



IIC3533 — Computación de Alto Rendimiento — 2025-2

Programa de Curso

Profesor: Ignacio Labarca, ijlabarca@uc.cl
Clases: Martes y Jueves, módulo 4. Sala B12.
Ayudantes: José Carter, jrcarter@uc.cl
Manuel Cifuentes, mecifuentes@uc.cl
Nicolás Olmos, nicolas.olmos@uc.cl,
Sebastián Salgado, ssalgadn@uc.cl

1 Descripción

Los sistemas modernos de cómputo poseen una capacidad de procesamiento impensada hasta hace algunos años. Estas nuevas capacidades ofrecen posibilidad para diversas áreas del cómputo científico, simulación, o búsqueda en espacios amplios de soluciones.

Para poder aprovechar toda la potencialidad de estos sistemas, es necesario comprender su funcionamiento junto a las bases que sustentan la ejecución de código paralelo o distribuido, y sus limitaciones.

2 Objetivo General

En este curso los estudiantes utilizarán técnicas y conceptos básicos para desarrollar, analizar e implementar algoritmos paralelos escalables y como sus programas pueden hacer uso de las tecnologías modernas de cómputo de alto rendimiento. Se analizarán problemas actuales y las tecnologías que permiten soportar este tipo de cómputo.

3 Competencias

Al finalizar el curso el alumno será capaz de:

1. **Describir** las arquitecturas más comúnmente utilizadas en la construcción de plataformas de alto rendimiento y sus mecanismos asociados.
2. **Analizar** los modelos utilizados y el rendimiento de algoritmos que se ejecutan en plataformas de alto rendimiento.
3. **Implementar** programas en ambientes paralelos utilizando memoria compartida, paso de mensajes, y arquitecturas manycore.
4. **Aplicar** técnicas de cómputo de alto rendimiento en áreas como ciencia de datos, aprendizaje de máquina, cómputo científico, u otras áreas de la ingeniería.

4 Contenido

1. Cómputo monoprocesador
 - 1.1 Arquitecturas y procesadores modernos.
 - 1.2 Arquitecturas multicore.
 - 1.3 Localidad y acceso a memoria.
 - 1.4 Conceptos de alto rendimiento.
2. Cómputo paralelo
 - 2.1 Procesos y threads.
 - 2.2 Arquitecturas para cómputo paralelo.
 - 2.3 Granularidad y programación paralela.
 - 2.4 Arquitecturas multi-threaded y coprocesadores (GPU).
3. Uso de plataformas de alto rendimiento
 - 3.1 Arquitecturas de cluster.
 - 3.2 Schedulers y colas de trabajo.
 - 3.3 Sistemas de archivos.
4. Programación con paso de mensajes: MPI
 - 4.1 Memoria distribuida y paso de mensajes.
 - 4.2 Paralelismo funcional: SPMD.
 - 4.3 Operaciones colectivas: gather, scatter.
 - 4.4 Modelos de comunicación.
5. Programación con memoria compartida: OpenMP
 - 5.1 Modelo de memoria compartida.
 - 5.2 Regiones paralelas y ciclos.
 - 5.3 Reducciones y compartición de trabajo.
 - 5.4 Sincronización de threads.
6. Programación en GPUs
 - 5.1 Arquitectura de GPUs
 - 5.2 CUDA y OpenCL.
7. Aplicaciones
 - 6.1 Dinámica molecular.
 - 6.2 Algoritmos combinatoriales.
 - 6.3 Algoritmos sobre grafos.
 - 6.4 N-Body.
 - 6.5 Métodos de Monte Carlo.
 - 6.6 Aplicaciones de machine learning.

5 Metodología

Se realizarán:

- **Clases de cátedra.**
- **Sesiones prácticas** donde se trabajará en un problema concreto, idealmente con computador. Se permitirá trabajo en grupo.
- **Sesiones de presentación** de artículos científicos o herramientas. Se publicará una lista de temáticas donde grupos de estudiantes deberán presentar un artículo o herramienta y realizar un análisis en base a una pauta. Solo estas sesiones serán con asistencia.

6 Evaluación

Se evaluará mediante:

- 30% Controles de 60 minutos (4 controles, se elimina el peor).
- 30% Tareas grupales (2).
- 30% Presentación Final grupal (último mes de clases).
- 10% Asistencia a presentaciones.

- **Único requisito de aprobación será**

$$N_F \geq 4.0.$$

Las fechas de los controles son:

Control 1	9 de Septiembre♠
Control 2	25 de Septiembre
Control 3	25 de Octubre♠
Control 4	12 de Diciembre♠

♠ Fecha y horario de interrogaciones y examen asignados al curso.

7 Bibliografía

El curso está preparado de manera que el material de clases y las referencias que se hagan durante ellas sean suficiente para comprender los contenidos. Sin embargo, aquellos que desean profundizar más en los contenidos están invitados a hacerlo. La principal fuente que se ha utilizado para preparar el material es:

1. Victor Eijkhout, The Art of HPC series. <https://theartofhpc.com/>

8 Compromiso del Código de Honor

Este curso adscribe el Código de Honor establecido por la Universidad, el que es vinculante. Todo trabajo evaluado en este curso debe ser propio. En caso de que exista colaboración permitida con otros alumnos, el trabajo deberá referenciar y atribuir correctamente dicha contribución a quien corresponda. Como alumno es su deber conocer el Código de Honor (www.uc.cl/codigodehonor).

Política de Integridad Académica del Departamento de Ciencia de la Computación

Los/as estudiantes de la Escuela de Ingeniería de la Pontificia Universidad Católica de Chile deben mantener un comportamiento acorde a la Declaración de Principios de la Universidad. En particular, se espera que mantengan altos estándares de honestidad académica. Cualquier acto deshonesto o fraude académico está prohibido; los/as estudiantes que incurran en este tipo de acciones se exponen a un Procedimiento Sumario. Es responsabilidad de cada estudiante conocer y respetar el documento sobre Integridad Académica publicado por la Dirección de Pregrado de la Escuela de Ingeniería.

Específicamente, para los cursos del Departamento de Ciencia de la Computación, rige obligatoriamente la siguiente política de integridad académica. Todo trabajo presentado por un/a estudiante para los efectos de la evaluación de un curso debe ser hecho individualmente por el/la estudiante, sin apoyo en material de terceros. Por “trabajo” se entiende en general las interrogaciones escritas, las tareas de programación u otras, los trabajos de laboratorio, los proyectos, el examen, entre otros.

En particular, si un/a estudiante copia un trabajo, o si a un/a estudiante se le prueba que compró o intentó comprar un trabajo, obtendrá nota final 1.1 en el curso y se solicitará a la Dirección de Pregrado de la Escuela de Ingeniería que no le permita retirar el curso de la carga académica semestral.

Por “copia” se entiende incluir en el trabajo presentado como propio, partes hechas por otra persona. En caso que corresponda a “copia” a otros estudiantes, la sanción anterior se aplicará a todos los involucrados. En todos los casos, se informará a la Dirección de Pregrado de la Escuela de Ingeniería para que tome sanciones adicionales si lo estima conveniente.

También se entiende por copia extraer contenido sin modificarlo sustancialmente desde fuentes digitales como Wikipedia o mediante el uso de asistentes inteligentes como ChatGPT o Copilot. Se entiende que una modificación sustancial involucra el análisis crítico de la información extraída y en consecuencia todas las modificaciones y mejoras que de este análisis se desprendan. Cualquiera sea el caso, el uso de fuentes bibliográficas, digitales o asistentes debe declararse de forma explícita, y debe indicarse cómo el/la estudiante mejoró la información extraída para cumplir con los objetivos de la actividad evaluativa.

Obviamente, está permitido usar material disponible públicamente, por ejemplo, libros o contenidos tomados de Internet, siempre y cuando se incluya la referencia correspondiente. Lo anterior se entiende como complemento al Reglamento del Estudiante de la Pontificia Universidad Católica de Chile (<https://registrosacademicos.uc.cl/reglamentos/estudiantiles/>). Por ello, es posible pedir a la Universidad la aplicación de sanciones adicionales especificadas en dicho reglamento.