

Planificación de la Asignatura: Computación de Alto Rendimiento

Fecha: 23/10/2024 13:02

Código: L1325

Carrera: Licenciatura en Bioinformática

Departamento Académico: Informática

Docente a cargo:

Correo del docente a cargo: luciano.garelli@uner.edu.ar

Régimen de Dictado: Cuatrimestral 2º Cuatrimestre

Carga Horaria Semanal: 4 horas semanales

Carga Horaria Total: 56 horas

Contenidos Mínimos:

Arquitectura de computadoras paralelas. Criterios de clasificación. Modelos de programación paralela. Estándares y librerías existentes (MPI, MPICH y OpenMP). Conceptos básicos de MPI. Compilación de programas y ejecución en paralelo. Comunicación punto a punto. Comunicaciones colectivas. MPI en ambientes Unix. Tiempos de comunicación y sincronización. Escalabilidad. Conceptos básicos de OpenMP. Interacción con OpenMP. Balance de carga. Regiones paralelas. Directivas.

Competencias Genéricas:

CT 1 - 1. Identificación, formulación y resolución de problemas de la disciplina Bioinformática. Nivel de dominio 1

CT 2 - 2. Concepción, diseño y desarrollo de proyectos de la disciplina Bioinformática. Nivel de dominio 2

CT 4 - 4. Utilización de técnicas y herramientas de aplicación en la disciplina Bioinformática. Nivel de dominio 3

Competencias Específicas:

CE 3 - Desarrollar estudios en metodologías estadísticas, matemáticas y computacionales para analizar el genoma y la expresión génica. Nivel de dominio 2

Argumentación de aportes marcados en la matriz de competencias:

La cátedra proporcionará a los estudiantes las bases teóricas y prácticas para identificar, formular y resolver problemas bioinformáticos que demande gran capacidad de cómputo.

Se abordarán casos de estudio reales para que los estudiantes puedan aplicar sus conocimientos a situaciones concretas.

Se fomentará el pensamiento crítico y la capacidad de análisis para la toma de decisiones.

La cátedra capacitará a los estudiantes en el uso de las principales técnicas y herramientas en el área de computación de alto rendimiento aplicadas en bioinformática.

Se brindará formación práctica en el uso de diferentes softwares (compiladores y librerías).

Se fomentará la capacidad de aprendizaje continuo para adaptarse a las nuevas tecnologías.

Correlativas Regulares para cursar:

Algoritmos y Estructuras de Datos

Redes de Computadoras

Correlativas Aprobadas para cursar:

Arquitecturas y Sistemas Operativos

Correlativas Aprobadas para promocionar o rendir el examen final:

Arquitecturas y Sistemas Operativos

Algoritmos y Estructuras de Datos

Insercion de la Asignatura en el plan de Estudios:

La materia “Computación de Alto Rendimiento” se encuentra en el tercer año del plan de estudios de la Licenciatura en Bioinformática, la cual vinculando los conocimientos y aptitudes adquiridas en materias como Fundamentos de Programación, Programación Avanzada, Algoritmos y Estructura de Datos y Redes de Computación, que conjuntamente con los temas desarrollados, le brindarán al alumno la posibilidad de abordar la resolución de problemas que demanden un alto costo desde el punto de vista computacional. Este tipo de problemas es cada día más frecuente, tanto en el ámbito de la industria como de la investigación, los modelos son cada vez más complejos y detallados incrementando por lo tanto la cantidad de datos a analizar y pos-procesar. Además, según lo indicado en la resolución de creación de la carrera de Bioinformática, se enuncia que el objetivo de esta carrera es:

“Explorar, analizar, desentrañar y ordenar datos biológicos obtenidos de distintas fuentes como por ejemplo secuenciación de genomas, proteomas, transcriptomas, modelización de sistemas biológicos, etc.”

Estas fuentes crecen en complejidad día a día, haciendo de la Computación de Alto Rendimiento un recurso imprescindible para llevar a cabo estas tareas en tiempos razonables. Por lo tanto, es necesario dotar al alumno de herramientas básicas para poder afrontar este tipo de desafío en su desarrollo como profesional. También le brindará la posibilidad de articular con el seminario de Avances en Biología Computacional pudiendo implementar algoritmos referentes a la Dinámica Molecular, Integración por Monte Carlo, Algoritmos Genéticos, Diagonalización de Matrices, entre otros.

Objetivo General:

El objetivo de esta materia es desarrollar las aptitudes del alumno en cuanto a la programación en C++ con los estándares/librerías MPI y OpenMP, los cuales son actualmente una combinación efectiva para la resolución de grandes problemas en Bioinformática empleando un clúster de computadoras o entornos de memoria compartida con procesadores multicore. La orientación de la materia es hacia el aprendizaje del uso intensivo de estas tecnologías.

Con las herramientas brindadas en esta materia el futuro profesional podrá extender sus conocimientos sobre Computación de Alto Rendimiento y Cálculo Paralelos a sistemas más específicos, como así también podrá utilizar otros tipos de librerías para la paralelización según sea su necesidad puntual.

También es importante instruir al estudiante en el uso de la computadora como herramienta de trabajo y que desarrolle un pensamiento lógico.

Objetivos Particulares:

- Conceptos y terminología dentro de Cálculo Paralelo.
- Modelos de programación en paralelo.
- Diseño de programas para su ejecución en paralelo.
- Reconocer las particularidades de las distintas arquitecturas de computadoras paralelas.
- Criterios de clasificación de las computadoras paralelas.
- Poder utilizar diferentes librerías para llevar a cabo la ejecución de programas en paralelo.
- Medir los tiempos de comunicación y estimar escalabilidad en un clúster o en una computadora multi-core.
- Determinar la granularidad de un determinado sistema.
- Conocer los tipos de comunicaciones existentes entre nodos de un clúster.
- Balance de carga para una óptima desempeño del clúster, ya sea del tipo homogéneo o heterogéneo.

- Implementar algoritmos básicos que permitan apreciar las ventajas del cálculo en paralelo.
- Utilizar combinaciones de librerías (MPICH – OpenMP) para explotar la potencialidad de los clúster multi-core.

Actividades de evaluación:

Estrategia de evaluación:

- Dos exámenes parciales.
- Evaluación de la participación pertinente en clase.
- Aprobación de los Trabajos Prácticos propuestos.
- Examen final

Programa Analítico:

Programa:

Tema 1: Introducción a la Computación de Alto Rendimiento.

- ¿Qué se conoce por Computación de Alto Rendimiento?.
- Ejemplos de utilización.
- Tendencias en el Cómputo de Alto Rendimiento (Placas gráficas GPGPU's, Co-procesadore Xeon Phi).

Tema 2: Conceptos y terminología.

- Arquitectura de Von Neumann.
- Taxonomía de las computadoras para cálculo paralelo.
- Elementos que componen un clúster.

Tema 3: Programación en paralelo.

- Modelos de programación en paralelo.
- Diseño de programas.
- Ejemplo de algoritmos.

Tema 4: Cómputo paralelo utilizando el estándar MPI (Message Passing Interface).

- Conceptos básicos de MPI. Memoria distribuida.
- Uso de MPI en programas simples.
- Estrategia master/slave en SPMD.
- Formato de llamadas.
- Funciones básicas.

Tema 5: Comunicaciones punto a punto y colectivas.

- Envoltura de un mensaje.
- Condiciones de recepción.
- Comunicaciones bloqueantes y no-bloqueantes.
- Llamadas colectivas. Broadcast.
- Tiempos de comunicación y sincronización. Escalabilidad.
- Ejemplo de utilización.

Tema 6: Comunicaciones colectivas avanzadas.

- Formato de llamadas.
- Funciones vectorizadas.
- Ejemplos de utilización

Tema 7: OpenMP en clústers de memoria compartida.

- Conceptos básicos de OpenMP.
- Datos privados y públicos.
- Condición de carrera (Race conditions).
- Regiones paralelas. Directivas. Funciones.
- Lazos paralelos. Sincronización.

Tema 8: Utilización de OpenMP.

- Balance de carga.
- Sincronización con barreras.
- Regiones críticas. Directivas single y master.
- Ejemplo: Productos de muchas matrices.
- Ejemplo: Producto matriz vector.

Metodología Didáctica:**SEGUNDO CUATRIMESTRE 2024**

Las instancias para el desarrollo de la asignatura serán clases teóricas, prácticas y de coloquio.

Para el desarrollo de cada una de las partes en que se divide la asignatura, se partirá de elementos simples sobre los que se estructurará el conocimiento.

El dictado de la cátedra será de carácter teórico y práctico, realizándose una conceptualización y aplicación inmediata por medio de estudio dirigido y trabajos prácticos sobre la computadora. Se estimulará el trabajo grupal y el pensamiento lógico, enfatizando la tarea crítica.

El docente actuará como orientador y estimulador en la resolución de problemas, tanto de carácter algorítmico como de programación.

La materia cuenta con una exigencia total de 56 horas que en las 14 semanas del cuatrimestre implican una carga horaria semanal de 4 horas, las que se distribuirán de la siguiente forma:

- Clase teórica.

Una clase de 2 a 3 horas semanales. Que estará a cargo del docente titular y/o adjunto.

- Coloquio.

Tendrá como objetivo servir de marco para la discusión tanto de los conceptos teóricos como de los resultados obtenidos en la práctica. También se discutirán las dudas que se presentasen en la lectura de la bibliografía y sobre todo a la discusión de la fundamentación teórica de los resultados prácticos.

- Trabajos prácticos y Estudio dirigido

Debido al carácter eminentemente práctico de la asignatura, se trabajará con una carga horaria aproximada de 2 Hs. Semanales siendo combinada con el coloquio. En éstas se realizará una evaluación continua del progreso de los alumnos, estimulando además la autoevaluación por parte del estudiante, como medio idóneo para aumentar su rendimiento, por ello será obligatoria la aprobación de los prácticos.

Formación Práctica:

Debido al carácter eminentemente práctico de la asignatura, la resolución de ejercicios y problemas básicos es fundamental para el aprendizaje de las técnicas de programación en paralelo. En una primera instancia se abordará la resolución de ejercicios, los cuales tienen como objetivo la utilización de las funciones y técnicas vistas en la teoría. Estos ejercicios le permitirán al alumno familiarizarse con el entorno del cálculo distribuido (Llamado a funciones de MPI, compilación de código, ejecución de programas en paralelo, etc.), como así también con la metodología de programación en paralelo.

Además de estos ejercicios, se plantearán algunos problemas más complejos con el fin de que el alumno lo pueda resolver utilizando lo aprendido. Luego se analizarán y discutirán los resultados obtenidos.

Los trabajos prácticos permiten la evaluación continua del progreso de los alumnos, estimulando además la autoevaluación por parte del estudiante, como medio idóneo para aumentar su rendimiento, por ello será obligatoria la aprobación de los prácticos.

Listado de Actividades de Formación Práctica:

- Comunicación punto a punto.
- Determinar ancho de banda y de bisección de la red.
- Distribución de datos entre los procesadores.
- Asignaciones de tareas a los distintos procesadores.
- Comunicaciones colectivas.
- Paralelizar códigos escritos para su ejecución secuencial.
- Escribir funciones colectivas.
- Utilizar dichas funciones para la resolución de problemas.
- Implementar un método iterativo en paralelo para la resolución de sistemas lineales empleando MPI.
- Programación en entornos de memoria compartida.
- Comparar distintas formas de acceso a memoria.
- Implementar un método iterativo en paralelo para la resolución de sistemas lineales empleando OpenMP.

Intensidad de la formación práctica

Detalle de la carga horaria total prevista para cada una de las siguientes actividades:

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 1: 10 horas

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 2: 7 horas

Actividades prácticas que aportan a las competencias específicas en el Nivel de dominio 3: 0 horas

Horas totales de actividades de formación práctica: 17 horas

Metodología de Evaluación Durante el cursado:**SEGUNDO CUATRIMESTRE 2024**

La evaluación de los alumnos estará centrada en las evaluaciones teórico – prácticas, y en la participación del alumnos en la clase. Se realizarán dos evaluaciones parciales con recuperatorio. Los exámenes contendrán una parte teórica a realizarse de manera sincrónica con el docente y una parte práctica donde se dará un plazo de tiempo para su realización y luego debe ser remitido al docente vía correo electrónico. Se realizarán trabajos prácticos individuales. Los trabajos se presentarán por escrito, evaluándose también tiempo y forma; y luego requerirá de una defensa verbal por parte del alumno sobre los trabajos realizados. Se propondrán días y horarios de consultas adicionales, además del uso del foro del campus virtual. Los alumnos podrán acceder a la condición de promocionales, regulares o libres.

Evaluaciones:

- Dos exámenes parciales – Aprobación 60% - Promoción 80% (Solo hay un recuperatorio para promoción en caso de no haber obtenido una nota inferior a 70% en el otro parcial)
- Aprobación de todos los trabajo prácticos. Aprobación 60%

Metodología de Evaluación en Exámenes Finales:**SEGUNDO CUATRIMESTRE 2024**

La evaluación final de los alumnos que no hayan promocionado o se encuentren en condición libre se realizará mediante un examen teórico - práctico, que incluirá una serie de preguntas conceptuales y la resolución de un problema práctico.

Para el caso de aquellos alumnos que tengan condición regular la parte teórica y práctica estará orientada a uno de los dos temas principales (MPI o OpenMP) de la materia. Mientras que para aquellos alumnos que rindan el examen final en condición libre la parte teórica contendrá preguntas que requerirá el conocimiento de todos los temas de programa y la parte práctica requerirá realizar un programa empleando programación híbrida, o sea, MPI y OpenMP.

Condiciones de Regularidad :

Condiciones de Regularidad y Promoción:

El alumno como condición para regularizar la materia deberá aprobar (2) dos exámenes parciales con un promedio igual o superior al 60 %. Ambos podrán recuperarse de no alcanzar el mínimo exigido.

De obtener un promedio de ambos parciales del 80 % o superior y en ninguno obtuvo una calificación inferior a 70%, y además tener aprobados los Trabajos prácticos, obtendrá la promoción directa de la materia.

Para aprobar la materia el alumno deberá:

- 1) Tener aprobados todos los trabajos práctico, y aprobar el examen final en caso de poseer la condición de Regular,
- 2) Tener aprobados todos los trabajos práctico, y aprobar el examen final para la condición Libre.

Cronograma de parciales durante el primer Cuatrimestre:

Cronograma de parciales durante el segundo Cuatrimestre:

Primer Examen Parcial: 11 de Septiembre de 2024

Segundo Examen Parcial: 30 de Octubre de 2024

Recuperatorio 01: 13 de Noviembre de 2024

Recuperatorio 02: 20 de Noviembre de 2024

Bibliografía Principal:

- 1) Cálculo Paralelo con MPI, por Luciano Garelli y Mario Storti, CIMEC (2011).
- 2) An introduction to parallel programming, Tovia Wittwer, 2006. (Accesible desde <http://www.vssd.nl/hlf/a019.htm>)
- 3) Using MPI: Portable Parallel Programming with the Message Passing Interface, W. Gropp, E. Lusk and A. Skeljumm. MIT Press 1995
- 4) MPI: A Message-Passing Interface Standard, June 1995 (accessible desde <http://www.mpiforum.org>).
- 5) MPI-2: Extensions to the Message-Passing Interface November 1996, (accessible desde <http://www.mpiforum.org>).
- 6) MPI: the complete reference, por Marc Snir, Bill Gropp, MIT Press (1998) (disponible en formato electrónico, [mpi-book.ps](#), [mpi-book.pdf](#)).
- 7) Parallell Scientific Computing in C++ and MPI: A Seamless approach to parallel algorithms and their implementations, por G. Karniadakis y RM Kirby, Cambridge U Press (2003).
- 8) MPICH2 User's Guide. (accessible desde <http://www.mcs.anl.gov/research/projects/mpich2>).
- 9) An introduction into OpenMP, Ruud van der Pass (accessible desde http://www.nic.uoregon.edu/iwomp2005/iwomp2005_tutorial_openmp_rvdp.pdf).
- 10) Using OpenMP: Portable Shared Memory Parallel Programming. Barbara Chapman, Gabriele Jost, Ruud Van Der Pas, MIT Press (MA), 2007.
- 11) OpenMP.org Resources (accesible desde <http://openmp.org/wp/resources>).

Bibliografía Complementaria:

Equipo de Cátedra:

Profesor Responsable (Profesor Adjunto dedicación simple). Dr. Ing. Luciano Garelli

Responsable del dictado de las clases teóricas, prácticas y coloquios. Corrección de los prácticos propuestos.

Actividades de Investigación Gestión y Extensión:

Requisitos de admisión para alumnos oyentes:

Se admite la presencia de alumnos oyentes

Infraestructura, equipamiento y recursos necesarios:

SEGUNDO CUATRIMESTRE 2024

Durante el dictado del seminario se utilizará una PC por alumno con sistema operativo Linux, Compilador C++ (g++) y librería MPICH2.

Otros: