

## ENSEÑANZA DE LA PROGRAMACION PARALELA EN LA UNIVERSIDAD POLITECNICA DE VALENCIA.

J.L. Pérez<sup>1</sup>, A.M. Vidal<sup>1</sup>, V. Hernández<sup>1</sup>

<sup>1</sup>*Facultad de Informática, Universidad Politécnica de Valencia*  
*e-mail: [jlperez,avidal,vhernand.@dsic.upv.es](mailto:jlperez,avidal,vhernand.@dsic.upv.es)*

**RESUMEN:** En este artículo se presenta una visión de la enseñanza de las técnicas de programación paralela en la Escuela Universitaria de Informática y la Facultad de Informática de la Universidad de Politécnica de Valencia (UPV). Dicha visión se centra principalmente en la descripción de la asignatura, Lenguajes y Entornos de la Programación Paralela, impartida en el último curso de la titulación Ingeniero en Informática. En dicha asignatura se pretende que el alumno adquiera un conocimiento avanzado en el manejo de las técnicas de programación paralela utilizado el paradigma de paso de mensajes.

### 1.- INTRODUCCIÓN.

La posibilidad de ejecutar eficientemente aplicaciones paralelas en agrupaciones de PC's, con un coste muy reducido, unido a la existencia de múltiples estándares de programación paralela, aplicables a todo tipo de plataformas, hace asequible la utilización de técnicas de programación paralela en aplicaciones con un elevado coste computacional. Es por esta razón, y por la creciente demanda de este tipo de conocimientos por parte del mundo laboral, que consideramos fundamental la enseñanza de dichas técnicas dentro de la formación del futuro Ingeniero en Informática.

En las siguientes secciones se pretende esbozar la enseñanza de las asignaturas que describen las técnicas de Programación Paralela en la Escuela y la Facultad de Informática de la Universidad Politécnica de Valencia durante los últimos años. El estudio se centrará en la asignatura de quinto curso de la Facultad de Informática, Lenguajes y Entornos de la Programación Paralela. Dicha asignatura fue creada hace 3 años dentro del marco de la reforma del Plan de Estudios de la Facultad de Informática efectuada en 1994, y pretende hacer frente a los retos que esta planteaba.

## 2.-DESCRIPCIÓN DE LA ASIGNATURA LENGUAJES Y ENTORNOS DE LA PROGRAMACIÓN PARALELA

La asignatura de Lenguajes y Entornos de la Programación Paralela, tal y como se ha comentado, se ha venido impartiendo en el quinto curso de la titulación de Ingeniero Informático. El alumno de esta asignatura previamente ha cursado la asignatura, Arquitectura de Computadores, troncal de cuarto curso, donde se le han presentado los conceptos básicos de las arquitecturas paralela. También suele haber cursado la asignatura, Computación Paralela, como asignatura optativa de tercer curso, donde se aprenden los conceptos fundamentales de la computación en paralelo y de la programación paralela orientada a *paso de mensajes*.

### a) Coordenadas principales

La posición de la asignatura en el último semestre del Plan de Estudio es un factor determinante de la organización que proponemos para ella. Tiene un carácter final muy acusado; es decir, el alumno que la elige completa con ella su formación en esta disciplina. Debe tenerse en cuenta que la asignatura coexiste, a veces, con el desarrollo del proyecto final de carrera, y que los alumnos están pensando ya en su próxima orientación profesional y en aplicar los conceptos adquiridos en el ejercicio de su profesión.

Todo ello nos lleva a proponer una organización de la asignatura que puede considerarse atípica respecto a la propuesta de otras asignaturas. En primer lugar, la asignatura debe tener un carácter muy práctico, imprescindible en una asignatura en la que priman aspectos de programación. Además, proponemos organizar la asignatura en distintos Casos de Estudio, elegidos de forma que ayuden al alumno a alcanzar los objetivos previstos, evitando en lo posible lecciones magistrales y provocando con ello una implicación real del alumno en el proceso de aprendizaje.

Es también importante destacar que la diversificación de especialidades en el último curso hace que el número de alumnos de cursan esta asignatura sea reducido (entre 20 y 30). Esto permite que la implantación de esta metodología no suponga un excesivo consumo de recursos, tanto a nivel humano, como de recursos físicos.

### b) Objetivos

De acuerdo con la filosofía de la asignatura, expresada en el subapartado anterior, se plantean los siguientes *objetivos generales*:

- Ser capaz de diseñar de forma eficiente una aplicación paralela.
- Adquirir habilidad en el manejo avanzado de, al menos, dos entornos de programación paralela.
- Ser capaz de evaluar el comportamiento, tanto a nivel teórico, como a nivel práctico, de los algoritmos implementados.

Los objetivos descritos pueden refinarse dando lugar a los siguientes *objetivos específicos*:

- O1 Conocer los entornos de programación paralela más típicos.
- O2 Ser capaz de determinar los parámetros básicos de una máquina paralela.
- O3 Ser capaz de implementar rutinas paralelas sencillas.
- O4 Ser capaz de implementar algoritmos paralelos robustos, portables, legibles y reutilizables.
- O5 Saber convertir el planteamiento secuencial de un problema a su equivalente paralelo sin modificar su estructura sustancialmente.
- O6 Adquirir habilidad en el manejo avanzado de dos entornos de programación paralela.
- O7 Ser capaz de evaluar y comparar los algoritmos paralelos implementados, así como los entornos de programación utilizados.
- O8 Ser capaz de implementar esquemas algorítmicos típicos utilizando el entorno de programación paralelo adecuado.

### **c) Metodología**

La asignatura se divide una serie de *Casos de Estudio* que el alumno debe resolver, individualmente, como si fueran pequeños proyectos. Estos Casos de Estudio se eligen convenientemente, de forma que su desarrollo exija al alumno alcanzar los objetivos previstos. Cada Caso de Estudio se desarrolla en tres etapas

*Información Teórica.* En esta etapa se le proporciona al alumno la información necesaria para entender el problema que se aborda en cada Caso de Estudio, así como las especificaciones requeridas para su resolución. Esta información se suele presentar en sesiones teóricas. Como resultado de esta etapa el alumno debe disponer de un modelo teórico del problema a resolver.

*Diseño e Implementación.* A partir de la solución teórica al problema obtenida en la etapa anterior, el alumno debe escoger la implementación que considere más eficiente, utilizando el entorno de programación que considere más adecuado. Así mismo, debe realizar todas aquellas pruebas experimentales que le permitan validar y evaluar su implementación.

*Memoria.* Por último, basándose en el trabajo realizado, el alumno confeccionará una memoria que describa el marco teórico, la aproximación algorítmica, los resultados y las conclusiones obtenidas.

Utilizando esta metodología se pretende obtener las siguientes ventajas:

*Aprendizaje orientado a la resolución de problemas.* El alumno se enfrenta a una serie de problemas prácticos (problemas tipo), similares a los que se presentan en la mayoría de aplicaciones reales.

*Carácter práctico de la asignatura.* El alumno piensa en problemas y no en contenidos. Desaparece la separación entre teoría y práctica; todo contenido teórico tiene sentido dentro de su ámbito de aplicación.

*Estimular la responsabilidad del alumno en el aprendizaje.* El alumno es el protagonista de las soluciones informáticas que obtiene. Se siente responsable ya que es él quien decide la estrategia a utilizar para resolver un problema determinado.

#### d) Casos de Estudio propuestos

Los casos de estudio propuestos (ver tabla 1) tiene como finalidad la consecución, de forma gradual, de los objetivos propuestos en la asignatura. Generalmente, los casos de estudio guardan relación con problemas resueltos en otras asignaturas utilizando un paradigma de programación secuencial, lo cual evita que el alumno deba prestar atención a detalles ajenos a la implementación. Además, existe la posibilidad de modificar el contenido de los Casos de Estudio atendiendo al perfil de los alumnos matriculados en un determinado año.

Caso	Título	Objetivos
CS1	El entorno PVM: Operaciones elementales y comunicaciones.	O1-3
CS2	Distribuciones de datos en entornos de programación paralela. Producto Matriz-Vector	O3-5
CS3	Algoritmo paralelos para el cálculo de las distancias mínimas en un grafo dirigido.	O3-5 y O8
CS4	El entorno MPI: Producto Matriz-Matriz	O4-6
CS5	Algoritmos paralelos para la resolución de Sistemas Triangulares.	O4-7
CS6	Algoritmo paralelos de tipo Divide y Vencerás.	O4-8

**Tabla 1:** Casos de Estudio de la asignatura de Lenguajes y Entornos de la Programación Paralela (curso 99/00) y relación con los objetivos.

La organización de la asignatura requiere una constante realimentación con el alumno con el fin de poder establecer el ritmo adecuado que permita, por una parte, la consecución de los objetivos, y por otra respecto a la filosofía del método que garantiza al alumno libertad para ir organizando su trabajo.

El contenido de los casos de estudio es flexible, ya que es susceptible de ser alterado en cursos sucesivos sin que se resientan los objetivos de la asignatura. Lo mismo sucede con los entornos de programación utilizados (en nuestro caso PVM y MPI), ya que podrían ser substituidos por

otros entornos de programación orientados a *paso de mensajes* sin que el contenido de los Casos de Estudio se alterase significativamente.

### **3.-EVALUACIÓN DE LA ASIGNATURA LENGUAJES Y ENTORNOS DE LA PROGRAMACIÓN PARALELA.**

En este apartado se muestra una evaluación crítica de la asignatura, Lenguajes y Entornos de la Programación Paralela, durante los tres años que esta ha sido impartida en la Facultad de Informática. En primer lugar se describirán las deficiencias observadas durante el transcurso de los dos primeros años de impartición. Posteriormente se describirán las modificaciones introducidas en el presente curso (tercer año) con el fin de mejorar la asignatura y las mejoras que se han percibido una vez finalizado este. Finalmente, se muestra un estudio realizado sobre los alumnos de la asignatura.

#### **a) Deficiencias observadas durante los 2 primeros cursos**

*Excesiva complejidad en la implementación.* Algunos casos de estudio resultaron tener una complejidad excesivamente elevada para algunos de los alumnos, debido a carencias en la estrategia de diseño de los algoritmos paralelos y a la incapacidad de hacer una separación entre las diferentes partes que constituyen la aplicación paralela. Como resultado, los algoritmos paralelos implementados eran monolíticos y difíciles de depurar.

*Poca adecuación del hardware.* Las prácticas se realizaban en un laboratorio, que constaban de 20 PC's. Las máquinas de este laboratorio se hallan interconectadas mediante una red Ethernet junto con el resto de máquinas de la UPV. Este hecho afecta significativamente al coste de comunicaciones de los algoritmos paralelos implementados e impide una evaluación efectiva de los mismos.

*Escasez de material y referencias bibliográficas.* También se observó, en los dos primeros años de impartición de la asignatura, una falta de referencias adecuadas a las necesidades de los alumnos. Los manuales estándares de PVM y MPI son excesivamente poco amigables para usuarios no avanzados en su manejo, y por tanto, inadecuados para su uso docente.

#### **b) Modificaciones introducidas en el tercer curso**

*Simplificación y reordenación de los casos de estudio.* Se ha intercambiado el orden de algunos casos de estudio con el fin de alcanzar los objetivos de una forma más gradual. Se ha añadido el Caso de Estudio dedicado a la implementación de las distribuciones de datos más comunes (CS2) con el fin de facilitar el diseño de los siguientes Casos de Estudio. Además, en los primeros Casos de Estudio, se han proporcionado a los alumnos pequeñas librerías de funciones para la realización de determinadas tareas.

*Utilización de una metodología de diseño de los algoritmos paralelos.* Se ha enseñado la necesidad de estructurar adecuadamente las aplicaciones paralelas con el fin de facilitar la etapa de diseño e implementación. Se ha incidido especialmente en la conveniencia de encapsular cada una de las partes de la aplicación paralela, lo que permitirá la reutilización de gran parte del código de un Caso de Estudio en futuros Casos de Estudio.

*Disponibilidad de un Agrupación de PC's en modo dedicado.* Se adquirió una máquina paralela de coste reducido, que constaba de 5 PC's interconectados con una red Fast Ethernet aislada del resto de ordenadores de la UPV. Con esta máquina se espera que los alumnos puedan realizar una evaluación realista de las prestaciones de los algoritmos paralelos

*Libros de apuntes y bibliografía en Internet.* Se han proporcionado boletines explicativos para cada caso de estudio, junto con libros de apuntes destinados a la explicación de los entornos PVM y MPI. Además, se han proporcionado direcciones de Internet donde obtener información adicional para la elaboración de algunos de los Casos de Estudio.

### c) Mejoras observadas durante el transcurso del tercer curso

*Diseño de los algoritmos paralelos.* El hecho de proporcionar funciones ya implementadas supone que el alumno realiza más rápidamente los primeros Casos de Estudio, lo cual le proporciona una motivación positiva, e impide, que alumnos con una base pobre en conocimientos de paralelismo abandonen la asignatura prematuramente. Además, el alumno comprende los beneficios que supone el uso de la programación estructurada y modular en el ámbito de la programación paralela, y la posibilidad, utilizando esta técnica, de reutilizar gran parte del código implementado.

*Evaluación de los algoritmos paralelos.* La posibilidad de disponer de una máquina paralela, pese a su reducido coste, de uso exclusivo para los alumnos de la asignatura, les ha permitido evaluar de forma cómoda los algoritmos paralelos implementados. Al tener la posibilidad de contrastar las prestaciones reales de los algoritmos paralelos con las teóricas, el alumno ha prestado especial atención al análisis a priori del coste de los algoritmos implementados.

### d) Evaluación de la asignatura según los alumnos

En la Tabla 2 se muestra el resultado de una encuesta realizada por los alumnos que en cierta forma corrobora las mejoras efectuadas en el curso 99/00. Los resultados son meramente orientativos debido a que únicamente fueron contestadas por la mitad de los alumnos matriculados.

Pregunta		Cont. por curso (0-10)		
		97/98	98/99	99/00
Interés	Interés general por la asignatura	7,4	7,6	8
	Utilidad futura de la asignatura	6	7,2	7,5
Dificultad	La asignatura se sigue con comodidad	5,9	6,25	7,7
	% de trabajo realizado en clase	50%	50%	63%
Asimilac.	Grado de asimilación de la asignatura	7,8	8,1	8,3
	Evaluación global de la asignatura	6,8	7,7	8,1
	Media de todas las preguntas	6,4	6,6	7,7

**Tabla 2:** Resultados de la evaluación de Lenguajes y Entornos de la Programación Paralela por los alumnos de la asignatura.

#### **4.-CONCLUSIONES**

La asignatura de Lenguajes y Entornos de la Programación Paralela pretende que los alumnos con la titulación de Ingeniero en Informática, tengan la posibilidad de tener un conocimiento avanzado en el manejo de las técnicas de programación paralela utilizando librerías de paso de mensajes. Podemos afirmar, tras la experiencia de los últimos tres años impartiendo esta asignatura, que la metodología utilizada permite conseguir los objetivos establecidos inicialmente. La consecución de estos objetivos no supone un esfuerzo excesivo al alumno de último curso de carrera. Por otra parte, la disponibilidad una máquina paralela muy económica, formada por una agrupación de solo cinco PC's, ha demostrado ser suficiente para permitir a los alumnos observar los beneficios prácticos de la programación en paralelo.

## 5.-BIBLIOGRAFÍA

- [1] A.M. Vidal y J.L. Pérez, Introducción a la Programación en MPI. Ed. Servicio de Publicaciones de la Universidad Politécnica de Valencia. Dep. Legal: V-1280-2000, 2000.
- [2] V.Kumar. Introduction to Parallel Computing: Design and Analysis of Algorithms. The Benjamin/Cummings Publishing Com. Inc. 1994
- [3] A. Geist, PVM: Parallel Virtual Machine. The MIT Press, 1994
- [4] I.Foster. Designing and Building Parallel Programs, (Online), Addison-Wesley, <http://www-unix.mcs.anl.gov/dbpp/>