

# Computación paralela y distribuida

LEAD University

Bachillerato de Ingeniería en Ciencias de Datos

|                                   |  |
|-----------------------------------|--|
| Universidad                       | Universidad Liderazgo, excelencia, avance y desarrollo         |
| Nombre del curso                  | Computación paralela y distribuida                             |
| Código del curso                  | BCD-9218   |
| Carrera                           | Bachillerato en Ingeniería en Ciencia de Datos                 |
| Sede                              | San José   |
| Nombre del profesor               | Christian Asch   |
| Naturaleza del curso              | Teórico-Práctico   |
| Nivel                             | Bachillerato   |
| Ubicación                         | IX Cuatrimestre  |
| Modalidad                         | Híbrida  |
| Duración                          | Un cuatrimestre  |
| No. Sesiones/semana               | 1  |
| Horas teóricas                    | 2  |
| Horas prácticas                   | 1  |
| Horas estudio independiente       | 8  |
| Requisitos                        | Sistemas operativos BCD-6209<br>Redes de computadoras BCD-7212 |
| Créditos                          | 4  |
| Horario propuesto                 | J: 19:00 - 22:00   |
| Asistencia                        | Obligatoria  |
| Laboratorios-talleres             | N/A  |
| Lugar donde se impartirá el curso | Sede LEAD University/Clases Virtuales                          |

## Descripción del curso

La Ley de Moore ha guiado la industria de los semiconductores desde hace ya varias décadas. Sin embargo, la tecnología de fabricación de transistores se está acercando a los límites físicos, generando limitaciones térmicas en el diseño de los procesadores. Por este motivo, las arquitecturas computacionales modernas han optado por incorporar múltiples procesadores a un único circuito integrado, en lugar de hacer monoprocesadores más complejos. Esta tendencia ha llevado a que los multiprocesadores sean encontrados en sistemas como los dispositivos móviles y hasta las supercomputadoras. Es necesario entonces que quienes se encuentran en áreas relacionadas a la computación, sean capaces de dominar la programación de sistemas paralelos para crear las aplicaciones y los sistemas que requiere una sociedad y una industria de alta tecnología.

En este curso se estudia el tema del paralelismo en la computación y se muestra cómo, tanto el hardware como el software, deben ser considerados para lograr la eficiencia en la computación en paralelo. Se presentarán los conceptos fundamentales de la computación paralela y se analizarán

las principales herramientas y algoritmos para el diseño, implementación y validación de aplicaciones paralelas.

Entre los valores y actitudes que se fomentarán se encuentran:

- Pensamiento crítico y sistémico.
- Creatividad.
- Orden.
- Comunicación eficaz.

## Objetivo general

Explorar las técnicas de computación en paralelo en ambientes de memoria compartida así en como ambientes distribuidos, mediante el análisis del hardware, el sistema operativo, las herramientas de software y los algoritmos involucrados en el desarrollo de aplicaciones paralelas.

## Objetivos específicos

- Analizar las principales arquitecturas paralelas existentes y sus implicaciones en el desarrollo de aplicaciones.
- Comparar y aplicar las principales herramientas para lograr la computación paralela tanto en sistemas de memoria compartida como aquellos de memoria distribuida.
- Comprender los modelos analíticos y las métricas de rendimiento de programas paralelos.

## Requisitos

- El curso supone que el estudiante matriculado cumple con los siguientes requisitos:
- Familiaridad con algún lenguaje de programación como Python o C.
- Familiaridad con ambientes UNIX/Linux y el uso de la línea de comandos.
- Conocimiento de los fundamentos de análisis de algoritmos y estructuras de datos.

## Contenidos

### Tema 1. Introducción a la programación paralela

- Conceptos de programación paralela
- Repaso del lenguaje de programación C

### Tema 2. Programación paralela en memoria compartida

- Conceptos de memoria compartida
- Programación con OpenMP en C
- Consideraciones de rendimiento
- Programación paralela basada en tareas

### Tema 3. Programación paralela en memoria distribuida

- Programación con MPI en C y Python
- Programas MPI+OpenMP
- Topologías virtuales

## Tema 4. Programación con aceleradores gráficos (GPU)

- Conceptos de aceleradores gráficos.
- Programación de aceleradores gráficos con OpenMP en C.
- Programación de aceleradores gráficos con numba en Python.

## Tema 5. Herramientas de ciencias de datos distribuidas

## Metodología de enseñanza

La metodología que se utilizará es el aprendizaje colaborativo para la resolución de los problemas y tiene como objetivo hacer transitar al estudiante por caminos similares a los que transitó el especialista para llegar a sus conclusiones.

El curso se desarrolla mediante lección expositiva, fomentando siempre la participación del estudiante y la resolución de problemas. En todos los temas se llevan a cabo demostraciones y aplicaciones de los teoremas y algoritmos, además los proyectos y problemas son resueltos fomentando la participación en equipo de los estudiantes.

La metodología de este curso es coherente con los principios que sustentan el plan de estudios. Los estudiantes requieren de una formación analítica a través de los cursos en los que debe propiciarse su desarrollo. La metodología predominante será la lección expositiva combinada con ejercicios prácticos, buscando alternativas en las que los estudiantes sean expuestos conscientemente al uso de su capacidad lógica.

En los ejercicios prácticos deberán aplicar el fruto de sus investigaciones soportada sobre una base predeterminada y las experiencias observadas en el registro de operaciones comerciales.

La capacidad analítica y deductiva constituye un elemento básico para la interpretación de resultados, lo que permite ampliar aún más el esquema al ser utilizado como instrumento de proyección.

El curso está diseñado para que el estudiante pueda aplicar los conceptos aprendidos en clase, éstos ponerlos en práctica; para ello, el curso se complementa con el desarrollo de un proyecto.

## Estrategias de aprendizaje

Quienes cursen este programa deberán atender en las clases de manera participativa y realizar antes de la clase los ejercicios y lecturas previamente asignadas. La realización de grupos de estudio entre los estudiantes, donde compartan prácticas externas a las brindadas por el docente, les permitirá reforzar el conocimiento adquirido y la interpretación crítica.

Las estrategias de aprendizaje recomendadas para este curso se agrupan en las siguientes categorías:

- Estrategias para el rescate de los conocimientos previos de los estudiantes: revisión de materia y generación de lluvia de ideas.
- Estrategias para provocar el pensamiento crítico: discusiones grupales y estudio de casos prácticos.

- Estrategias para provocar la construcción del conocimiento: Estudiar de forma crítica de las diferentes teorías, brindados como ejemplo por parte del docente o bien ubicados en internet.

## Recursos didácticos

Las aulas están equipadas con diversas herramientas tecnológicas que fomentan el aprendizaje, de acuerdo con la estrategia de enseñanza a utilizar. Los salones de clase están equipados con pantallas plegables, proyectores, computadora, pizarras acrílicas y electrónicas. Además, se cuenta con un aula virtual, acceso a internet, correo electrónico, paquete de Microsoft Office 365 y una plataforma tecnológica para la presentación de videoconferencias. Software: MPI, OpenMP

Asimismo, se cuenta con laboratorios de computación (software y hardware) necesarios para el desarrollo de las asignaciones y proyectos propuestos por el profesor.

## Evaluación

La evaluación será literal, sobre la base de lo establecido en el artículo 15 del Reglamento Académico de la Universidad:

| RUBROS   | PUNTOS EQUIVALENTES |
|--|---------------------|
| Tareas (4) (15 puntos cada una) Las tareas buscan reforzar el conocimiento adquirido por el estudiante en clase y fomentar además la investigación para la resolución de problemas complejos. Estas tareas serán variadas y pueden ser desde resolución de ejercicios basados en teoría hasta el desarrollo de aplicaciones paralelas. | 60                  |
| Desarrollo de Proyecto (1 proyecto)<br>Primera entrega (10)<br>Segunda entrega (10)<br>Entrega final (10)<br>Presentación (10)   | 40                  |
| <b>TOTAL DE PUNTOS</b>   | <b>100</b>          |

## Rúbrica para evaluar la resolución de tareas y proyecto: 100%

Para evaluar la resolución de problemas, se emplea la siguiente rúbrica.

|                                |  |   |  |  |  |                   |
|--------------------------------|--|---|--|--|--|-------------------|
| Docente                        |  |   |  |  |  |                   |
| Alumno                         |  |   |  |  |  |                   |
| Grupo                          |  |   |  |  |  |                   |
|                                |  |   |  |  |  |                   |
| Criterios                      | <b>A: Excelente<br/>(90-100%)</b>  | <b>B: Bueno<br/>(83%-90%)</b>   | <b>C: Satisfactorio<br/>(75%-83%)</b>  | <b>D: Poco<br/>Satisfactorio<br/>(68%-75%)</b>                                     | <b>F: Reprobado<br/>(67% o<br/>menos)</b>  | <b>Puntuación</b> |
| <b>Identificar</b>             | Identifica y presenta ordenadamente los datos e incógnitas de un problema            | Identifica y presenta sin orden los datos e incógnitas de un problema                     | Identifica y presenta parcialmente los datos e incógnitas de un problema                 | Le cuesta identificar y presentar los datos e incógnitas de un problema            | No identifica, ni presenta los datos e incógnitas de un problema                       |                   |
| <b>Plantear</b>                | Al planear relaciona los datos con las incógnitas de manera sintetizada              | Al planear relaciona los datos con las incógnitas   | Al planear no relaciona los datos con las incógnitas                                     | Le cuesta planear relaciones entre datos con las incógnitas                        | No planea relaciones entre datos con las incógnitas                                    |                   |
| <b>Resolver</b>                | Resuelve las operaciones siguiendo un proceso ordenado y da la respuesta correcta    | Resuelve las operaciones con algún desorden u omisión de algunos pasos.                   | Resuelve parcialmente las operaciones con algún desorden u omisión de algunos pasos.     | Resuelve con algún desorden u omisión de algunos pasos.                            | No resuelve las operaciones  |                   |
| <b>Evaluar</b>                 | La exposición del caso demuestra un excelente entendimiento al problema en estudio   | La exposición del caso demuestra un mediano entendimiento al problema en estudio          | El informe del caso demuestra algún entendimiento del tema tratado                       | El informe del caso demuestra un entendimiento muy limitado del problema           | No presenta el informe del caso y demuestra un entendimiento muy limitado del problema |                   |
| <b>Actitud ante la crítica</b> | Siempre estuvo receptivo a aceptar críticas y sugerencias de los miembros del equipo | Casi siempre estuvo receptivo a aceptar críticas y sugerencias de los miembros del equipo | Pocas veces estuvo receptivo a aceptar críticas y sugerencias de los miembros del equipo | Nunca estuvo receptivo a aceptar críticas y sugerencias de los miembros del equipo | Nunca participó con el equipo  |                   |

## Bibliografía

Pacheco, P. (2011). An introduction to parallel programming. Elsevier.

Grama, A., Kumar, V., Gupta, A., & Karypis, G. (2003). Introduction to parallel computing. Pearson Education.

Eijkhout, V. (2013). Introduction to High Performance Scientific Computing. Lulu.

Patterson, D. A, and Hennessy, J., (2005). Computer organization and design: the hardware/software interfaz. Kaufmann Publishers.

Hennessy, J. and Patterson, D. A., (2011). Computer Architecture, Fifth Edition: A Quantitative Approach (5th. ed.). Morgan Kaufmann Publishers Inc.

Trobec, R., Slivnik, B., Bulić, P., & Robič, B. (2018). Introduction to Parallel Computing: From Algorithms to Programming on State-of-the-Art Platforms. Springer.

## Cronograma

| Semana | Tema   | Asignación   |
|--------|--|--|
| 1      | Introducción al curso y a la programación paralela y distribuida   | Publicación de tarea 1<br>Publicación del proyecto |
| 2      | Repaso de programación en C <ul style="list-style-type: none"><li>• Compilación.</li><li>• Conceptos básicos.</li><li>• Estructuras.</li><li>• Manejo manual de memoria.</li></ul> Uso del clúster Kabré |  |
| 3      | Conceptos de programación paralela con memoria compartida.<br>Introducción al uso de OpenMP. <ul style="list-style-type: none"><li>• Single Program Multiple Data</li><li>• Sincronización</li></ul>     |  |
| 4      | Medición de rendimiento <ul style="list-style-type: none"><li>• Ley de Amdahl</li><li>• Ley de Gustafson</li></ul> Ciclos paralelos <ul style="list-style-type: none"><li>• Reducciones</li></ul>        | Entrega tarea 1<br>Publicación de tarea 2          |

|    |  |   |
|----|--|---|
|    | <ul style="list-style-type: none"> <li>• nowait</li> </ul> Ambiente de datos   |   |
| 5  | Paralelismo basado en tareas con OpenMP.                                       | Entrega del primer avance del proyecto <ul style="list-style-type: none"> <li>- Grupos</li> <li>- Posibles temas</li> </ul>                                 |
| 6  | Conceptos de memoria distribuida.<br>Introducción al uso de MPI en C y Python. | Entrega tarea 2<br>Publicación tarea 3  |
| 7  | Comunicación punto a punto con MPI.<br>Comunicación asíncrona.                 | Entrega del segundo avance del proyecto <ul style="list-style-type: none"> <li>- Cronograma</li> <li>- Estado actual</li> <li>- Cosas que faltan</li> </ul> |
| 8  | Comunicación colectiva con MPI.<br>Topologías virtuales y comunicadores.       |   |
| 9  | Programación con MPI y OpenMP en conjunto                                      | Entrega tarea 3<br>Publicación tarea 4  |
| 10 | Conceptos de programación de GPUs con OpenMP                                   |   |
| 11 | Programación de GPUs con OpenMP  |   |
| 12 | Programación de GPUs con Python  | Entrega tarea 4   |
| 13 | Programación de GPUs con Python  | Entrega del proyecto  |
| 14 | Presentaciones del proyecto  |   |
| 15 | Presentaciones del proyecto  |   |