

Nociones básicas de computación paralela

Javier Cuenca¹, Domingo Giménez²

¹Departamento de Ingeniería y Tecnología de Computadores
Universidad de Murcia

²Departamento de Informática y Sistemas
Universidad de Murcia

domingo@um.es, <http://dis.um.es/~domingo>

Contenido

- 1 Planificación y Bibliografía
- 2 Sistemas paralelos
- 3 Sistemas multicore
- 4 Posibles trabajos

Planificación

5 de noviembre

17:00-18:00: Nociones generales y programación en multicore.

Domingo Giménez

18:00-19:00: Programación en multicore con OpenMP. Javier

Cuenca

19:00-21:00: Prácticas de programación con OpenMP (opcional).

Javier Cuenca

19 de noviembre

17:00-18:00: El supercomputador Ben-Arabí. Personal del Centro de Supercomputación de la Fundación Parque Científico de Murcia

18:00-19:00: Programación en clusters con MPI. Javier Cuenca

19:00-21:00: Prácticas de programación con MPI (opcional). Javier

Cuenca

Planificación

5 de noviembre

17:00-18:00: Nociones generales y programación en multicore.

Domingo Giménez

18:00-19:00: Programación en multicore con OpenMP. Javier

Cuenca

19:00-21:00: Prácticas de programación con OpenMP (opcional).

Javier Cuenca

19 de noviembre

17:00-18:00: El supercomputador Ben-Arabí. Personal del Centro de Supercomputación de la Fundación Parque Científico de Murcia

18:00-19:00: Programación en clusters con MPI. Javier Cuenca

19:00-21:00: Prácticas de programación con MPI (opcional). Javier

Cuenca

Bibliografía

- Francisco Almeida, Domingo Giménez, José Miguel Mantas, Antonio M. Vidal: Introducción a la programación paralela, Paraninfo Cengage Learning, 2008
 - Capítulos 1 y 2: introducción y computadores paralelos
 - Capítulo 3: programación con OpenMP y MPI
 - Códigos de los capítulos 3, 4 y 6 en la página de material suplementario de la editorial (www.paraninfo.es)
- y otros (mejores) en inglés:
 - Introduction to Parallel Computing. Grama, Gupta, Karypis, Kumar. Addison-Wesley. 2003
 - Parallel Programming in C with MPI and OpenMP. Quinn. McGrawHill. 2003

Bibliografía

- Francisco Almeida, Domingo Giménez, José Miguel Mantas, Antonio M. Vidal: Introducción a la programación paralela, Paraninfo Cengage Learning, 2008
 - Capítulos 1 y 2: introducción y computadores paralelos
 - Capítulo 3: programación con OpenMP y MPI
 - Códigos de los capítulos 3, 4 y 6 en la página de material suplementario de la editorial (www.paraninfo.es)
- y otros (mejores) en inglés:
 - Introduction to Parallel Computing. Grama, Gupta, Karypis, Kumar. Addison-Wesley. 2003
 - Parallel Programming in C with MPI and OpenMP. Quinn. McGrawHill. 2003

y material adicional

- Curso de Algoritmos y Programación Paralela, quinto Informática, dis.um.es/~domingo/app.html
Incluye esquemas algorítmicos paralelos
- Programación Paralela y Computación de Altas Prestaciones, máster de Informática, dis.um.es/~domingo/cap.html
Incluye algoritmos matriciales paralelos y librerías matriciales
- Curso de promoción Computación Científica en Clusters, 2010, www.ditec.um.es/~javiercm/curso_ccc/index.html
Incluye administración de clusters y programación en GPU
- Curso de promoción de Programación en el Supercomputador Ben-Arabí, marzo 2011, <https://casiopea.um.es>

y material adicional

- Curso de Algoritmos y Programación Paralela, quinto Informática, dis.um.es/~domingo/app.html
Incluye esquemas algorítmicos paralelos
- Programación Paralela y Computación de Altas Prestaciones, máster de Informática, dis.um.es/~domingo/cap.html
Incluye algoritmos matriciales paralelos y librerías matriciales
- Curso de promoción Computación Científica en Clusters, 2010, www.ditec.um.es/~javiercm/curso_ccc/index.html
Incluye administración de clusters y programación en GPU
- Curso de promoción de Programación en el Supercomputador Ben-Arabí, marzo 2011, <https://casiopea.um.es>

y material adicional

- Curso de Algoritmos y Programación Paralela, quinto Informática, dis.um.es/~domingo/app.html
Incluye esquemas algorítmicos paralelos
- Programación Paralela y Computación de Altas Prestaciones, máster de Informática, dis.um.es/~domingo/cap.html
Incluye algoritmos matriciales paralelos y librerías matriciales
- Curso de promoción Computación Científica en Clusters, 2010, www.ditec.um.es/~javiercm/curso_ccc/index.html
Incluye administración de clusters y programación en GPU
- Curso de promoción de Programación en el Supercomputador Ben-Arabí, marzo 2011, <https://casiopea.um.es>

Programación paralela

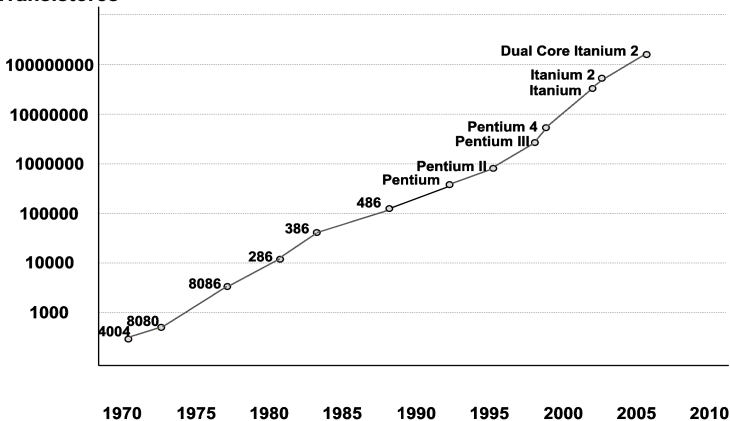
Uso de varios procesadores trabajando juntos para resolver una tarea común

- Cada procesador trabaja en una porción del problema
- Los procesos pueden intercambiar datos, a través de
 - La memoria: modelo de memoria compartida, OpenMP, esta sesión
 - Una red de interconexión: modelo de paso de mensajes, MPI, próximo viernes

¿Por qué programación paralela?

- Límites de memoria en los procesadores secuenciales
- Aceleración de la computación
- Dificultad de integración de componentes

Transistores



Aplicaciones de la programación paralela

- Por necesidades de memoria: NUMA, clusters y supercomputadores
- Problemas con grandes necesidades de computación
 - Alto coste: 2^n , $n!$,... Métodos aproximados, NUMA, clusters, supercomputadores, grid
 - Gran dimensión: n^3 y $n \geq 10000$. NUMA, clusters, supercomputadores
 - Tiempo real: coste polinomial. Multicore, GPU, procesadores dedicados

Aplicaciones de la programación paralela

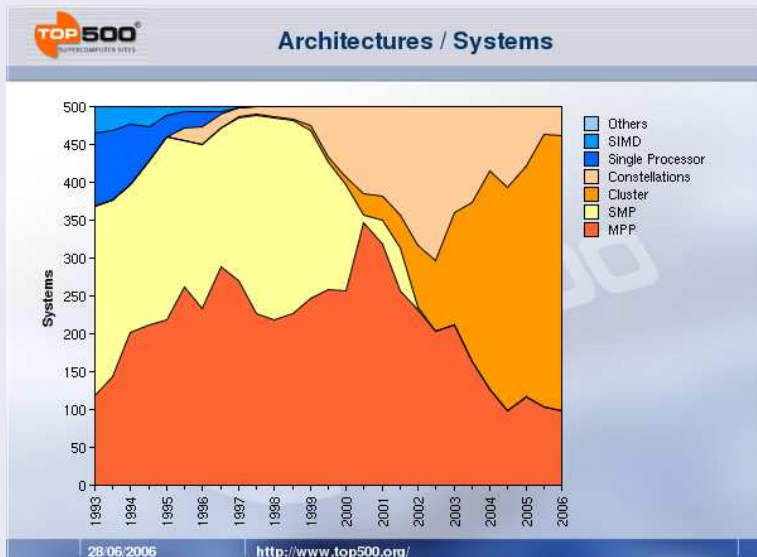
- Simulaciones: climáticas, de hidrodinámica, de moléculas, de contaminación...
- Problemas combinatorios, metaheurísticas: diseño de filtros, localización de centros de servicio, planificación de trabajos, análisis ADN...
- Codificación y tratamiento de datos: control de recursos, decodificación TDT, sistemas móviles y MIMO...

Sistemas paralelos

Sistema	Cores	Programación
portátil	2-4	OpenMP
PC	2-8	OpenMP
servidor	4-24	OpenMP
NUMA (Ben)	16-256	OpenMP
cluster	2-32 × 2-8	MPI+OpenMP
supercomputador (Arabí)	32-1024 × 8-32	MPI+OpenMP
GPU	112-512	CUDA
heterogéneo	cluster+NUMA	MPI+OpenMP
jerárquico	cluster+NUMA+GPU	MPI+OpenMP+CUDA
⋮		

Lista de máquinas más potentes

TOP500 (<http://www.top500.org>)



Arquitectura multicore

Machine (191GB)

NUMANode #0 (48GB)

Socket #0

L3 #0 (16MB)

L2 #0 (3072KB)

L2 #1 (3072KB)

L2 #2 (3072KB)

L1 #0 (32KB)

L1 #1 (32KB)

L1 #2 (32KB)

L1 #3 (32KB)

L1 #4 (32KB)

L1 #5 (32KB)

Core #0

Core #1

Core #2

Core #3

Core #4

Core #5

PU #0

PU #1

PU #2

PU #3

PU #4

PU #5

Más ejemplos en hwloc (<http://www.open-mpi.org/projects/hwloc/>)

Compilación

- gnu C, a partir de la versión 4
 - gcc programa.c -fopenmp
- Intel C
 - icc programa.c -openmp

Ejecución

- Seleccionar el número de threads
 - Antes de la ejecución: establecer variable de entorno OMP_NUM_THREADS
 - Durante la ejecución: con función openmp `omp_set_numthreads()`

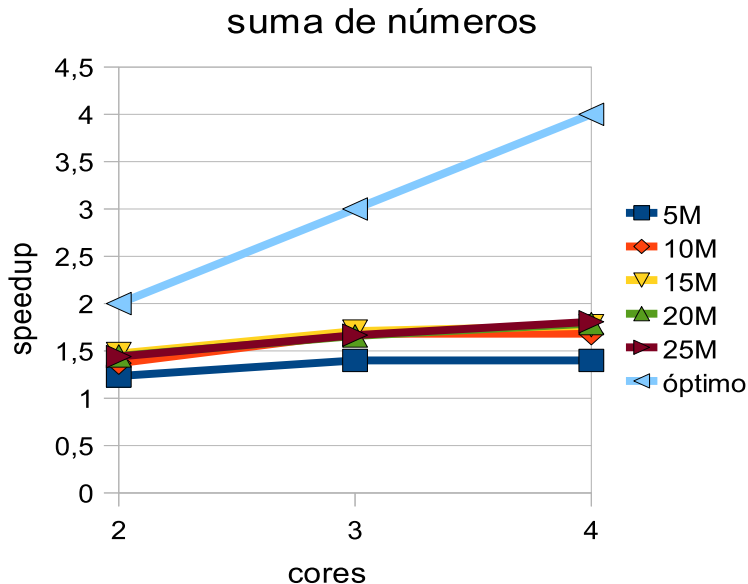
Compilación

- gnu C, a partir de la versión 4
 - gcc programa.c -fopenmp
- Intel C
 - icc programa.c -openmp

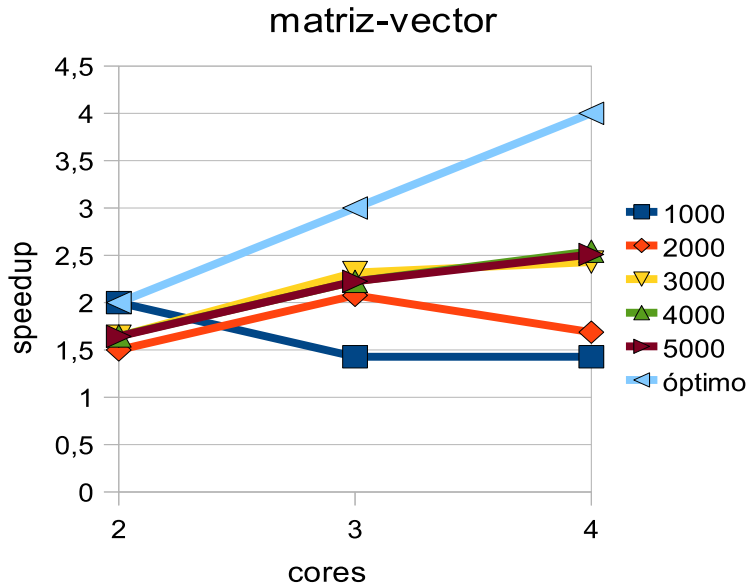
Ejecución

- Seleccionar el número de threads
 - Antes de la ejecución: establecer variable de entorno OMP_NUM_THREADS
 - Durante la ejecución: con función openmp omp_set_numthreads()

Aceleración de la computación



Aceleración de la computación



Caída de prestaciones

- Coste bajo: suma n , matriz-vector n^2
- Gestión de los threads: creación, asignación de recursos, sincronización...
- Conflicto en accesos a memoria
- Aumentan al aumentar el tamaño o el orden
- Usar programación paralela para problemas de alto coste
- la multiplicación de matrices tiene coste n^3 y se logran mejores prestaciones

Caída de prestaciones

- Coste bajo: suma n , matriz-vector n^2
- Gestión de los threads: creación, asignación de recursos, sincronización...
- Conflicto en accesos a memoria
- Aumentan al aumentar el tamaño o el orden
- Usar programación paralela para problemas de alto coste
- la multiplicación de matrices tiene coste n^3 y se logran mejores prestaciones

Librerías

- Optimizadas para distintos tipos de sistemas
- por ejemplo, de álgebra lineal
 - BLAS, LAPACK, multithreaded (memoria compartida)
 - ScaLAPACK para paso de mensajes
 - CCBLAS para GPU

Trabajos de aplicaciones

- Utilización de entornos de programación paralela en algún trabajo de interés para el alumno
 - estudio de interés de utilización del paralelismo
 - análisis del tipo de paralelismo
 - identificación de zonas de código susceptibles de beneficiarse del uso de paralelismo
 - programación y estudio experimental

Trabajos de herramientas

- Análisis de alguna herramienta de computación paralela
 - HWLOC, Portable Hardware Locality, <http://www.open-mpi.org/projects/hwloc/>
 - CUDA, http://www.nvidia.es/object/cuda_home_new_es.html
 - TBB, Threading Building Blocks, <http://www.threadingbuildingblocks.org/>
 - PLASMA, Parallel Linear Algebra for Scalable Multi-core Architectures, <http://icl.cs.utk.edu/plasma/>
 - ForestGOMP, An OpenMP platform for hierarchical architectures, <http://runtime.bordeaux.inria.fr/forestgomp/>
 - UPC, Unified Parallel C, <http://upc.gwu.edu/>
 - OpenCL, The open standard for parallel programming of heterogeneous systems, <http://www.khronos.org/opencl/>
 - otros a proponer por el alumno

Habrá que analizar la herramienta e intentar instalarla y usarla en algún ejemplo sencillo

Gestión de los trabajos

- Selección de aplicación con la que trabajar o propuesta de herramienta por el alumno (correo a domingo@um.es)
- se asignaría profesor y sistema computacional con que trabajar
- resumen de trabajo realizado a entregar antes del 15 de febrero, y revisión del trabajo con el alumno