

XV JENUI. Barcelona, 8-10 de julio de 2009

ISBN: 978-84-692-2758-9

<http://jenui2009.fib.upc.edu/>

## Sobre el papel de la programación paralela en los nuevos planes de estudios de informática

Francisco Almeida  
Departamento de  
Estadística, I. O.  
y Computación  
Univ. La Laguna  
Camino de la Rúa, 57  
38208 La Laguna  
falmeida@ull.es

Domingo Giménez  
Departamento de  
Informática  
y Sistemas  
Univ. Murcia  
Campus de Espinardo  
30071 Murcia  
domingo@um.es

José Miguel Mantas  
Departamento de  
Lenguajes y Sistemas  
Informáticos  
Univ. Granada  
Daniel Saucedo s/n  
18071 Granada  
jmmantas@ugr.es

Antonio M. Vidal  
Departamento de  
Sistemas Informáticos  
y Computación  
Univ. Pol. Valencia  
Camino de Vera s/n  
46022 Valencia  
avidal@dsic.upv.es

### Resumen

La difusión reciente de los sistemas paralelos hace que la programación paralela haya adquirido gran importancia, y dentro del proceso de reforma de los planes de estudios deberíamos plantearnos darle la importancia que realmente tiene. Los ingenieros en informática trabajan con estos sistemas y tienen que desarrollar software para ellos, por lo que no puede aducirse que es un tipo de programación complejo para no incluirlo o relegarlo de los estudios. Además, para obtener las máximas prestaciones de estos sistemas es necesario estudiar análisis y diseño de algoritmos paralelos. Este trabajo pretende motivar la discusión sobre cómo incluir los contenidos de programación y algoritmos paralelos. En nuestra opinión una buena opción sería incluir la programación paralela como parte de asignaturas de programación, junto con alguna asignatura específica de paralelismo.

### 1. Motivación

Tradicionalmente la programación paralela se ha venido utilizando para la resolución de problemas con alto coste computacional [3] que estaban en la frontera de los problemas científicos abordables por sistemas computacionales. Esta programación se realizaba por personal especializado, bien informáticos o científicos que aprendían programación paralela para resolver los problemas científicos en los que tra-

bajaban. Para la resolución eficiente de estos problemas es necesaria la utilización de entornos paralelos, y también la optimización o el rediseño de los algoritmos que se implementan para resolver los problemas. En cuanto a entornos de programación se han venido generando herramientas que facilitan la programación en los distintos sistemas: Poxis threads (pthreads) [6], OpenMP [7], MPI [9]... En lo que se refiere a la importancia del estudio y diseño de algoritmos paralelos basta con comprobar que la mayoría de los libros de introducción a la programación paralela dedican una parte importante al estudio de técnicas de programación y diseño de algoritmos [10, 4, 8, 1]. Es también común observar en los últimos tiempos, que en libros de algorítmica secuencial se introducen algunos temas relacionados con la algorítmica y la programación paralela [2, 5].

Más recientemente, con la agregación de varios ordenadores para formar redes de procesadores se consiguió que la programación paralela se popularizara entre los grupos científicos que tenían en sus propios laboratorios un sistema computacional con el que resolver sus problemas de dimensión más reducida, y utilizaban los supercomputadores para abordar problemas de mayor dimensión. La aparición de estándares de programación paralela facilitó el desarrollo de programas portables que se podían utilizar tanto en las redes de ordenadores como en los supercomputadores.

En los últimos años, la aparición de los procesadores multinúcleo ha puesto al alcance del

público en general los sistemas paralelos. Hoy en día es normal disponer de un portátil biprocesador o de un ordenador de sobremesa con cuatro u ocho núcleos, y la tendencia es que el número de núcleos aumente.

En la actualidad nos encontramos, por tanto, con la posibilidad de utilizar el paralelismo a varios niveles: supercomputadores, redes de ordenadores, sistemas distribuidos, procesadores multinúcleo, procesadores gráficos, vídeo consolas, *cloud computing*...

Esta popularización de los sistemas paralelos debería ir acompañada de una difusión similar de las técnicas de programación paralela, de forma que los desarrolladores naturales de aplicaciones informáticas (los ingenieros informáticos) para las plataformas de cómputo actuales y futuras estuvieran en condiciones de hacer un buen uso de los recursos computacionales a su alcance. Para esto es necesario introducir en los estudios de informática los conocimientos básicos de programación paralela, pero también del análisis y diseño de algoritmos paralelos.

Mientras que el estudio de técnicas algorítmicas secuenciales está más o menos unificado (en todos los planes de estudios se incluye, en un curso u otro y con mayor o menor profundidad, el análisis de algoritmos secuenciales y su diseño, ya sea con un enfoque basado en problemas o en paradigmas) no ocurre lo mismo con las técnicas algorítmicas paralelas. Desde nuestro punto de vista, teniendo en cuenta la importancia y la amplia difusión actual de los sistemas paralelos, que previsiblemente se irá ampliando más en el futuro, deberían incluirse conceptos de algorítmica paralela en los nuevos planes de estudio de informática, y debería hacerse un esfuerzo por unificar posturas en cuanto a la docencia en este tema.

Con este trabajo se pretende propiciar el debate en este sentido. Para ello, se empieza analizando la situación actual de la enseñanza de la programación paralela y los algoritmos paralelos en los actuales planes de estudios, centrándonos principalmente en la situación en las universidades en que trabajamos los autores (Granada, La Laguna, Murcia y Politécnica de Valencia). A continuación se analizan distintas

posibilidades de futuro, revisando sus aspectos positivos y negativos. El trabajo proviene de la colaboración continua de los autores durante los últimos años. Trabajan en universidades distintas y en investigación en programación paralela pero en campos variados (esquemas algorítmicos paralelos, entornos de programación paralela, computación heterogénea, álgebra lineal numérica paralela, aplicaciones de la programación paralela...), lo que hace que no se presente un enfoque personal, sino que el trabajo se ha realizado a partir de una visión múltiple.

## 2. La programación paralela y los algoritmos paralelos en los actuales planes de estudios

La enseñanza de algoritmos paralelos requiere de conocimientos previos de sistemas paralelos y de programación paralela, incluyendo entornos y herramientas de programación. Estas materias se encuentran en mayor o menor medida en todos los libros de introducción a la programación paralela. Por tanto, analizaremos la situación en los planes de estudios actuales de las siguientes materias:

- Sistemas computacionales paralelos, arquitecturas paralelas, paralelismo de bajo nivel.
- Nociones básicas de programación paralela, programación concurrente y programación en sistemas distribuidos.
- Lenguajes, entornos y herramientas de programación paralela, *threads*, OpenMP, MPI.
- Análisis de algoritmos paralelos.
- Diseño de programas paralelos, esquemas algorítmicos paralelos.

No todos estos temas se tratan y se deben tratar con la misma amplitud. Por ejemplo, las nociones básicas de programación paralela y las herramientas de programación se estudiarían conjuntamente, y el análisis y diseño de algoritmos deberían formar una unidad. Como

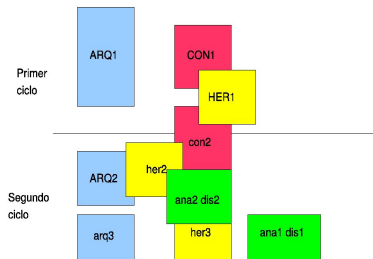


Figura 1: Distribución aproximada en los actuales planes de estudios de informática de contenidos relacionados con la programación paralela

en este trabajo nos centramos principalmente en la programación y algoritmos paralelos, hemos dividido estos temas en más apartados.

Un esquema (aproximado) de la situación actual en los distintos temas se muestra en la figura 1, y a continuación comentamos la situación en cada uno de ellos:

- **ARQUITECTURAS PARALELAS:**

Desde los primeros cursos de las ingenierías informáticas se estudian temas de estructura, tecnología y arquitectura de ordenadores (llamaremos a estas asignaturas **ARQ1**). En estos cursos iniciales se incluyen temas de paralelismo intraprocador y de bajo nivel (segmentación, encauzamiento, replicación de unidades, módulos de memoria, especulación...) En segundo ciclo hay una asignatura troncal de arquitectura de ordenadores (la llamamos **ARQ2**), con contenidos de arquitecturas paralelas. Adicionalmente, dependiendo de la universidad, puede haber alguna asignatura más obligatoria, o una serie de asignaturas optativas que estudian temas de arquitecturas y sistemas paralelos (las llamamos **arq3**, en minúscula para indicar que corresponden a contenidos no obligatorios en todas las universidades): procesadores superescalares y vectoriales, multiprocesadores, VLSI, redes de conexión...

- **PROGRAMACIÓN CONCURRENTE:**

En los planes de estudios suele haber una o dos asignaturas obligatorias con contenidos de programación concurrente en segundo o tercer curso (**CON1**), en ocasiones asociadas a materias de sistemas operativos. Estas asignaturas tienen normalmente los contenidos tradicionales de la programación concurrente (*deadlock*, condición de carrera, abrazo mortal, semáforos...) y algunos conceptos de herramientas de programación paralela, como pueden ser *pthread*s y programación con espacio de direcciones compartido, llamadas remotas, paso de mensajes...

Suele haber otras asignaturas optativas (**con2**) relacionadas con temas de programación paralela, como pueden ser: sistemas de tiempo real, programación distribuida...

- **HERRAMIENTAS DE PROGRAMACIÓN PARALELA:**

Como hemos comentado, en **CON1** se suele incluir el estudio de alguna herramienta para programación paralela, mayoritariamente *pthread*s, pero también algunas veces las herramientas que ofrecen lenguajes tradicionales para gestionar procesos. Llamamos a estos contenidos incluidos en asignaturas de primer ciclo como **HER1**.

También en las asignaturas de segundo ciclo se suele incluir el estudio de herramientas, quizás en las prácticas de la asignatura. Este estudio no está tan generalizado como en las asignaturas básicas de programación paralela, y algunas veces se incluye en asignaturas optativas. Por esto llamamos a estos contenidos de segundo ciclo **her2**.

En algunos casos se incluyen, normalmente en segundo ciclo, asignaturas optativas de herramientas de programación paralela, o de programación en algún tipo de sistema particular (supercomputadores, paso de mensajes, memoria compartida...) A estas asignaturas nos referimos con **her3**.

- ANÁLISIS DE ALGORITMOS PARALELOS:

No es habitual encontrar contenidos obligatorios de análisis de algoritmos paralelos, aunque sí de algoritmos secuenciales. En CON1 se puede tener en cuenta de manera intuitiva temas de prestación de los programas, pero no se suele realizar un estudio sistemático de este tema.

Es usual que en segundo ciclo se incluya alguna asignatura de programación paralela (**ana1**) donde se estudia el análisis de algoritmos paralelos, o que se tengan en cuenta aspectos de prestaciones de programas en asignaturas como **her2** y **her3** (nos referimos a estos contenidos estudiados en asignaturas no propiamente de programación paralela como **ana2**).

- DISEÑO DE ALGORITMOS PARALELOS:

El diseño de algoritmos paralelos se puede estudiar junto con el análisis de los algoritmos (**ana1**). Llamamos a estos contenidos **dis1**. Pero también se puede estudiar metodología de programación paralela sin entrar directamente en el estudio sistemático de esquemas algorítmicos. En las asignaturas no dedicadas directamente a la programación paralela (**her2** y **her3**) se puede abordar el diseño de programas paralelos desde este punto de vista, quizás analizando metodologías para sistemas particulares que se estudian en esas asignaturas. Llamamos a estos contenidos más de metodología de la programación como **dis2**.

Como vemos, la situación en arquitecturas paralelas y en programación y algoritmos paralelos es totalmente distinta. En arquitectura se incluyen en los distintos cursos y asignaturas referencias a paralelismo, que será intra-procesador y de bajo nivel en los primeros cursos, y en cursos superiores temas de estructura de sistemas paralelos. La situación en programación es distinta, encontrándonos con que normalmente se estudian las nociones básicas de programación, las estructuras de datos y los

esquemas algorítmicos sin incluir en la mayoría de los casos referencias al paralelismo.

Además, el estudio del paralelismo se suele realizar dentro de asignaturas de arquitecturas de ordenadores o de sistemas operativos, con lo que la visión de alto nivel propia de la algorítmica puede perderse en gran medida. También suele haber alguna asignatura específica de programación concurrente, pero aislada de las restantes asignaturas de programación y centrada en conceptos básicos, por lo que muchas veces tiene una orientación de sistemas operativos más que algorítmica.

La prolongación de la situación actual creemos que daría lugar a formar titulados con conocimientos alejados de los necesarios para trabajar de forma adecuada en el desarrollo de aplicaciones para los sistemas computacionales actuales, por lo que se plantea la necesidad de unificar posturas en cuanto a los conocimientos de algoritmos paralelos que deberían incluirse en los próximos planes de estudios de informática. Dado que la organización futura de los planes de estudios (duración, organización, continuidad de grado y máster...) no está clara todavía, es imposible dar una propuesta definitiva. Por esto, en la siguiente sección se analizan distintas posibilidades sobre las que debatir.

### 3. La programación paralela y los algoritmos paralelos en los futuros planes de estudios

El análisis de los actuales planes de estudios nos lleva a formularnos la siguiente pregunta: ¿Está justificada en la situación actual la diferencia en la forma de enfocar el paralelismo en asignaturas de arquitectura y de programación? Desde nuestro punto de vista, teniendo en cuenta la amplia difusión actual de los sistemas paralelos, el enfoque de la docencia de la programación paralela se debería aproximar al enfoque actual en las asignaturas de arquitecturas, incluyendo referencias al paralelismo en diferentes niveles de los estudios.

El conocimiento de las arquitecturas paralelas es conveniente para poder desarrollar aplicaciones paralelas eficientes, pero no es im-

prescindible su conocimiento para el estudio de herramientas, metodologías y algoritmos paralelos, que pueden ser estudiados de modo abstracto (usando modelos de arquitecturas paralelas) y al mismo tiempo que la arquitectura de ordenadores, consiguiéndose con el estudio de arquitecturas, algoritmos y lenguajes tener una visión general de la programación paralela. Existen modelos simplificados de arquitecturas paralelas que permiten desarrollar, usando herramientas y técnicas de análisis/diseño estandarizadas, programas paralelos capaces de aprovechar de forma muy satisfactoria el potencial de la mayoría de arquitecturas paralelas más difundidas. Por tanto, el aprendizaje de estos conocimientos debería ser obligatorio para un ingeniero informático, dadas las características de los actuales sistemas de cómputo convencionales.

Por otro lado, los conceptos básicos de programación concurrente y de sistemas distribuidos se suelen estudiar de una manera alejada de la programación paralela real, donde lo que se pretende es reducir el tiempo de ejecución de problemas de alto coste computacional, o resolver problemas en tiempo real, para lo que también es necesario tener en cuenta la reducción del tiempo de ejecución. Creemos que deberían estudiarse los temas básicos de programación paralela con referencias a las prestaciones de los programas y algoritmos paralelos, tal como se hace con la programación secuencial, donde se suele estudiar nociones intuitivas de análisis de prestaciones en los primeros cursos, para pasar a un estudio más sistemático del análisis y diseño de algoritmos en cursos sucesivos.

Algunas posibilidades de organización del estudio de la programación paralela en los nuevos planes de estudio se discuten a continuación.

### 3.1. No incluir programación paralela

Se podría decidir no incluir programación paralela en el grado, si se considera que es un tema avanzado y se debe dejar para el postgrado.

Esta postura se basa principalmente en la reducción en cursos y contenidos prevista para

los nuevos planes. Así, en el grado nos centraríamos en conceptos básicos (programación y algoritmos secuenciales), y los conceptos avanzados (la programación paralela) se dejaría para cursos avanzados. Como ya hemos comentado, esta postura no nos parece adecuada pues los sistemas paralelos son los sistemas estándar actuales, y por tanto los titulados de informática deberían tener conocimientos no sólo sobre las arquitecturas paralelas, sino también de la programación. Es verdad que normalmente en algunas asignaturas de arquitectura se incluyen conceptos básicos de programación paralela, pero entendemos que el estudio de la programación debe realizarse con un enfoque abstracto, no centrado en la arquitectura sobre la que van a ejecutarse los programas. Esto, que es válido y admitido para la programación secuencial, no es siempre admitido para la paralela, y dificulta su desarrollo.

Hay que tener en cuenta también que si se equipara el grado con la Ingeniería Informática se estarían lanzando al mercado profesionales sin formación en competencias básicas relacionadas con la programación de los computadores del mercado actual.

### 3.2. Asignaturas de programación paralela

Se puede decidir incluir programación paralela, pero entendiendo que es un tipo de programación diferenciado, por lo que debe haber al menos una asignatura de fundamentos básicos de paralelismo y concurrencia.

Esta es la situación en muchos casos, donde se incluye una asignatura de programación concurrente o para sistemas distribuidos, y complementariamente se pueden incluir en asignaturas de arquitecturas conceptos de programación paralela. La ventaja de este enfoque es que el profesorado de estas asignaturas se especializaría en paralelismo, mientras que el impartir los conceptos propios de paralelismo dentro de asignaturas de arquitectura o programación secuencial podría ocasionar que el profesorado no estuviera especializado en paralelismo y diera a su enseñanza un enfoque de más bajo nivel o lo incluyera como un apéndice del contenido general del curso.

Si este enfoque de asignaturas diferenciadas podía ser adecuado hace unos años, cuando el paralelismo no estaba tan ampliamente difundido, no está tan claro que sea una buena opción para el futuro, pues seguiría diferenciando la programación paralela de la secuencial (que efectivamente son diferentes) situándolas a un nivel distinto de importancia (que no está tan claro que lo estén).

### 3.3. Integración en otras asignaturas

Se puede considerar incluir programación paralela pues todos los computadores actuales son paralelos y, teniendo en cuenta los diferentes aspectos del paralelismo (arquitecturas, herramientas, programación, algoritmos...), decidir que deben incluirse aspectos de paralelismo en diferentes asignaturas.

Quizás esta sería la opción más adecuada, dada la omnipresencia (actual y prevista para el futuro) de los sistemas paralelos, pero por los motivos mencionados en el apartado anterior puede que cambiar a un enfoque de este tipo en el momento actual no sea lo más adecuado. Está claro que esta postura se puede compaginar con la anterior en cuanto a la conveniencia de incluir alguna asignatura específica de nociones básicas de programación paralela, y con la de incluir nociones de programación paralela en las asignaturas de arquitectura, pero no se dejaría el peso de la programación en esas asignaturas que tratan el paralelismo a más bajo nivel.

La organización en este caso se muestra en la figura 2. Comentamos los aspectos generales de esta organización. En el grado se incluirían conocimientos obligatorios de paralelismo a todos los niveles: arquitecturas (ARQ), programación y herramientas paralelas (CON + HER) y análisis y diseño de algoritmos paralelos (ANA + DIS). En los postgrados estarían la mayoría de las asignaturas optativas, aunque podrían incluirse algunas en los grados si así se estima conveniente. Los volúmenes de cada apartado en la figura no están relacionados con la importancia que se concede a cada uno de ellos. El estudio de las nociones básicas de programación paralela y de herramientas de programación paralela se contempla como un todo

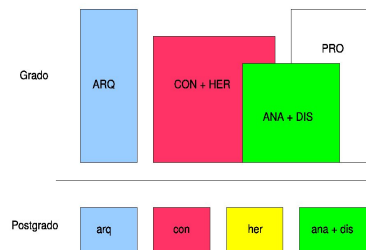


Figura 2: Organización de la programación paralela en los futuros planes de estudios

(CON + HER), y el análisis y diseño de algoritmos paralelos se abordaría de forma conjunta (ANA + IDS), pero no necesariamente de una manera diferenciada, sino que puede ir dentro de CON + HER y de asignaturas propias de programación secuencial PRO. Discutimos brevemente los contenidos de cada uno de los apartados:

- ARQUITECTURAS PARALELAS:

Entendemos que la estructura actual de la enseñanza de estructura de ordenadores y arquitecturas paralelas es la adecuada, por lo que se incluye un módulo ARQ que abarca todos los cursos del grado. Estas asignaturas contendrían temas de programación paralela sólo para ejemplificar el uso de las arquitecturas paralelas que se estudien, pero el estudio de la programación y los algoritmos paralelos se dejaría a asignaturas propias de programación.

- PROGRAMACIÓN CONCURRENTE, HERRAMIENTAS DE PROGRAMACIÓN:

Algunas nociones de programación paralela se pueden incluir en otras asignaturas. Por ejemplo, el estudio de *threads* y algunas nociones básicas de su programación puede incluirse en asignaturas de sistemas operativos o arquitecturas; y al utilizar lenguajes que incluyen herramientas

para paralelismo se puede indicar la existencia de estas herramientas y mostrar su utilización (por ejemplo en C la gestión de procesos, o la opción de compilación `-openmp`, y la gestión de la concurrencia en Java). Pero en cualquier caso, este tema debería incluirse en una o dos asignaturas especializadas, dedicadas a programación concurrente, distribuida, de tiempo real..., y estas asignaturas no se incluirían en el primer curso.

- **ANÁLISIS Y DISEÑO DE ALGORITMOS PARALELOS:**

No creemos necesario en el grado la inclusión de una asignatura específica para estos temas diferenciada de las asignaturas de programación concurrente y distribuida, aunque si se considerara conveniente incluir más asignaturas relacionadas con el paralelismo alguna de ellas podría estar dedicada a supercomputación y a reducción del tiempo de solución de problemas de alto coste, y sería esta asignatura la que incluiría estos contenidos.

Lo que sí creemos necesario es el estudio de algoritmos paralelos desde un nivel de abstracción similar al que se utiliza en la actualidad para abordar el estudio de algoritmos secuenciales. Se deberían incluir temas de análisis y diseño de algoritmos paralelos en las asignaturas propias de programación, tanto las de programación secuencial (**PRO**) como las de programación concurrente y distribuida (**CON + HER**). Esta visión se ve reforzada por la inclusión de capítulos de paralelismo en libros clásicos y recientes de algoritmos [2, 5]. Cuando se describe el modelo de máquina secuencial se puede incluir uno equivalente de máquina paralela. Al estudiar el análisis de algoritmos secuenciales se puede ver de forma intuitiva la reducción en el tiempo que se puede conseguir con la programación paralela, nociones básicas de análisis de algoritmos paralelos, y las dificultades de obtener las máximas prestaciones. Se podría incluir algún esquema paralelo adicionalmente a

los secuenciales. Las herramientas de programación se incluirían en **CON + HER** pero, como hemos mencionado, algunas que sean parte de un lenguaje secuencial se pueden estudiar junto con éste.

#### 4. Conclusiones

En este trabajo se analiza la situación de la enseñanza de la programación y los algoritmos paralelos en los planes de estudios de Informática. Debido a la difusión creciente de los sistemas paralelos y al proceso de reforma de los planes de estudio, creemos que es el momento adecuado para replantearse la organización de la enseñanza de estos temas.

Los ingenieros en informática son los profesionales que trabajan con estos sistemas y que tienen que desarrollar software para sistemas paralelos, por lo que en nuestra opinión no puede aducirse que es un tipo de programación complejo para no incluirlo o relegarlo de los estudios. Para obtener las máximas prestaciones de estos sistemas es necesario estudiar técnicas de análisis y diseño de algoritmos paralelos, y deberían estudiarse con un nivel de abstracción similar al que se utiliza en la programación secuencial.

Este trabajo pretende motivar la discusión sobre cómo incluir contenidos de programación y algoritmos paralelos en los nuevos planes de estudios. En nuestra opinión una buena opción sería incluir la programación paralela como parte de las asignaturas de programación, aunque con alguna asignatura específica de paralelismo.

#### 5. Agradecimientos

Este trabajo ha sido financiado en parte por la Fundación Séneca, proyecto 08763/PI/08, y por el Ministerio de Educación, proyecto TIN2008-06570-C04.

#### Referencias

- [1] Francisco Almeida, Domingo Giménez, José Miguel Mantas, and Antonio M.

- Vidal. *Introducción a la programación paralela*. Paraninfo Cengage Learning, 2008.
- [2] Brassard and Bratley. *Fundamentos de algoritmia*. Prentice-Hall, 1997.
- [3] Jack Dongarra, Ian Foster, Geoffrey Fox, William Gropp, Ken Kennedy, Linda Torczon, and Andy White, editors. *Sourcebook of Parallel Computing*. Morgan Kaufmann Publishers, 2003.
- [4] A. Grama, A. Gupta, G. Karypis, and V. Kumar. *Introduction to Parallel Computing*. Addison-Wesley, second edition, 2003.
- [5] Ellis Horowitz, Sartaj Shani, and Sangt-hevar Rajasekaran. *Computer Algorithms/C++*. Computer Science Press, 1997.
- [6] Bradford Nichols, Dick Buttlar, and Jacqueline Proulx Farrel. *Pthreads programming: A Posix Standard for Better Multiprocessing*. O'Reilly, 1996.
- [7] OpenMP. <http://www.openmp.org/blog/>.
- [8] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. McGraw Hill, 2004.
- [9] Marc Snir and William Gropp. *MPI. The Complete Reference. 2nd edition*. The MIT Press, 1998.
- [10] Barry Wilkinson and Michael Allen. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers*. Prentice-Hall, second edition, 2005.