Actas Simposio-Taller JENUI 2012, Ciudad Real, 10-13 de julio 2012
I.S.B.N. 10: 84-695-3941-8 | I.S.B.N. 13:978-84-695-3941-5
Páginas 49-56

Una experiencia de iniciación al paralelismo en segundo curso del Grado de Ingeniería Informática

Manuel E. Acacio, Javier Cuenca, Lorenzo Fernández, Ricardo Fernández-Pascual
Departamento de Ingeniería y Tecnología de Computadores, Universidad de Murcia, 30071 Murcia
meacacio@ditec.um.es, jcuenca@um.es, lfmaimo@ditec.um.es, r.fernandez@ditec.um.es
Joaquín Cervera, Domingo Giménez
Departamento de Informática y Sistemas, Universidad de Murcia, 30071 Murcia
jcervera@um.es, domingo@um.es
M. Carmen Garrido, Juan A. Sánchez Laguna
Departamento de Ingeniería de la Información y las Comunicaciones, Universidad de Murcia, 30071 Murcia
carmengarrido@um.es, jlaguna@um.es
José Guillén, Juan Alejandro Palomino Benito, María-Eugenia Requena
Centro de Supercomputación, Fundación Parque Científico, 30071 Murcia
{jguillen,jpalomino,mrequena}@parquecientificomurcia.es

Resumen

En este artículo se analiza una experiencia de introducción del paralelismo de forma temprana en el Grado de Ingeniería Informática. En la experiencia participan cuatro asignaturas de segundo curso, impartidas por tres departamentos distintos y con la colaboración de un centro de computación. En este curso se realiza la primera aproximación de los alumnos al paralelismo, y se pretende realizar un acercamiento coordinado y práctico a diferentes materias.

Summary

This work presents an experience of early introduction to parallelism in the Degree on Computing Engineering. Four courses of the second year participate in the experience and also a computing centre. The courses are tought by three departments. In the second year the students are introduced to parallelism for the first time, and with our experience we intend to approach different topics of parallelism in a coordinated and practical way.

Palabras clave

Enseñanza del paralelismo, arquitecturas paralelas, programación paralela, algoritmos paralelos, Grado de Ingeniería Informática.

1. Introducción

La computación paralela es cada día más popular. En la actualidad los sistemas computacionales básicos son paralelos (portátiles y equipos de sobremesa duales y quadcore e incluso hexacores o con *multithreading*, incluyendo además procesadores gráficos programables), y también lo son los clusters de ordenadores y los supercomputadores, que están compuestos de nodos multinúcleo. Esta situación ha motivado varias iniciativas del IEEE Technical Committee on Parallel Processing (TCPP) [3]:

- Propuesta de un curriculum del tipo de ACM con las materias de paralelismo que deberían incluirse en los estudios de informática.
- Realización del workshop EduPar de enseñanza del paralelismo, que celebrará su segunda edición en el IPDPS de 2012.
- Una convocatoria de *Early Adopters* financiada por Intel para experiencias de inclusión de las materias del curriculum del TCPP.

La experiencia que se presenta aquí está financiada dentro de la segunda convocatoria de *Early Adopters* (otoño de 2011), y tiene como objetivo la introducción del paralelismo de forma temprana en el segundo curso del Grado de Ingeniería Informática (GII) en la Universidad de Murcia, estudiando diferentes materias de los distintos apartados del curriculum del TCPP (Arquitectura, Programación, Algoritmos y Temas transversales), con un enfoque que engloba cuatro asignaturas impartidas por tres departamentos, y con la participación del Centro de Supercomputación de la Fundación Parque Científi-

co de Murcia (CSC), que contribuye a dar a la experiencia una orientación más práctica, al permitir la experimentación con sistemas de distintas características, desde portátiles de los alumnos hasta un cluster de cuatro nodos y 32 procesadores del CSC.

El resto del trabajo está organizado en la siguiente forma. En la sección 2 se resume el contexto en que se desarrolla el proyecto, en la sección 3 se muestran sus líneas generales, y su desarrollo se comenta en la sección 4. Finalmente, en la sección 5 se muestran las conclusiones y líneas de trabajo futuro.

2. Contexto

El proyecto tiene por finalidad analizar la iniciación al paralelismo de forma temprana en el GII. Al ser los sistemas computacionales actuales mayoritariamente paralelos, todos los estudiantes deben adquirir conocimientos de diferentes aspectos del paralelismo, por lo que su estudio no debe dejarse exclusivamente para las especializaciones en los últimos cursos. Así, nos planteamos que esta iniciación puede hacerse en segundo curso, después de un curso completo sin tratar ninguna materia de paralelismo.

Las asignaturas de segundo curso en el GII en la Facultad de Informática de la Universidad de Murcia (FI-UM) se muestran en la tabla 1, junto con una breve descripción de su contenido. La estructura completa del GII en la FI-UM se encuentra en [2]. Las asignaturas que participan en este proyecto aparecen resaltadas, y representan el 40 % de los créditos del segundo curso, con dos asignaturas en cada cuatrimestre, y con asignaturas de arquitectura y sistemas, y de programación. Sólo aparecen algunas materias de paralelismo en los descriptores de PCD y AEC, y esos eran los únicos temas de paralelismo que se trataban en segundo. Con nuestra propuesta se incluyen nuevas materias y de todos los apartados del curriculum TCPP: arquitectura (ISO y AEC), programación (ISO, PCD y AED), algoritmos (AED) y temas transversales (ISO, AEC y PCD).

Primer cuatrimestre
Algoritmos y Estructuras de Datos I
Especificaciones algebraicas, Hashing, Estructuras de datos múltiples y duales, Árboles, Grafos, Algoritmos de búsqueda
Autómatas y Lenguajes Formales
Lenguajes formales, Expresiones regulares, Gramáticas, Autómatas
Programación Orientada a Objetos
Modelo de objetos: abstracción, encapsulación y modularidad, Diseño orientado a objetos, Reutilización y mantenimiento, Clases, Objetos, Herencia, Polimorfismo, Ligadura dinámica, Diseño por contrato, Manejo de excepciones, Validación con pruebas unitarias
Introducción a los Sistemas Operativos, ISO
Procesos, Memoria, Ficheros, Entrada/Salida, Guiones shell, Gestión de usuarios, Administración de sistemas de ficheros, Copias de seguridad, Arranque y parada del sistema, Monitorización
Ampliación de Estructura de Computadores, AEC
Análisis de prestaciones, Segmentación, Planificación estática y dinámica de instrucciones, Tratamiento de dependencias de control, Mejora de prestaciones del sistema de memoria
Segundo cuatrimestre
Compiladores
Máquinas virtuales y lenguajes intermedios, Analizadores léxicos, Análisis sintáctico y semántico, Comprobación de tipos, Abstracción, Optimización
Bases de Datos
Sistemas de bases de datos, Bases de datos relacionales, Lenguajes de consulta de bases de datos, Transacciones, Concurrencia, Recuperación de fallos
Redes de Comunicaciones
Modelos de referencia ISO y TCP/IP, Protocolos de las capas de enlace y de red, LANs, WANs, Routing, Enrutamiento en redes, Control de congestión, Programación con sockets TCP/UDP
Algoritmos y Estructuras de Datos II, AED
Análisis de algoritmos, Complejidad, Avance rápido, Backtracking, Ramificación y acotación, Divide y vencerás, Programación dinámica
Programación Concurrente y Distribuida, PCD
Sistemas de programación fuertemente y débilmente acoplados, Paradigmas clásicos de programación en sistemas distribuidos

Tabla 1: Asignaturas, y sus descriptores, en el segundo curso del GII en la FI-UM.

3. Estructura del proyecto

El proyecto abarca a cuatro asignaturas de segundo curso, con implicación de tres departamentos (Departamento de Ingeniería y Tecnología de Computadores, ISO y AEC, Departamento de Ingeniería de

la Información y las Comunicaciones, PCD, y Departamento de Informática y Sistemas, AED) y del CSC, que facilita un uso restringido de algunos de sus equipos para realizar experiencias docentes.

La tabla 2 muestra las materias del curriculum de TCPP que se cubren en el proyecto. La segunda columna contiene las materias del curriculum. Las letras en la tabla tienen el mismo significado que en el curriculum de TCPP: K representa que se conoce la materia, C que forma parte del núcleo de la asignatura y A que se aplica de algún modo. Sólo dos de las asignaturas incluían temas de paralelismo, y con la aplicación del proyecto se tratan nuevas materias. Las letras resaltadas corresponden a los nuevas materias tratadas o a modificaciones en su tratamiento que hacen que se aborden con más profundidad.

3.1. Actividades del proyecto

El proyecto se desarrolla en el curso 2011-2012 y se planifican una serie de actividades (figura 1) para cubrir las materias de la tabla 2. Durante el primer cuatrimestre están involucradas dos asignaturas (ISO y AEC) y se han realizado dos actividades. La web del proyecto contiene el texto de la solicitud al programa de *Early Adopters* y un fichero y material por cada una de las actividades, con una descripción general de la actividad, detallando las materias que se tratan en ella y la forma en que se tratan, así como el material utilizado y los resultados de la evaluación o cómo se planifica realizarla. Tanto la página como los ficheros y el material deberían estar en inglés, pero algún material se incluye sólo en español, al estar originalmente en este idioma. Algunas actividades (figura 1) son particulares de una asignatura (1, 5 y 6), pero otras serán coordinadas entre dos asignaturas (3 y 4) y en otras colabora el CSC (2, 7 y 8). Se pretende realizar una aproximación coordinada a las diferentes materias, y acercar a los estudiantes a sistemas y aplicaciones reales. Comentamos brevemente cada una de las actividades:

- **Act-1**, ISO: Se introduce el concepto de hilo, con ejemplos de su uso (solapamiento de I/O y procesamiento, y mejora de prestaciones utilizando *multithreading*). Se analizan cuestiones de implementación de hilos, y se comparan implementaciones de usuario y del núcleo. Se enumeran algunos problemas de sincronización. En sesiones de laboratorio se analiza el uso de la

TCPP		ISO	AEC	PCD	AED
ARQUITECTURA					
Clases	K			K **C**	
Superescalar	K		C		
SIMD/Vectorial	K		K		
Pipelines	K		C		
Ejecución OoO	N		C		
Multicore	C	**K**			
NUMA	N	**K**			
Organización de caché	C		K		
Atomicidad	N			C	
Impacto jerarquía mem. en soft.	N	**K**	K **A**		**K**
Ciclos por instrucción (CPI)	C		C		
Benchmarks	K		C		
Spec marks	K		C		
Prestación pico	C		K		
MIPS/FLOPS	C		C		**C**
Prestación sostenida	C		K		
PROGRAMACIÓN					
Memoria compartida	A			C **A**	**A**
Memoria distribuida	C			C	**K**
Ciente/Servidor	C			C	
Task/thread spawning	A			A	**K**
SPMD	C			**C**	**A**
Notaciones de mem. compartida	C			K **A**	
Extensión de lenguajes	K			K	
Bibliotecas	C			A	
Notaciones de SPMD	C			**A**	
MPI	C			**C**	
Semántica de tareas y hilos	K	**C**		K **C**	**K**
Sincronización	A			A	
Regiones críticas	A			A	**A**
Productor-consumidor	A			A	**A**
Monitores	K			A	
Deadlocks	C			K	
Modelos de memoria	N			K	
Computación y scheduling	C		C		
Estrategias de descomposición	C		K		
Fusión de bucles	N		A		
Scheduling and mapping	C	**K**	C		
Monitorización de prestaciones	K	**A**			
Métricas de prestaciones	C		C		**A**
Speed-up	C		C	**A**	**A**
Eficiencia	C		C		**A**
Ley de Amdahl	K		C		
ALGORITMOS					
Coste asintótico	C				**C**
Tiempo	C				**C**
Espacio	C				**C**
Speed-up	C				**C**
Nociones de scheduling	C			K	**K**
Divide y Vencerás	C				**A**
Broadcast	C			K	
Asincronismo	K			K	
Sincronización	K			A	
Ordenación	C				**A**
Grafos de búsqueda	C				**K**
Computaciones especializadas	C				**K**
TEMAS TRANSVERSALES					
El porqué y qué es la PPD	K			C	
Concurrencia	K			C	
No determinismo	K			A	
Potencia	K		K		
Localidad	C	**K**	C		
Seguridad en sist. distribuidos	K	**K**			

Tabla 2: Materias del curriculum de TCPP en el segundo curso del GII en la FI-UM.

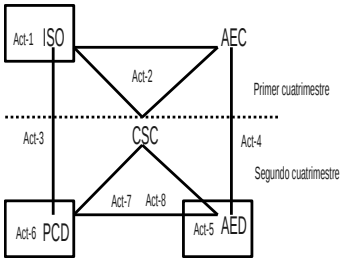

Figura 1: Actividades planificadas en el proyecto.

CPU cuando se arrancan varios hilos, usando algunas herramientas del sistema.

- **Act-2**, ISO+AEC+CSC: Se organiza una visita al CSC, con una presentación del personal del centro sobre aspectos prácticos de un laboratorio de computación paralela. Se analizan en un entorno real algunas materias estudiadas en ISO y AEC y otros de los temas transversales. Se realiza al final del primer cuatrimestre, una vez que las materias se han estudiado en las asignaturas correspondientes.
- **Act-3**, ISO+PCD: En el segundo cuatrimestre, al estudiar en PCD los constructores de memoria compartida y paso de mensajes, se analiza la influencia del uso de varios hilos o procesos en las prestaciones de los núcleos. Se realiza en colaboración con profesores de ISO y aplicando herramientas y conceptos estudiados en el primer cuatrimestre.
- **Act-4**, AEC+AED: En el segundo cuatrimestre, después de estudiar análisis de algoritmos secuenciales en AED, se analiza de forma práctica la influencia de la jerarquía de memoria en el tiempo de ejecución. Se realiza en colaboración con profesores de AEC y utilizando conceptos estudiados en el primer cuatrimestre.
- **Act-5**, AED: Se organizan seminarios de aspectos algorítmicos del paralelismo en temas de AED: medidas de prestaciones paralelas, después de análisis de algoritmos; y esquemas paralelos de divide y vencerás, programación dinámica y *backtracking* después de estudiar los esquemas secuenciales.
- **Act-6**, PCD: Se introducen los constructores básicos de memoria compartida (OpenMP) y paso de mensajes (MPI), que se usan en Act-7 y Act-8.
- **Act-7**, PCD+AED+CSC: Se utilizan los constructores de memoria compartida estudiados en PCD para implementar alguno de los algoritmos paralelos introducidos en AED. Se realizan experimentos en el CSC, con la herramienta Mooshak [1], utilizada en prácticas de AED en los dos cuatrimestres y que se ha adaptado para concursos paralelos y se ha instalado en el CSC para usar un cluster de 4 nodos, cada uno con 8 núcleos. La herramienta se utiliza en el Concurso Español de Programación Paralela [4], y proporciona la aceleración obtenida con programas OpenMP y MPI.
- **Act-8**, PCD+AED+CSC: Se utilizan los constructores de paso de mensajes estudiados en PCD para implementar algunos de los algoritmos paralelos de AED. Los experimentos se realizan en el mismo cluster que en la actividad anterior.

3.2. Resultados esperados

Con la realización de estas actividades se espera:

- Reforzar la iniciación en conceptos de computación paralela a estudiantes del GII, de forma temprana y dentro de asignaturas obligatorias, de manera que estos conceptos lleguen a todos los estudiantes de informática independientemente de su especialización. Dado que actualmente los sistemas computacionales son paralelos (y previsiblementee en el futuro todavía más), creemos que todos los alumnos delGII deben adquirir algunos conocimientos mínimos de cada uno de los apartados del curriculum de TCPP. Además, las materias se introducen de una forma coordinada y analizando aspectos prácticos de sistemas y aplicaciones paralelas.
- Al tiempo que se inicia a los alumnos en el paralelismo, se pueden incorporar a la experiencia profesores que no utilizaran paralelismo con anterioridad. Esto propiciará un mejor entendimiento del paralelismo y de su importancia, y consecuentemente una mejor utilización de él, ya sea en cursos de computación paralela o en otros donde el paralelismo puede ser de utilidad.
- La colaboración en esta experiencia de profesores de distintos departamentos y que están involucrados en cursos de mayor nivel en distintas

especialidades puede propiciar la realización de experiencias conjuntas en las especialidades o en estudios de máster.
- Para cada actividad se generan materiales (documentos, ejemplos, prácticas...) que se puedan utilizar en otras experiencias en la Universidad de Murcia o en otros centros. Este material se incluye en la web del proyecto [5].

3.3. Evaluación de la experiencia

Al ser una experiencia pionera, se hace necesaria su evaluación para decidir si seguir con ella y si es conveniente reorientarla. Al final del curso 2011-2012 se analizará con qué profundidad se ha tratado cada materia y la utilidad de cada actividad. Los estudiantes y profesores participantes en la experiencia darán su opinión (subjetiva) sobre la cobertura de cada materia y la utilidad de cada actividad. Para cada actividad se realizará una evaluación a los alumnos para comprobar los conocimientos adquiridos sobre cada una de las materias tratadas en ella. En algunos casos la evaluación tendrá peso en la nota final de la asignatura; en otros, las actividades son complementarias, sin influencia en la nota final.

4. Desarrollo de la experiencia

Detallamos en distintas subsecciones cómo se trata cada una de las materias. Para cada materia se indica la asignatura, el número aproximado de horas y la profundidad en que se trata.

4.1. Arquitectura

La mayoría de las materias sobre arquitectura se trataban previamente, principalmente en AEC y algunas en PCD. Con el proyecto se estudian nuevas materias, y otras tratadas previamente se estudian en más profundidad y en coordinación con otras asignaturas, así como con una aproximación práctica.

Clases de arquitecturas (PCD 0.5 C): Se hace una descripción de la ejecución de programas concurrentes en diferentes arquitecturas, considerando sistemas fuerte y débilmente acoplados.

Superscalar (ILP) (AEC 0.5 C): Se considera la ejecución de múltiples instrucciones por ciclo en procesadores segmentados in-order y out-of-order y VLIW.

SIMD/Vector (AEC 0.5 K): Se introducen estos conceptos, pero no se estudian en profundidad y se tratarán en cursos posteriores.

Segmentación (AEC 6 C): Se dedica aproximadamente una tercera parte del curso al estudio de la segmentación y del diseño de procesadores RISC segmentados. La explicación incluye *forwarding*, predicción de saltos e implementación de excepciones.

Ejecución OoO (AEC 4 C): La ejecución fuera de orden se explica a través del algoritmo de Tomasulo. Se estudian los conceptos de especulación y técnicas más avanzadas de predicción de saltos.

Multinúcleo (ISO 1 K): Se describe la arquitectura multinúcleo, y en Act-1 y Act-3 se ve la utilidad del *multithreading*.

NUMA (ISO 0.5 K): Se explica brevemente el concepto de NUMA, y en Act-1, Act-2 y Act-3 se ve la importancia de la localidad de datos en la programación *multithreading*.

Organización de la *cache* (AEC 6 K): Se dedica aproximadamente el 25 % del curso al estudio de diferentes formas de analizar y mejorar las prestaciones de la jerarquía de memorias: *way prediction*, *prefetching*, *caches* multinivel, *early-restart* y *critical-word-first*, *write-merging*, *caches* virtuales, *caches* con etiquetado físico e indexado virtual, etc.

Atomicidad (PCD 2 C): Se introduce el concepto de atomicidad tanto de software como de hardware. Debido a su importancia se introduce en PCD y es básico para Act-3, Act-6, Act-7 y Act-8.

Impacto de la jerarquía de memoria en el software (ISO 0.5 K + AEC 2 A + AED 1 K): En ISO se explica la gestión de memoria por el sistema operativo, y se describen los diferentes niveles en la jerarquía y el movimiento de datos entre ellos. Las ideas básicas del impacto de la jerarquía de memoria en el software se analizan en AEC. En AED se dedica una sesión de laboratorio (Act-4) a la aplicación de las ideas estudiadas en AEC para explicar las diferencias en el tiempo de ejecución de distintas versiones de la multiplicación de matrices.

Ciclos por instrucción (AEC 0.5 C): Este concepto se menciona en un curso anterior, y se explica con más profundidad en AEC.

Benchmarks (AEC 0.5 C), **Spec mark** (AEC 0.5 C): Se explica el concepto de *benchmark* y sus principales características,y se discuten brevemente varios grupos de *benchmarks*.

Prestación pico (AEC 0.5 K): Este concepto se introduce junto con otros conceptos relativos a la medida de prestaciones de sistemas computacionales.

MIPS/FLOPS (AEC 0.5 C + AED 0.5 C): La definición de MIPS y FLOPS se da en AEC, y en AED se calculan para implementaciones de varios algoritmos, en las sesiones de laboratorio Act-4, Act-7 y Act-8.

Rendimiento sostenido (AEC 0.5 K): Se introduce junto con otros conceptos relativos a la medida de prestaciones de sistemas computacionales.

4.2. Programación

Igual que con las materias de arquitectura, la mayoría de las materias de programación se trataban previamente, principalmente en PCD. Dentro de este proyecto se estudian nuevas materias y las ya estudiadas se tratan con mayor profundidad, incluyendo la realización de prácticas con OpenMP y MPI en un cluster del CSC.

Memoria compartida (PCD 10 A + AED 2 A), **Memoria distribuida** (PCD 10 C + AED 2 K): En PCD se estudian los conceptos básicos de programación en memoria compartida y por paso de mensajes, y en Act-5 se presentan algunos esquemas algorítmicos de memoria compartida y paso de mensajes. Se realizan algunos experimentos con programas básicos de memoria compartida (Act-7) y paso de mensajes (Act-8), y se analiza cómo funcionan los mecanismos básicos de programas de memoria compartida (generación de hilos, sincronización, reducción del tiempo de ejecución...) y paso de mensajes (puesta en marcha de procesos, comunicación, *broadcast*, barreras, reducción del tiempo de ejecución...)

Cliente-servidor (PCD 0.5 C): Se introduce el paradigma a través de diferentes esquemas de paso de mensajes.

Generación de tareas e hilos (PCD 2 A + AED 0.5 K), **SPMD** (PCD 1 C + AED 1 A): Estos conceptos se estudian en PCD, y se analizan para los programas básicos usados en Act-7 y Act-8.

Notaciones de memoria compartida (PCD 10 A): Se estudian diferentes formas de trabajar con memoria compartida. Estas notaciones se estudiaban previamente en PCD, y sirven como conceptos básicos en Act-6 y Act-7.

Extensiones de lenguajes (PCD 1 K): Se introducen algunas extensiones a lenguajes de programación para el manejo de tareas e hilos.

Bibliotecas (PCD 10 A): Se estudian las diferentes clases Java que permiten el diseño de programas concurrentes, y se usan para diseñar programas en Act-3.

Notaciones SPMD (PCD 3 A), **MPI** (PCD 3 C): En Act-6 se introducen algunas notaciones básicas de SPMD en OpenMP y paso de mensajes, y se analizan con diferentes ejemplos en Act-7 (OpenMP) y Act-8 (MPI).

Tareas e hilos (ISO 2 C + PCD 3 C + AED 0.5 K): En ISO se introduce el concepto de hilo y se comparan implementaciones de usuario y del núcleo. En Act-7 y Act-8 se analiza la gestión de tareas e hilos en algunos programas.

Sincronización (PCD 2 A): Se introduce para ilustrar la necesidad de ciertas reglas entre tareas e hilos diferentes.

Regiones críticas (PCD 2 A + AED 0.5 A): Se definen en PCD, y en AED se usan en algunos esquemas algorítmicos paralelos.

Productor-consumidor (PCD 1 A + AED 0.5 A): Se introduce el esquema en PCD, y se usa en Act-7 y Act-8.

Monitores (PCD 4 A): Se introduce la idea, las características, y las ventajas sobre otras primitivas tales como semáforos y variables condicionales.

Abrazo mortal (PCD 0.5 K): Se describen diferentes escenarios de mal funcionamiento de programas debido a abrazo mortal.

Modelos de memoria (PCD 0.5 K): Se introducen las características de diferentes modelos de memoria.

Planificación y computación (AEC 2 C): Se explica la planificación estática, y los estudiantes serán capaces de planificar fragmentos cortos de código evitando paradas. En sesiones de laboratorio tienen que optimizar código ensamblador generado con un compilador sin optimización, planificando las instrucciones para evitar paradas en un procesador segmentado simulado.

Estrategias de descomposición de la computación (AEC 1 K): Se explica el desenrrollado de bucles y se menciona la idea de ejecutar grupos independientes de iteraciones de un bucle en procesadores diferentes.

Fusión de bucles (AEC 0.5 A): Se menciona como una de las transformaciones para reducir la tasa de fallos de las *caches*.

Planificación y asignación (ISO 2 K + AEC 1 C): En Act-1 de ISO se muestra la importancia de la asignación de tareas entre núcleos diferentes de la CPU. La planificación de instrucciones se discute en AEC.

Monitorización de prestaciones (ISO 2 A): En Act-1 se utilizan herramientas del sistema y *profilers* para analizar el uso de la CPU cuando se inician varios hilos.

Métricas de prestaciones (AEC 1 C + AED 0.5 A): En AEC se dan algunas métricas de prestaciones de programas paralelos, y se usan en AED en teoría (Act-5) y experimentalmente (Act-7 y Act-8).

Aceleración (AEC 1 C + PCD 0.5 A + AED 0.5 A), **Eficiencia** (AEC 0.5 C + AED 0.5 A): Se definen en AEC y se calculan para algunos esquemas en Act-5 y para algunos programas en Act-7 y Act-8.

Ley de Amdahl (AEC 1 C): Se utiliza para evaluar la influencia sobre el rendimiento de algunas de las técnicas explicadas.

4.3. Algoritmos

Previamente se trataban únicamente algunos algoritmos paralelos básicos en PCD. En este proyecto se introducen más algoritmos paralelos, lo que se hace de forma práctica con la colaboración de PCD y AED y el uso de un cluster del CSC.

Costes asintóticos de computación (AED 0.5 C): Cuando se estudian las notaciones asintóticas de complejidad de algoritmos secuenciales se introducen también las correspondientes notaciones paralelas.

Tiempo (AED 0.5 C), **Espacio** (AED 0.5 C): En Act-5 se estudian el tiempo de ejecución y la necesidad de memoria de algunos algoritmos paralelos.

Aceleración (AED 0.5 C): En Act-5 se estudia el *speed-up* asintótico de algunos algoritmos paralelos, y se analiza experimentalmente en Act-7 y Act-8.

Nociones de planificación (PCD 0.5 K + AED 0.5 K): Las nociones básicas de planificación se introducen en PCD, y se consideran en algunos de los esquemas en Act-5, y en la práctica en Act-7 y Act-8.

Divide y vencerás (AED 0.5 A): En Act-5 se presentan esquemas paralelos divide y vencerás, y se usan en algunos programas (por ejemplo mergesort) en Act-7 y Act-8.

Broadcast (PCD 0.5 K): Se muestran diferentes modelos de comunicación y entre ellos se describe el *broadcast*.

Asincronismo (PCD 1 K), **Sincronización** (PCD 1 A): Al estudiar distintos modelos de comunicación se analizan las ventajas y desventajas del paso de mensajes síncrono y asíncrono. Estos conceptos se usan en Act-8.

Ordenación (AED 0.5 A): Un programa paralelo de ordenación (además de la ordenación por mezcla) se analiza en Act-7 y Act-8.

Algoritmos sobre grafos (AED 0.5 K): En Act-5 se analiza un algoritmo paralelo de búsqueda en grafo, que se usa en Act-7 y Act-8.

Computación especializada (AED 1 K): Algunos algoritmos de multiplicación de matrices se usan en Act-4 para estudiar la influencia de la jerarquía de memoria en el tiempo de ejecución, un algoritmo de Strassen paralelo en Act-5, y en Act-7 y Act-8 para analizar algunos aspectos del paralelismo.

4.4. Temas transversales

Se amplía la introducción de materias del apartado de *cross-cutting*, y colabora el CSC con una presentación y una visita al Centro, donde se presenta la implantación en un centro de computación de algunos de los temas introducidos en las asignaturas.

El porqué y la descripción de la computación paralela y distribuida (PCD 1 C): La diferencia entre computación paralela y distribuida se discute al principio del curso.

Concurrencia (PCD 1 C): Es una de las primeras materias del curso, y es fundamental para entender por qué son necesarios mecanismos tales como semáforos, monitores y paso de mensajes.

No determinismo (PCD 1 A): Se describe el concepto, qué lo causa y las consecuencias de no evitarlo.

Potencia (AEC 0.5 K): Se explica la importancia del consumo de energía y se menciona el efecto que algunas técnicas de arquitectura tienen en él.

Localidad (ISO 0.5 K + AEC 1 C): La localidad espacial y temporal se introduce en un curso previo, y en este se explican más en profundidad, con estudio de optimizaciones de programas que mejoran la localidad. En ISO se describe la influencia de la localidad de datos en las prestaciones del software.

Seguridad en sistemas distribuidos (ISO 1 K): Se describe cómo proteger la información de accesos no autorizados y de destrucción o alteración ma-

lintencionada.

4.5. Evaluación

La experiencia se inició en el primer cuatrimestre del curso 2011-2012, en el que estaban planificadas únicamente las dos primeras actividades. En la primera de ellas los conocimientos adquiridos por los alumnos se evaluaron con preguntas en un examen teórico, con una parte de test (Multinúcleo, NUMA, Impacto de la jerarquía de memoria en el software, Semántica de tareas e hilos, Localidad, Seguridad en sistemas distribuidos) y otra de ejercicios teórico-prácticos (Planificación y mapeo), y con prácticas de laboratorio (Monitorización de prestaciones). La evaluación de la actividad se ha realizado de manera informal. Al final de cada sesión teórica se establece un debate con los alumnos sobre las distintas materias, para que comprueben si han captado los temas tratados. En las sesiones de laboratorio se supervisa el trabajo de los alumnos mientras resuelven los ejercicios, y se establece un debate sobre las materias tratadas al final de la sesión. Teniendo en cuenta la interacción con los alumnos en los debates citados y los resultados obtenidos en las pruebas escritas, consideramos que todas las materias se han tratado de manera satisfactoria. La segunda actividad ha consistido en una visita al CSC, con asistencia voluntaria. Esta actividad también ha tenido buena aceptación entre los alumnos, que participaron en buen número y plantearon diversas cuestiones al personal del CSC con las que aclarar algunos de los temas tratados en teoría. La evaluación de las actividades del segundo cuatrimestre se está llevando a cabo en la actualidad. Se realiza con participación en laboratorios y con una parte opcional de paralelismo en las entregas de prácticas. La participación de los alumnos empezó siendo satisfactoria, pero la entrega de la parte de paralelismo en las prácticas está previsto que se reduzca, debido a la coincidencia con el final del curso y con las entregas de prácticas de otras asignaturas.

5. Conclusiones y trabajo futuro

Se presenta una experiencia de iniciación al paralelismo en segundo curso del GII. La experiencia está dentro del programa *Early Adopters* del IEEE TCPP patrocinado por Intel. Las idea general del proyecto es realizar una aproximación coordinada entre distintos departamentos y en distintas asignaturas, con una aproximación práctica en la que colabora el CSC (lo que se observa en la Sección 4, donde se indica para cada materia las asignaturas implicadas y la particiapción del CSC), e introduciendo nociones de los distintos aspectos del paralelismo del curriculum del TCPP (arquitectura, programación, algoritmos y temas transversales) a todos los alumnos del grado al llevarse a cabo en asignaturas obligatorias de segundo curso. Se han incorporado al proyecto profesores que no tenían experiencia previa con paralelismo. En este curso han participado en la experiencia dos de los tres grupos de segundo, pues en uno de ellos los profesores en el proyecto no impartían el total de las asignaturas. Para cursos sucesivos intentaremos implicar a más profesores para extenderlo a todos los grupos. El material generado se encuentra en la página web [5], y puede ser consultado para considerar su utilización en otros cursos o centros.

A partir de este proyecto se está trabajando en otro que afecta a asignaturas optativas de paralelismo en los últimos cursos, con participación de grupos de seis universidades, y se pretende analizar las materias a incluir en las especialidades del GII.

Agradecimientos

Subvencionado por la Fundación Séneca de la Región de Murcia, 08763/PI/08.

Referencias

[1] Mooshak, system for managing programming contests on the web. http://mooshak.dcc.fc.up.pt/~zp/mooshak/.

[2] Grado de Ingeniería Informática, Universidad de Murcia. http://www.um.es/informatica.

[3] IEEE Technical Committee on Parallel Processing. http://www.cs.gsu.edu/~tcpp/curriculum/index.php.

[4] Concurso Español de Programación Paralela. http://cpp.fpcmur.es.

[5] Proyecto *EarlyAdopter* en la Universidad de Murcia. http://www.um.es/earlyadopters.