



PRESENTACIÓN

La necesidad de utilizar cada vez más la potencia de los ordenadores para realizar tareas de cálculo, tratamiento de datos e inteligencia artificial, ha dado lugar de forma natural a lo que llamamos computación avanzada (“High performance computing” en inglés). El presente es un curso de programación en paralelo con un carácter esencialmente práctico y orientado a la resolución de problemas científicos, tanto de modelado como de procesamiento de datos.

Con vistas al cálculo 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.

Finalmente, la inteligencia artificial se ha convertido en una herramienta esencial para la ciencia. En la última parte de la asignatura estudiaremos los elementos principales que componen a una red neuronal y trabajaremos en programas sencillos para construir/entrenar redes neuronales simples en Python.

- **Titulación:** MÁSTER UNIVERSITARIO EN CIENCIA DE DATOS PARA CIENCIAS EXPERIMENTALES
- **Módulo/Materia:** Módulo III Optativo. Materia 3.1. Optativas
- **ECTS:** 3
- **Curso, semestre:** Curso 1o y semestre 2o
- **Carácter:** Optativa
- **Profesorado:** Dr. Reinaldo García García, Dr. Raúl Cruz Hidalgo
- **Idioma:** Español (bibliografía en inglés).
- **Aula, Horario:** Consultar calendario del máster
- **Es recomendable disponer de un ordenador portátil para instalar los programas necesarios.**
- **Se requiere conocimiento previo de programación en lenguaje Python.**

RESULTADOS DE APRENDIZAJE (Competencias)

RAO3 Desarrollar e implementar algoritmos de programación paralela y utilizar GPUs para el análisis de grandes conjuntos de datos.



PROGRAMA

Parte I: Cuda-GPU

1) Introducción a la programación sobre tarjetas gráficas (GPU), CUDA-NVIDIA en Python. 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.

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 cálculo paralelo OpenMP.

1) Introducción a la utilización de la librería OpenMP dentro de unos programas en Python.

Necesidad de la computación paralela en Ciencias y Ciencias de datos. El problema de la comunicación entre procesadores.

2) Programa básico usando OpenMP. 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.

Parte III: Introducción a las redes neuronales

1) Perceptrón. Características fundamentales. Función de activación.

2) Conectando neuronas: Estructura de una red neuronal.

3) Back propagation y aprendizaje no supervisado.

ACTIVIDADES FORMATIVAS

- AF1: Clases expositivas, 20 horas lectivas con presencialidad 100%
- AF2: Clases prácticas, seminarios y/o talleres, 10 horas lectivas con presencialidad 100%
- AF3: Trabajos dirigidos, individuales o en equipo, 20 horas no presenciales
- AF4: Estudio Personal, 20 horas no presenciales.



Universidad de Navarra

- AF6: Evaluación (Presentación de trabajo final), 3 horas presenciales
- AF7: Tutorías, 2 horas, presenciales.

EVALUACIÓN

- La nota final de la asignatura se calcula teniendo en cuenta un 20% en actividades continuas (trabajos de clase, asistencia y tareas) y un 80% en un trabajo final a desarrollarse bajo la supervisión del profesor.

HORARIOS DE ATENCIÓN

Dr. Reinaldo García García regarciag@unav.es

- Despacho O-171 Edificio. Castaños Planta 1
- Horario de tutoría: A solicitud de los estudiantes mediante concertación previa con el profesor.

BIBLIOGRAFÍA Y RECURSOS

- El profesor utiliza recursos en Castellano y en Inglés que tienen origen diverso y se comparten con los estudiantes a través de la carpeta de Material Docente.