

Curso: **Supercomputación**

Prof.: Fernando Díaz; Saturnino Vicente

Descripción

Objetivos:

Un sistema paralelo de computación o multiprocesador es un conjunto de procesadores de algún tipo que ejecutan un mismo programa repartido entre ellos. Esta idea es un tema clásico en arquitectura de computadores, ya que si la paralelización del programa o tarea se lleva a cabo con éxito, entonces el *rendimiento* (la inversa del tiempo de ejecución) aumentará considerablemente. Pero este límite no es fácil de conseguir, y las principales barreras provienen tanto del software como del hardware: bajo grado de paralelismo de la mayoría de aplicaciones, dificultad de escribir programas paralelos, elevado coste de los sistemas paralelos, etc. Debido a lo anterior, la búsqueda de un sistema paralelo de computación eficiente ha sido, un clásico en la historia de la Informática.

En esta asignatura se profundiza en el estudio tanto de las arquitecturas de computadores avanzadas, como de su programación. Se mostrará al alumno una revisión de las arquitecturas y técnicas hardware más avanzadas, mostrándole como influyen estas en las aplicaciones modernas. Además, se hará hincapié en como generar y optimizar las aplicaciones para que exploten el paralelismo de las nuevas arquitecturas.

Contenidos:

- Introducción: Monoprocesadores frente a multiprocesadores
- Paralelismo de datos y de instrucciones
- Tipos de multiprocesadores paralelos
- Evolución de las prestaciones del software y del hardware.
- Aplicaciones actuales
- Aplicaciones futuras
- Modelos de computación paralela
- Laboratorio de supercomputación

Metodología:

En el curso se planifican aproximadamente 30 horas. De estas, 8 corresponden a clases teóricas presenciales, 10 a prácticas supervisadas en un laboratorio, y 12. horas destinadas a trabajo personal del alumno. Incluyendo, no solamente la preparación de las prácticas, sino la realización de un trabajo escrito sobre un tema asociado al curso.

Las clases teóricas presenciales se dividen en cuatro clases de dos horas de duración.

Tras la explicación de los conceptos teóricos, se verán en el laboratorio algunos ejemplos de programación paralela, usando un cluster de 10 nodos. La programación se hará principalmente a tres niveles:

- Ejecuciones paramétricas, usando Enfuzion.
- Comunicación a través de tuberías estándar Unix (usando el sistema operativo OpenMosix).
- Librerías de paso de mensajes. Se verán ejemplos gráficos usando Xpvm y algunos en MPI.

A cada alumno del curso se le abrirá una cuenta en el cluster, para que después pueda desarrollar sus propias aplicaciones paralelas. El trabajo a realizar por el alumno se elige de entre una serie de propuestas que realiza el profesor. El alumno deberá exponer los resultados al resto de los alumnos en una sesión especial. El tiempo máximo de exposición será de 20 minutos.

Criterios de evaluación:

En la evaluación se tienen presentes tres aspectos: la asistencia y participación en las clases presenciales, la realización de prácticas supervisadas y la entrega de trabajos realizados, así como su exposición.

En cuanto al trabajo realizado y su posterior exposición, existirá una doble puntuación, por un lado la que obtenga de los profesores de la asignatura, y por otro lado, la que sus compañeros de curso le asignen.

Se considerará que un alumno ha superado el curso cuando la calificación global que obtiene en el sistema de evaluación sea igual o superior a 5 sobre una nota máxima de 10.

Bibliografía básica:

- Julio Ortega Lopera; Mancia Anguita López; Alberto Prieto Espinosa. Arquitectura De Computadores. Edit. Paraninfo. (2005). ISBN: 8497322746.
- Shen, John Paul & Lipasti Mikko H. Arquitectura De Computadores. (Editorial McGraw-Hill) 1ª edición (FEB-06). ISBN: 8448146425.
- D. Culler, J. P. Singh (with A. Gupta). Parallel Computer Architecture, A Hardware/Software Approach. Edit. Morgan Kaufmann. 1999.
- A.S. Tanenbaum. Modern Operating Systems. 2nd Ed. Prentice Hall PTR. 2001
- J.L. Hennessy, Patterson "Computer Architecture. A Quantitative Approach". Morgan-Kaufmann (Fourth Edition). 2006
- Patterson, Hennessy. Estructura y Diseño de Computadores. Interfaz Circuitería-Programación. Reverté. 2000.
- A. S. Tanenbaum. Structured Computer Organization. Prentice Hall. 1999. Quinta Edición.
- Intel_Tutorial Intel Architecture Optimization Manuals. Disponible en www.intel.com
- Parallel Programming in OpenMP. R. Chandra, D. Kohr, L. Dagum, et al. Morgan Kaufmann. ISBN 1558606718. 2000
- Introduction to Parallel Computing. Design and Análisis of Algorithms. V. Kumar, A. Grama, A. Gupta, G. Karypis. Benjamin/Cummings. ISBN: 0805331700. 1994
- The Message Passing Interface (MPI) standard <http://www-unix.mcs.anl.gov/mpl/>
- A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, V. Sunderam. PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing. Edit. MIT Press. Serie: Scientific and Engineering Computation. 1996.
- M. Snir, S. W. Otto, S. Huss-Lederman, D. W. Walker, J. Dongarra. MPI: The Complete Reference. Edit. MIT Press. Serie: Scientific and Engineering Computation. 1996.
- Gregory F. Pfister. In Search Of Clusters. Edit. Prentice-Hall. 1998.