocer los principales problemas que se presentan en la adquisición y tratamiento de datos experimentales y cómo darles respuesta.
CG4	Comunicar tanto de manera oral como escrita un tema o datos de investigación en el área de las ciencias experimentales.
	COMPETENCIAS ESPECÍFICAS
CE5	Aplicar los métodos computacionales de procesamiento de datos a un problema científico particular de la disciplina de interés para el estudiante.



CE6	Diseñar un experimento científico para que sea rico en información, recogiendo gran cantidad de datos de manera estructurada que faciliten su procesamiento posterior.
CE7	Integrar en el análisis científico datos obtenidos de fuentes heterogéneas.
CE8	Adquirir datos (bien en el laboratorio, o bien mediante minería on-line), organizarlos, filtrarlos, procesarlos, representarlos y refinarlos.
CE9	Extraer información de los datos con técnicas computacionales siguiendo un método científico.
CE10	Presentar los datos experimentales y la información científica de manera que se comuniquen de manera eficiente y fidedigna.
	COMPETENCIAS ESPECÍFICAS DE OPTATIVIDAD
CEOP1	Obtener mediciones cuantitativas de imágenes digitales obtenidas en el laboratorio.

PROGRAMA

Parte I (Cuda-GPU)

1) Introducción a la programación sobre tarjetas gráficas (GPU), CUDA-NVIDIA en C/C++. Estructura de programación en CUDA-NVIDIA. Declaración, reserva e inicialización de memoria en host (CPU) y device (GPU). Transferencia de datos host-device-host. Funciones y Kernels.

2) Bloques (Blocks) y hilos (Threads). Recursos asignados a los bloques.

Sincronización entre hilos, orden de ejecución. Tipos de memoria, la memoria como limitante.

3) Desarrollo de kernels en CUDA-NVIDIA. Compilación y ejecución de algoritmos sencillos. Manipulación de matrices de datos. CUDA-toolkit.



Universidad
de Navarra

4) Modelación de Elementos Discretos (DEM) y dinámica molecular usando CUDA-NVIDIA. Desarrollo de algoritmos de promediación espacial y promediación temporal usando CUDA-NVIDIA.

5) Introducción al desarrollo de algoritmos CUDA-NVIDIA usando MATLAB. Generalidades y ejemplos.

Parte II: Manejo de la librería de calculo paralelo MPI ("Message Passing Interface").

1) Introducción a la utilización de la librería MPI dentro de unos programas en C /C++. Necesidad de la computación paralela en Ciencias y Ciencias de datos. El problema de la comunicación entre procesadores. Instalación de la librería MPI en ordenadores personales.

2) Programa básico en MPI. Primer ejemplo "Hello World" programa. Envío y recepción de información (datos) entre procesadores. Tipos de comunicaciones.

3) Una aplicación numérica de la programación paralela. Integración numérica con el método de trapecio.

4) Una segunda aplicación. El camino aleatorio.

5) Tercera aplicación la ecuación de la difusión en un sistema bidimensional.

6) Discusiones de posibles proyectos realizables en programación paralela.

El programa práctico se adaptará a cada estudiante.

Es recomendado acudir a las clases con un ordenador portátil.

ACTIVIDADES FORMATIVAS

EVALUACIÓN

La evaluación se realizará de forma continua mediante la realización de diversos proyectos pequeños durante todo el curso. Dichos trabajos se complementarán con un trabajo final de mayor complejidad. Los estudiantes podrán elegir (previa consulta con los profesores de la asignatura) el proyecto que deseen para el trabajo final y lo desarrollarán bajo nuestra supervisión.

HORARIOS DE ATENCIÓN

Dr. Reinaldo García García (regarciag@unav.es)

- Despacho O171 Edificio Castaños.
- Horario de tutoría:



Universidad
de Navarra

BIBLIOGRAFÍA

Hay que tener en cuenta que en esta asignatura muchos recursos no están en forma de libro, sino en herramientas y contenidos digitales que serán presentados durante el curso: por ejemplo, la ayuda en línea de las interfaces de programación. En la biblioteca (UNAV) se puede consultar las siguientes referencias.

General:

Introduction to Scientific Computing and Data Analysis [Localízalo en la Biblioteca](#) [Recurso electrónico]

Holmes, Mark H.

Cham : Springer International Publishing, 2016.

CUDA-GPU:

GPU Computing and Applications [Localízalo en la Biblioteca](#) [Recurso electrónico]

edited by Yiyu Cai, Simon See.

Singapore : Springer Singapore, 2015.

MPI:

Using MPI: portable parallel programming with the Message-Passing-Interface [Localízalo en la Biblioteca](#) [Recurso electrónico]

Gropp, William,

Cambridge, MA : The MIT Press, 2014.

Tres sitios web para explorar en relación con la asignatura:

TOP 500 supercomputers : <https://www.top500.org/>

GPU Computing : <https://devblogs.nvidia.com/learn-gpu-computing-with-hands-on-labs-gtc-2015/>

MPI usage : <https://mpitutorial.com/tutorials/>