

Guía docente

2703127 - HPC - Computación de Alto Rendimiento

Última modificación: 30/01/2026

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 701 - DAC - Departamento de Arquitectura de Computadores.
Titulación: GRADO EN BIOINFORMÁTICA (Plan 2024). (Asignatura obligatoria).
Curso: 2025 **Créditos ECTS:** 6.0 **Idiomas:** Inglés

PROFESORADO

Profesorado responsable: JOSE RAMON HERRERO ZARAGOZA

Otros: Segon quadrimestre:
CHRISTIAN GUZMAN RUIZ - 11, 12
JOSE RAMON HERRERO ZARAGOZA - 11, 12
SANDRA ADRIANA MENDEZ VALERIO - 11, 12
MIQUEL ANGEL SENAR ROSELL - 11, 12

CAPACIDADES PREVIAS

Para completar este curso satisfactoriamente, los participantes deben poseer una sólida base en programación imperativa secuencial y estructuras de datos fundamentales. Es imprescindible conocer el lenguaje de programación C, con especial énfasis en punteros y gestión dinámica de memoria. Se requiere familiaridad con la línea de comandos de Linux/Unix y las herramientas básicas de compilación para desenvolverse eficazmente en el entorno de laboratorio. Además, los estudiantes deben saber realizar la manipulación de expresiones matemáticas, ya que esta habilidad es necesaria para comprender los conceptos teóricos y los modelos de rendimiento presentados en las sesiones. Sin embargo, no se requieren conocimientos previos de programación paralela (MPI, OpenMP u OpenACC).

RESULTADOS DE APRENDIZAJE

Conocimientos:

K3. Identificar los fundamentos matemáticos, las teorías informáticas, los esquemas algorítmicos y los principios de organización de la información aplicables al modelado de sistemas biológicos y a la resolución eficiente de problemas bioinformáticos mediante el diseño de herramientas computacionales.

K4. Integrar los conceptos ofrecidos por los lenguajes de programación de mayor uso en el ámbito de las Ciencias de la Vida para modelar y optimizar estructuras de datos y construir algoritmos eficientes, relacionándolos entre sí y con sus casos de aplicación.

Habilidades:

S7. Implementar métodos de programación y análisis de datos orientados a partir de la elaboración de hipótesis de trabajo, dentro del área de estudio.

S8. Enfrentarse a la toma de decisiones, y defenderlas con argumentos, en la resolución de problemas de las áreas de biología, así como, dentro de los ámbitos adecuados, las ciencias de la salud, las ciencias de la computación y las ciencias experimentales.

Competencias:

C6. Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar este conocimiento.

METODOLOGÍAS DOCENTES

Cada semana el estudiante tendrá dos horas de sesión teórica y dos de sesión práctica. Las clases teóricas introducen todos los conocimientos, técnicas, conceptos necesarios para resolver problemas y trabajos de laboratorio. Algunos problemas pueden resolverse en las sesiones teóricas. Se espera del estudiante la realización de trabajo personal adicional resolviendo el resto de problemas de la colección de problemas. Adicionalmente, dos horas de sesiones laboratorio son también realizadas semanalmente; se valorará la participación activa y el rendimiento durante las sesiones de laboratorio (trabajar durante la sesión, avanzando el trabajo para alcanzar los objetivos de la sesión).

Los estudiantes deben trabajar autónomamente antes de las sesiones para prepararse para las clases y sacarle partido. Después, deben trabajar autónomamente para practicar y consolidar los conceptos y habilidades desarrolladas a lo largo del curso.

Para las prácticas y problemas de este curso se utilizará mayoritariamente el lenguaje de programación C y los modelos de programación paralela MPI, OpenMP y OpenACC. Las sesiones prácticas de laboratorio se realizan en una máquina paralela de tipo cluster con el sistema operativo Linux. El acceso al clúster deberá realizarse utilizando el ordenador portátil del estudiante. Por tanto, los estudiantes deben garantizar que pueden conectarse al clúster de Linux desde su ordenador portátil.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

- 1.Capacidad para formular modelos simples de rendimiento dada una estrategia de paralelización de una aplicación, que permita estimar la influencia de aspectos arquitectónicos principales: número de elementos de procesamiento, costo de acceso a datos y costo de interacción entre elementos de procesamiento, entre otros.
- 2.Capacidad para identificar los diferentes tipos de paralelismo que se pueden explotar en una arquitectura de computadores (ILP, TLP y DLP dentro de un procesador, multiprocesador y multicomputador) y describir sus principios de funcionamiento.
- 3.Capacidad para compilar y ejecutar un programa paralelo, utilizando las herramientas esenciales de línea de comandos para medir el tiempo de ejecución.
- 4.Capacidad para elegir la estrategia de descomposición más adecuada para expresar paralelismo en una aplicación (tareas, datos).
- 5.Capacidad para aplicar las técnicas básicas para sincronizar la ejecución paralela, evitando condiciones de carrera y bloqueos y posibilitando el solapamiento entre computación e interacción, entre otras.
- 6.Capacidad de programar en MPI la versión paralela de una aplicación secuencial.
- 7.Capacidad de programar en OpenMP la versión paralela de una aplicación secuencial.
- 8.Capacidad de programar en OpenMP la versión paralela de una aplicación secuencial.
- 9.Capacidad para medir, mediante herramientas de instrumentación, visualización y análisis, el rendimiento alcanzado con la implementación de una aplicación paralela y detectar factores que limitan dicho rendimiento: granularidad de tareas, carga equitativa e interacción entre tareas, entre otros.

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

Tipo	Horas	Porcentaje
Horas grupo grande	30,0	20.00
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00

Dedicación total: 150 h

CONTENIDOS

Introducción a la computación de alto rendimiento y la computación paralela

Descripción:

Introducción a la computación de alto rendimiento, las arquitecturas y la computación paralela

Entendiendo el paralelismo

Descripción:

Conceptos teóricos. Métricas de rendimiento.

Programación de memoria distribuida con MPI

Descripción:

Arquitecturas de memoria distribuida: programación con Message Passing Interface (MPI)

Modelado de rendimiento

Descripción:

Capacidad para formular modelos simples de rendimiento dada una estrategia de paralelización de una aplicación, que permita estimar la influencia de aspectos arquitectónicos principales: número de elementos de procesamiento, costo de acceso a datos y costo de interacción entre elementos de procesamiento, entre otros.

Programación de memoria compartida con OpenMP

Descripción:

Arquitecturas de memoria compartida: programación con OpenMP

Computación en clúster y sistemas de colas por lotes

Descripción:

Computación en clúster y sistemas de colas por lotes: SLURM

Python para computación de alto rendimiento

Descripción:

Python para computación de alto rendimiento

Computación con tarjetas gráficas (GPU) mediante OpenACC

Descripción:

Computación con tarjetas gráficas (GPU) mediante OpenACC

ACTIVIDADES

Teoría

Descripción:

Asumir la responsabilidad de su proceso de aprendizaje. Esto implica asistir puntualmente a clases, laboratorios y tutorías, y participar activamente en las discusiones. Los estudiantes son responsables de administrar su tiempo eficazmente para completar todas las tareas, lecturas y proyectos requeridos dentro de los plazos establecidos. También deben adherirse a las políticas de honestidad académica de la universidad, solicitar aclaraciones cuando el material no les quede claro y respetar el ambiente de aprendizaje interactuando con sus compañeros y el profesor de manera profesional.

Objetivos específicos:

1, 2, 4, 5, 6, 7, 8

Dedicación: 72h

Grupo grande/Teoría: 27h

Aprendizaje autónomo: 45h

Sesiones prácticas

Descripción:

El estudiante debe desarrollar tareas prácticas utilizando un clúster paralelo. En dicho entorno, sus responsabilidades incluyen la ejecución activa, la resolución de problemas y la gestión de recursos. Debe:

- Prepararse con antelación revisando los conceptos teóricos y las lecturas previas al laboratorio o las instrucciones de configuración inicial antes de la sesión de laboratorio.
- Ejecutar las tareas de forma independiente, documentando cuidadosamente los procedimientos, las entradas y las salidas, incluyendo cualquier error encontrado y su resolución.
- Desarrollar habilidades prácticas de depuración y utilizar los recursos técnicos disponibles (documentación, herramientas de software específicas) para resolver los desafíos técnicos antes de buscar la ayuda del instructor.
- Presentar código/informes funcionales y bien documentados que cumplan con el formato, el control de versiones y las convenciones de nomenclatura especificados.

Objetivos específicos:

3, 4, 5, 6, 7, 8, 9

Dedicación: 73h

Grupo mediano/Prácticas: 28h

Aprendizaje autónomo: 45h

Examen 1ª parte

Descripción:

Examen 1ª parte

Objetivos específicos:

1, 2, 4, 5, 6, 9

Dedicación: 1h 30m

Actividades dirigidas: 1h 30m



Examen Laboratorio 1ª parte (LE1)

Descripción:

Examen Laboratorio 1ª parte (LE1)

Objetivos específicos:

3, 4, 5, 6, 9

Dedicación: 1h

Actividades dirigidas: 1h

Examen 2ª parte

Descripción:

Examen 2ª parte

Objetivos específicos:

2, 4, 5, 7, 8

Dedicación: 1h 30m

Actividades dirigidas: 1h 30m

Examen Laboratorio 2ª parte (LE2)

Descripción:

Examen Laboratorio 2ª parte (LE2)

Objetivos específicos:

3, 4, 7, 8, 9

Dedicación: 1h

Actividades dirigidas: 1h

SISTEMA DE CALIFICACIÓN

La nota del curso se calculará como la media ponderada de 6 notas (3 notas para cada una de las partes del curso):

$$N = 0.05 * SL1 + 0.15 * EL1 + 0.3 * ET1 + 0.05 * SL2 + 0.15 * EL2 + 0.3 * ET2$$

Donde:

- SL_i está relacionado con entregas de prácticas de la parte - i-ésima
- EL_i examen de laboratorio de la parte i-ésima
- ET_i examen de teoría de la parte i-ésima

IMPORTANTE: La realización y presentación de todos los informes de seguimiento del laboratorio (SL_i) será condición necesaria para superar la asignatura. Sólo se considera informe elaborado y presentado un informe con un mínimo de contenido. Informes vacíos o con sólo las preguntas, por ejemplo, no se consideran realizados ni presentados.

Los estudiantes aprobarán el curso si han completado todos los exámenes y han obtenido una nota del curso (N) igual o superior a 5 sobre 10.

Los estudiantes que suspendan el curso con una nota media final superior a 3 sobre 10, podrán realizar un examen de recuperación de la parte teórica. El examen de recuperación constará de dos partes: habrá un examen de recuperación para cada una de las 2 partes (ER1 y/o ER2) para los estudiantes que hayan suspendido esta parte específica. En el examen de reevaluación sólo se podrá recuperar la parte teórica del curso. La nota ER_i sustituirá a la nota obtenida en el curso regular para el examen de teoría correspondiente, ET_i.

BIBLIOGRAFÍA

Básica:

- Grama, Ananth. Introduction to parallel computing. 2nd ed. Pearson Addison Wesley, 2003. ISBN 9780201648652.
- Hager, Georg; Wellein, Gerhard. Introduction to high performance computing for scientists and engineers. CRC Press, cop. 2011. ISBN 9781439811924.
- Gropp, William; Lusk, Ewing; Skjellum, Anthony. Using MPI : portable parallel programming with the Message-Passing-Interface. Third edition. MIT Press, [2014]. ISBN 9780262326605.
- Chapman, Barbara; Jost, Gabriele; Pas, Ruud van der. Using OpenMP : portable shared memory parallel programming. MIT Press, cop. 2008. ISBN 9780262533027.

Complementaria:

- McCool, Michael; Robison, Arch D; Reinders, James. Structured parallel programming : patterns for efficient computation. 1st edition. Elsevier/Morgan Kaufmann, [2012]. ISBN 9786613689603.
- Gropp, William; Lusk, Ewing; Skjellum, Anthony. Using MPI : portable parallel programming with the Message-Passing-Interface. Third edition. MIT Press, [2014]. ISBN 9780262326605.
- Eijkhout, Victor. The Science of Computing (formerly: Introduction to High-Performance Scientific Computing). CC-BY 4.0 license. 2022.
- Eijkhout, Victor. Parallel Programming in MPI and OpenMP The Art of HPC, volume 2. CC-BY 4.0 license. 2022.

RECURSOS

Enlace web:

- <https://computing.llnl.gov/tutorials/mpi/>- <https://hpc-tutorials.llnl.gov/openmp/>- <https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial-> <https://slurm.schedmd.com/>- <https://www.mpi-forum.org/docs/>- <https://www.openacc.org/>- <https://www.openmp.org/resources/>