

**ANÁLISIS DE RENDIMIENTO DE  
APLICACIONES PARALELAS**

Año 2018

**Docentes Responsables:**

Dr. Javier Balladini - Dr. Emmanuel Frati

**Carrera:**

Doctorado en Ciencias Informáticas.

**Créditos:** 4**Duración:** 70 horas**DESTINATARIOS**

- Profesionales con título de grado/postgrado en carreras afines a la disciplina informática.

Requisitos:

- Conocimientos de programación en C/C++
- Conocimientos básicos sobre arquitectura de computadoras, sistemas operativos y redes.
- Conocimientos generales de programación paralela - OpenMP / OpenMPI

**FUNDAMENTACIÓN**

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.

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.

**OBJETIVOS GENERALES**

El objetivo de este curso es que el asistente aprenda a:

- 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.

### **CONTENIDOS MÍNIMOS**

- Características esperadas de las aplicaciones paralelas. Métricas de rendimiento.
- Arquitecturas de máquinas paralelas.
- Instrucciones vectoriales.
- Modelo de programación paralela de memoria compartida.
- Modelo de programación paralela de memoria distribuida.
- Profiling. Contadores de hardware. Tracing
- Instrumentación de aplicaciones.
- Herramientas para análisis de rendimiento.
- Localidad espacial y temporal.
- Falsa compartición.
- Aplicaciones limitadas por CPU.
- Aplicaciones limitadas por Memoria.
- Comunicaciones.
- Solapamiento Cómputo/Comunicaciones.

### **METODOLOGÍA DE ENSEÑANZA**

Se presentan los conceptos en clases teórico-prácticas expositivas por medio de transparencias. Los asistentes trabajan en sus equipos siguiendo las indicaciones de los docentes a cargo.

Duración:

- 20 horas de clases teórico-prácticas presenciales.
- 50 horas de trabajo fuera de clase (a realizar en un plazo máximo de seis meses)

### **EVALUACIÓN**

Está contemplada una evaluación durante las dos últimas horas del curso, que será complementada con un trabajo de investigación (de realización no presencial) para completar las horas previstas del curso, con soporte virtual de los profesores. La fecha límite de entrega será seis meses después de realizado el curso. La calificación final se determinará basándose en la siguiente ponderación:

- Evaluación presencial: 50% de la calificación
- Trabajo no presencial: 50% de la calificación



## **RECURSOS NECESARIOS**

Las clases se imparten con proyector, y es necesario el uso de un laboratorio de informática (preferentemente equipado con una computadora por alumno) con acceso a Internet.

El sistema operativo de cada máquina debe ser Linux (debian/ubuntu/mint) y debe tener instalados los siguientes paquetes:

- Herramientas de desarrollo (build-essentials)
- openSSH
- Extrae-Paraver
- Perf
- PAPI

## **BIBLIOGRAFÍA**

- Sourcebook of Parallel Computing. Dongarra, Foster, Fox, Gropp, Kennedy, Torczon, White. Morgan Kaufman. 2003.
- Introduction to Parallel Computing. Ananth Grama, Anshul Gupta, George Karypis, Vipin Kumar. Pearson Addison Wesley, 2da edición. 2003.
- Parallel Programming: for multicore and cluster systems. Rauber, T., & Rüniger, G. Springer. 2010.
- An introduction to parallel programming. Pacheco, P. Elsevier. 2011
- Performance Analysis Tools: Details and Intelligence | BSC-Tools. [En línea]. Disponible en: <https://tools.bsc.es/>
- Introduction to Parallel Computing. Blaise Barney, Lawrence Livermore National Laboratory. Recurso On-line ([https://computing.llnl.gov/tutorials/parallel\\_comp/](https://computing.llnl.gov/tutorials/parallel_comp/))
- Parallel Programming in OpenMP. Rohit Chandra et al. 2000. Morgan Kaufmann. ISBN-10: 1558606718
- Tools Guide - 2017. Virtual Institute – High Productivity Supercomputing (VI-HPS). <http://www.vi-hps.org/upload/material/general/ToolsGuide.pdf>
- Profiling and Tracing Tools for Performance Analysis of Large Scale Applications. Jerry Eriksson, Pedro Ojeda-May, Thomas Ponweiser, Thomas Steinreiter. Partnership for Advanced Computing in Europe (PRACE). 2017. <http://www.prace-ri.eu/IMG/pdf/WP237.pdf>
- Tutorial - perf wiki. (s.f.). <https://perf.wiki.kernel.org/index.php/Tutorial>. Descargado 2018-05-03, de <https://perf.wiki.kernel.org/index.php/Tutorial>
- Pin: building customized program analysis tools with dynamic instrumentation. Luk, C.-K., Cohn, R., Muth, R., Patil, H., Klauser, A., Lowney, G., . . . Hazelwood, K. 2005. En Proceedings of the 2005 acm sigplan conference on programming language design and implementation (pp. 190–200). ACM.
- A portable programming interface for performance evaluation on modern processors. Garner, B. D., Browne, S., Dongarra, J., Garner, N., Ho, G., y Mucci, P. 2000. The International Journal of High Performance Computing Applications, 14 , 189–204.