

Curso: Programación Paralela y Distribuida				
Profesores: Omar Antonio Hernández Duany, Máster en Ciencias, Profesor Auxiliar, Investigador Auxiliar, CUJAE				
Acerca del curso	Modalidad: A Distancia	Duración: 30 horas	Créditos: 2	Idioma: Español
Destinatarios	Desarrolladores de software que requieren acelerar procesos computacionales de diversos entornos académicos y empresariales del sector de las tecnologías informáticas y de comunicaciones.			
Fundamentación y objetivos	<p>El curso tiene como objetivo fundamental socializar las técnicas de análisis, diseño y programación paralela y distribuida con el propósito de contribuir a que los desarrolladores de software puedan trabajar en función de maximizar el aprovechamiento de las arquitecturas de hardware emergentes, que usualmente son portadoras de procesadores con múltiples núcleos (CPU), o unidades de procesamiento gráfico (GPU). Estos recursos de hardware están presentes en muchos medios de cómputo de la actualidad como son: computadoras masivamente paralelas, clúster de computadoras de alto rendimiento (HPCC por sus siglas en inglés), servidores, computadoras personales, teléfonos móviles, dispositivos lógicos programables, que no siempre son aprovechados óptimamente. Es importante considerar que en el sector de las tecnologías informáticas y de comunicaciones existen muchas áreas de investigación, desarrollo e innovación tecnológica en las que se requiere desarrollar o gestionar aplicaciones que deben minimizar los tiempos de ejecución de sus procesos intrínsecos, basado en el aprovechamiento de los rendimientos picos teóricos de las infraestructuras computacionales disponibles. Los avances tecnológicos que experimenta este campo de las ciencias repercute notablemente en el aumento de la efectividad de los sistemas de gestión de las organizaciones, con la consiguiente mejora de sus indicadores productivos y la disminución de los costos de sus procesos intrínsecos que pueden contribuir a la amortización más rápida de las inversiones tecnológicas, la generación de utilidades y la consiguiente ampliación sistemática del capital productivo.</p>			
Contenidos del curso y bibliografía	Temas del curso 1. Arquitecturas paralelas y distribuidas.			

2. Análisis del rendimiento.
3. Diseño de algoritmos paralelos y distribuidos.
4. Taller I: Análisis de complejidad computacional de problemas, rendimientos y criterios de diseño paralelos y distribuidos. Presentación de problemas a solucionar por equipos.
5. Programación OpenMP.
6. Programación MPI.
7. Programación GPU: OpenCL y CUDA.
8. Vectorización de aplicaciones.
9. Programación híbrida.
10. Taller II: Discusión de proyectos finales.

Bibliografía

1. Schmidt B., Gonzalez-Dominguez J., Hundt C., Schlarb M., "Parallel Programming: Concepts and Practice", Morgan Kaufmann, 2017.
2. McCool M., Robinson A., Reinders J., "Structured Parallel Programming Patterns for Efficient Computation", Morgan Kaufmann, Elsevier, 2012.
3. Kirk David, Hwu Wen-mej, "Programming massively parallel processors: A Hands-on Approach", Elsevier, 2010
4. Gebali F., "Algorithms and Parallel Computing", Wiley, 2011.
5. Mattson T. G, "Patterns for Parallel Programming", Software Patterns Series, 2004.
6. Andrist B., Sehr V., "C++ High Performance Master the art of optimizing the functioning of your C++ code", Packt Publishing, 2020.
7. Robey R., Zamora Y., "Parallel and High Performance Computing", Manning, 2021.
8. Grama A., Gupta A., Karypis G., Kumar V., "Introduction to parallel computing", Addison-Wesley, 2018.
9. Balaji P., "Programming Models for Parallel Computing", The MIT Press, 2019.
10. Jaegeun Han, Bharatkumar Sharma, "Learn CUDA Programming", Packt Publishing, 2019.
11. Ruenger G., Rauber T., "Parallel programming for Multicore and Cluster Systems", Springer, 2013.
12. Foster I., "Designing and building parallel program", Springer, 2010.

	<p>13. Marlow S., "Parallel and Concurrent Programming in Haskell: Techniques for Multicore and Multithread", O'REILLY, 2011.</p> <p>14. Barlas G., "Multicore and GPU Programming: An integrated Approach", Elsevier, 2010.</p> <p>15. Fagerlund OA., "Multi-core programming with OpenCL performance and portability", 2010.</p> <p>16. Michael McCool, Arch D. Robison, James Reinders, "Structured Parallel Programs", Elsevier Inc. 2012.</p> <p>17. George Em karniadakis, Robert M. Kirby, "Parallel scientific computing in C++ and MPI", Cambridge University Press, 2003.</p> <p>18. Aaftab Munshi, Benedict R. Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg, "The OpenCL programming book", Pearson Education Inc., 2012.</p> <p>19. Aaftab Munshi Benedict R. Gaster, Timothy G. Mattson, James Fung, Dan Ginsburg; "OpenCL Programming guide", Pearson Education, Inc, 2012.</p> <p>20. Maurice Herlihy, Nir Shavit, "The Art of Multiprocessor Programming", Elsevier, Inc, 2012.</p> <p>21. Deitel, Cómo programar en C/C++, 6ta edición, Deitel & Associates (Harvey & Paul), Prentice Hall, 2008.</p>
<p>Síntesis sobre desarrollo del curso e interacción con actores, elementos del cronograma</p>	<p>Síntesis sobre desarrollo del curso</p> <p>El curso se impartirá mediante 10 actividades: 8 conferencias y 2 talleres. Los estudiantes tendrán una participación activa en cada actividad en las que se promueve el análisis y discusión de problemas comunes de este campo de investigación en correspondencia con los objetivos propuestos.</p> <p>La evaluación del curso se realizará de forma sistemática basado en la medición de la comprensión gradual de los aspectos teóricos y los avances en cada una de las etapas de desarrollo de la solución al problema que presentarán como proyecto final. Los proyectos finales deberán resolver un problema que describe a priori una elevada complejidad computacional.</p> <p>En el Taller I los estudiantes presentarán el problema que pretenden solucionar mediante la aceleración u optimización de la aplicación empleando el conocimiento de los temas objetos de estudio. En el Taller II se realizará la evaluación integral del proyecto con el objetivo de medir el grado de asimilación de los métodos de análisis, diseño y utilización de los paradigmas de programación empleados en la solución del problema planteado. También se</p>

realizará la evaluación de la aplicación construida a partir de analizar el rendimiento y la eficiencia alcanzados como resultado del empleo de los paradigmas de programación paralela y distribuida aprendidos durante el curso. Los estudiantes demostrarán en qué proporción han logrado acelerar sus respectivos procesos en función de las arquitecturas disponibles. Podrán solucionarse problemas heterogéneos de sus respectivos campos de investigación y desarrollo.

Aspectos orientadores

El objetivo del proceso de evaluación es propiciar que los estudiantes demuestren los conocimientos, habilidades y competencias adquiridos para el desarrollo de aplicaciones paralelas y distribuidas a partir de los objetivos esbozados en el curso. Las implementaciones podrán realizarse preferiblemente en lenguaje C/C++.

Para la implementación del proyecto final los estudiantes solucionarán un problema específico de su campo de investigación o afín. Se acepta la presentación de una implementación previamente desarrollada accesible a través de internet u otras fuentes en cuyo caso deberá demostrarse el dominio de cada uno de sus componentes, criterios de diseño e implementación empleados, así como las características de su proceso de desarrollo entre otros aspectos.

En el Taller I los estudiantes someterán a discusión el problema a resolver para el proyecto final empleando los conceptos introducidos en las tres primeras conferencias. Se realizará el análisis del proceso de solución de un problema típico como referencia para la comprensión del proceso de desarrollo de una aplicación paralela y distribuida. Pueden conformarse equipos de trabajo de hasta tres integrantes.

En cada uno de los temas se presentarán los avances del proceso de desarrollo del proyecto final en base al problema propuesto originalmente a partir de considerar los objetivos propuestos en el tema precedente, de modo que puedan consolidarse sistemáticamente los conocimientos adquiridos.

	<p>En el Taller II: Como actividad final los estudiantes demostrarán la funcionalidad de la aplicación paralela desarrollada o asimilada, realizándose la evaluación de las métricas de rendimiento alcanzado por la aplicación en función de la arquitectura de hardware disponible.</p>
Acerca de los profesores	<p>El profesor es Licenciado en Análisis de Sistemas Especiales de Comunicaciones (1986) y Licenciado en Cibernética Matemática, UH, en el año (1989). Máster en Ciencias en Procesamiento Digital de Señales (2005). Es profesor, investigador, analista de sistemas y programador en la Facultad de Telecomunicaciones y Electrónica de la Universidad Tecnológica de la Habana (CUJAE) donde se desempeña como jefe de la disciplina de programación y del Laboratorio de Cómputo de Alto Rendimiento para las Telecomunicaciones, integrante de la Comisión Nacional de Carrera y del Comité Editorial de la revista Telemática.</p>