



**MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN
PROPUESTA DE CURSO DE POSGRADO**

1- DATOS GENERALES DE LA ACTIVIDAD CURRICULAR					
1.1 Título del Curso	ANÁLISIS DE RENDIMIENTO DE APLICACIONES PARALELAS				
1.2 Área temática ¹	Arquitecturas, Sistemas Operativos y Redes.				
2- COMPOSICIÓN DEL EQUIPO DOCENTE					
2.1 Responsable a cargo de la actividad curricular	Dr. Javier Balladini				
2.2 Docentes	Dr. Javier Balladini – Dr. Emmanuel Frati				
3- CARGA HORARIA					
Carga horaria teórica	30 hs				
Carga horaria práctica	30 hs				
Carga horaria total	60 hs				
Distribución horaria: Cursado: 40 hs (2 semanas) Preparación trabajo final: 20 hs	Lu 4 hs	Ma 4 hs (autoguiado)	Mie 4 hs.	Jue 4 hs (autoguiado)	Vie 4 hs
Fecha de inicio sugerida	16 de Agosto de 2021				
4- BREVE RESUMEN DE CONTENIDOS					
<p>La computación paralela refiere a un tipo de cómputo en el que los cálculos se realizan de forma simultánea. Si bien el paralelismo ha sido empleado históricamente en la computación de altas prestaciones, ha ganado un enorme interés debido al impedimento para seguir aumentando la frecuencia de reloj de los procesadores. La solución, para continuar aumentando la velocidad de ejecución de las aplicaciones, fue incrementar la cantidad de unidades de procesamiento, dando así lugar a la aparición de procesadores multinúcleos. Desde entonces, la computación paralela se ha convertido en el paradigma dominante en la arquitectura de computadoras.</p> <p>Para algunas aplicaciones, será suficiente con utilizar una plataforma comprendida por una única computadora con uno o más procesadores multinúcleos. En otros casos, será necesario el poder de cómputo de un cluster de computadoras. Aprovechar al máximo el potencial de estas plataformas es un gran desafío debido a las complejas interacciones que se producen entre el hardware y el software. En este contexto, el análisis de rendimiento juega un rol central en el proceso de desarrollo de aplicaciones paralelas.</p>					
5- CONOCIMIENTOS PREVIOS REQUERIDOS					
<ul style="list-style-type: none">● Programación en C/C++● Arquitectura de computadoras, sistemas operativos y redes.					
6- OBJETIVOS					
<p>El objetivo de este curso es que el asistente aprenda a:</p> <ul style="list-style-type: none">● Evaluar el desempeño de una aplicación paralela a través de las métricas asociadas al paralelismo.● Detectar oportunidades de mejora en aplicaciones paralelas usando herramientas de análisis de rendimiento.● Proponer modificaciones en programas paralelos, guiadas por la evaluación realizada con las herramientas de análisis, con el fin de mejorar su rendimiento.					

¹ Corresponde a uno de los siguientes tópicos: Algoritmos y Lenguajes; Teoría de la Computación; Ingeniería de Software, Bases de Datos y Sistemas de Información; Arquitecturas, Sistemas Operativos y Redes.



7- CONTENIDOS (organizados en unidades)
<p>1- Introducción: Clasificación de máquinas paralelas: Taxonomía de Flynn y Memoria compartida vs. distribuida. Plataformas de cómputo paralelo. Evaluación de rendimiento de sistemas paralelos y aplicaciones paralelas. Métricas. Optimización de programas para procesadores de propósito general. Instrucciones vectoriales.</p> <p>2- Modelos de programación paralela: Modelo de programación paralela de memoria compartida OpenMP. Modelo de programación paralela de memoria distribuida MPI. Solapamiento Cómputo/Comunicaciones.</p> <p>3- Análisis de rendimiento: Factores que afectan el rendimiento computacional. Balance de carga. Sobrecarga por paralelización. Comunicaciones. Accesos a memoria (localidad espacial y temporal, falsa compartición). Mapping. Modelo roofline. Herramientas. Clasificación. Profiling. Contadores de hardware. Tracing. Instrumentación de aplicaciones.</p>
8- PROPUESTA DIDÁCTICA (metodología de trabajo de clases teóricas y prácticas)
<p>Inicialmente se trabajará sobre una introducción a la arquitectura de computadoras paralelas para luego desarrollar casos de ejemplos de programas en los modelos de programación tradicionales: memoria distribuida y memoria compartida. En cada caso se trabajará sobre los aspectos a considerar en una solución paralela en cada modelo (sincronización y comunicación de tareas) y a los aspectos que definen el rendimiento del programa resultante. Luego se trabajará sobre técnicas y métodos que permiten obtener información sobre la ejecución de estos programas, pasando desde una instrumentación manual de los programas hasta el uso de herramientas diseñadas con ese fin. Todas el trabajo experimental se realizará en un cluster con todas las herramientas necesarias previamente instaladas, el cual será accedido mediante protocolo SSH por los estudiantes. Se buscará conocer en detalle las opciones disponibles destacando las ventajas y reconociendo las desventajas de cada tecnología. Como parte del cursado se deberá desarrollar un proyecto simple, complementado con actividades prácticas a ser realizadas principalmente durante las clases presenciales. Cada actividad se cerrará con un espacio de reflexión sobre lo trabajado. Para aprobar cada actividad el estudiante deberá demostrar capacidades sobre el uso de esta tecnología.</p>
9- MODALIDAD DE EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN²
<p>Para la aprobación se solicitará la resolución de actividades prácticas y la elaboración de un proyecto final. Las actividades prácticas serán realizadas durante el desarrollo del curso. El proyecto final será realizado de forma autónoma por los estudiantes en un plazo máximo de 45 días a partir de la finalización del curso. Además, para poder hacer el proyecto final los estudiantes deberán acreditar como mínimo un 80% de asistencia a clases y un 60% de los trabajos prácticos propuestos resueltos.</p>
10- BIBLIOGRAFÍA DE LECTURA OBLIGATORIA CORRESPONDIENTE A CADA UNIDAD Y GENERAL
<p>Unidad 1 y 2:</p> <ul style="list-style-type: none">● Grama, A; Gupta, A; Karypis, G.; Kumar, V. (2003). Introduction to Parallel Computing. Pearson Addison Wesley, 2da edición.● Dongarra, Foster, Fox, Gropp, Kennedy, Torczon, White. Morgan Kauffman. (2003). Sourcebook of Parallel Computing.● Blaise Barney, L. (2020). Introduction to Parallel Computing. Livermore National Laboratory. Recurso On-line (https://computing.llnl.gov/tutorials/parallel_comp/)● Rauber, T., & Rüniger, G. (2010). Parallel Programming: for multicore and cluster systems. Springer.● Pacheco, P. (2011). An introduction to parallel programming. Elsevier.● Rohit Chandra et al. (2000). Parallel Programming in OpenMP. Morgan Kaufmann. ISBN-10: 1558606718 <p>Unidad 3:</p> <ul style="list-style-type: none">● BSC-Tools (2020). Performance Analysis Tools: Details and Intelligence BSC-Tools. Disponible en: https://tools.bsc.es/

² Son condiciones mínimas para la aprobación de todos los cursos: cumplir con un mínimo del 80% de asistencia a las clases, realizar las tareas y aprobar las evaluaciones que se hayan propuesto en el programa, con una calificación no menor a 7 (puntos). Los trabajos de evaluación pautados y la calificación de los alumnos deberán realizarse dentro de los 60 días posteriores a la finalización del curso.



- Garner, B. D., Browne, S., Dongarra, J., Garner, N., Ho, G., y Mucci, P. (2000). A portable programming interface for performance evaluation on modern processors. *The International Journal of High Performance Computing Applications*, 14, 189–204.
- Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., . . . Hazelwood, K. (2005). Pin: building customized program analysis tools with dynamic instrumentation. En *Proceedings of the 2005 acm sigplan conference on programming language design and implementation* (pp. 190–200). ACM.
- perf wiki. (2020). Tutorial perf. Descargado de <https://perf.wiki.kernel.org/index.php/Tutorial>
- Virtual Institute (2017). Tools Guide. High Productivity Supercomputing (VI-HPS). <http://www.vi-hps.org/upload/material/general/ToolsGuide.pdf>
- Jerry Eriksson, Pedro Ojeda-May, Thomas Ponweiser, Thomas Steinreiter (2017). Profiling and Tracing Tools for Performance Analysis of Large Scale Applications. *Partnership for Advanced Computing in Europe (PRACE)*. 2017.
- Isaacs, K. E., Giménez, A., Jusufi, I., Gamblin, T., Bhatele, A., Schulz, M., ... & Bremer, P. T. (2014). State of the Art of Performance Visualization. In *EuroVis (STARS)*.
- Samuel Williams, Andrew Waterman, and David Patterson (2009). Roofline: an insightful visual performance model for multicore architectures. *Commun. ACM* 52, 4 (April 2009), 65–76.
- Morán, M., Balladini, J., Rexachs, D., & Luque, E. (2019). Prediction of Energy Consumption by Checkpoint/Restart in HPC. *IEEE Access*, 7, 71791-71803.
- Frati, F. E. (2014). Evaluación de técnicas de detección de errores en programas concurrentes [Tesis, Facultad de Informática]. <http://hdl.handle.net/10915/36923>

11- INFRAESTRUCTURA E INSUMOS REQUERIDOS³

Se proveerá, a cada asistente, una cuenta de acceso al Cluster CDER del NSF/IEEE-TCPP (<https://tcpp.cs.gsu.edu/curriculum/?q=node/21615>). Cada estudiante deberá disponer de una computadora con acceso a Internet para utilizar el cluster.

12 – OTRA INFORMACIÓN RELEVANTE

³ Deberá constar aquí si la realización del curso requiere contar con instalaciones especiales (laboratorio, sala de informática, equipamiento audiovisual, etc). Explicitar si se estima que el curso debe tener un número máximo determinado de asistentes para poder ser dictado.