

# Un curso práctico de Programación Paralela basado en problemas del Concurso de Programación Paralela

Domingo Giménez

Departamento de Informática y Sistemas, Universidad de Murcia



JENUI, Almería, 8 de julio de 2016

# Problema a solucionar

Número de estudiantes en el curso, y número y porcentaje de alumnos que aprueban.

Curso	03	04	05	06	07	08	09	10	11	12	13	14
Alumnos	10	8	13	18	10	4	13	12	19	29	26	41
#Aprob.	6	5	9	7	7	4	8	7	10	12	9	19
%Aprob.	60.0	62.5	69.2	38.9	70.0	100	61.5	58.3	52.6	41.4	34.6	46.3

# Contenido

- 1 Organización del curso
- 2 Concurso de Programación Paralela
- 3 Uso del sistema del Concurso
- 4 Evaluación del aprendizaje

# Descripción del curso

## Metodología de la Programación Paralela

- Primer cuatrimestre de cuarto del Grado en Ingeniería Informática.
- En el curso anterior una asignatura de Arquitectura y Organización de Computadores.
- En segundo una asignatura de Programación Concurrente y Distribuida.
- Obligatoria en la especialización de Ingeniería del Software, optativa en la de Ingeniería de Computadores.
- Alrededor de 35 estudiantes por año.
- Aproximación práctica, con problemas y herramientas del Concurso Español de Programación Paralela (<http://luna.inf.um.es>).

# Contenido

- Repaso de arquitecturas paralelas
- Paradigmas de programación paralela
- Herramientas de programación paralela: OpenMP, MPI y CUDA
- Esquemas algorítmicos paralelos
- Análisis de algoritmos paralelos
- Metodología de la programación paralela

# Planificación del curso

Se evalúa con trabajos prácticos, trabajando con las herramientas y problemas del Concurso de Programación Paralela.

Tema	Semana	
	Teoría	Práctica
Introducción a la programación paralela, necesidad de paralelismo	1	
Sistemas paralelos y paradigmas de programación paralela	2	3
Programación de memoria compartida, OpenMP	3	4
Programación de paso de mensajes, MPI	4	5
Computación híbrida y heterogénea	5	6
Programación de <i>manycores</i> , CUDA	6	7
Paralelismo de datos y particionado de datos	7	8
Algoritmos relajados y paralelismo síncrono	8	9
Algoritmos en árbol y grafo, <i>Pipeline</i>	9	10
Paralelismo en Divide y Vencerás y Programación Dinámica	10	11
Paralelismo en búsqueda en árbol, Maestro-Esclavo, Bolsa de tareas, Granja de procesos, Trabajadores replicados	11	12
Análisis de algoritmos paralelos	12	
Metodología de la programación paralela	13	

# Partes de la evaluación

El curso se evalúa con prácticas, organizadas en tres bloques:

- Presenciales:
  - Entornos de paralelismo (25%):  
Cinco sesiones (5% cada una): Sistemas paralelos, OpenMP, MPI, híbrido y heterogéneo, CUDA.
  - Esquemas algorítmicos paralelos (50%):  
Cinco sesiones (10% cada una): cada una sobre uno o dos paradigmas algorítmicos.

Después de cada sesión práctica se revisan las soluciones de los alumnos, que tienen cuatro días para completar o mejorar las soluciones.

- Práctica no presencial (25%):  
Cada alumno trabaja con un problema del Concurso, hace análisis teórico y experimental, y aplica las técnicas estudiadas en clase de teoría.

# Estructura

- La sexta edición en 2016.
- *On-line*, con las *Jornadas de Paralelismo*, en Septiembre, este año coincidiendo con el CEDI.
- Grupos de 1 o 2 estudiantes de universidades españolas, de no más de 28 años, y un profesor-entrenador.
- No estudiantes, no españoles o mayores de 28 pueden participar en modo Exhibición.
- Se resuelven problemas con OpenMP (un nodo con 12 núcleos), MPI o MPI+OpenMP (dos nodos, 6 núcleos cada uno), y CUDA (una GPU Tesla K20c con 2496 núcleos CUDA).
- Se proporciona una solución secuencial de cada problema.
- La clasificación se calcula con la velocidad respecto a la solución secuencial proporcionada.
- En tiempo real y de forma automática, usando Mooshak



# Mooshak


- Una herramienta para organizar concursos. En la Universidad de Murcia la usamos para cursos de programación y algoritmos.
- Adaptada para trabajar en un clúster y calcular la puntuación en función de la velocidad.
- Se accede a Mooshak a través de la web del Concurso ([luna.inf.um.es](http://luna.inf.um.es))
- Los participantes envían soluciones a Mooshak, que las compila y enlaza con un esquema (para entrada/salida, validación y puntuación), envía el código compilado a una cola, obtiene los resultados y el tiempo de ejecución, valida la solución y calcula la clasificación.
- El profesor puede controlar el trabajo de los participantes o alumnos *on-line* y en tiempo real.

# Web del Concurso

VI Spanish Parallel Programming Contest - Mozilla Firefox

Correo :: Entrada

luna.inf.um.es/2016/index.php?lang=en



## VI Spanish Parallel Programming Contest

L. Peter Deutsch  
To iterate is human,  
to recurse divine.

Home | Rules | REGISTRATION | Organization | RESULTS | Mooshak | Contact

### Important information


**The VI Spanish Parallel Programming Contest (non-Spanish teams can participate in exhibition mode, without option to price) will celebrate online one day between 14 and 16 of September 2016, as part of the [Jornadas Sarteco](#), to celebrate in Salamanca, Spain. The inscription will be through this page beginning middle June and until September 8th. The inscriptions be given an account in the tests of previous editions to practice with them as warm-up session.**


Since parallel systems have become a standard, and are easy to find nowadays from domestic laptops to the most powerful supercomputers ([Top 500 Supercomputing](#)), the Spanish Parallel Programming Contest is organized with the aim of promoting this programming paradigm among University students, following the lines of other contests such [ACM Programming Contest](#) (sequential), [Maraton of Parallel programming](#) (annually held in Brazil), [Student Cluster Competition](#) (in SC)...

This contest is directed to University students (first or second cycle degree, master or doctorate) with a maximum age of 28 at the time of registration. **NEW** Non Spanish individuals or teams, without limits of age, not necessarily from university, can participate in Exhibition Mode: without be included in the classification and without option to prizes.

The contest will consist of solving a number of problems in a given time. For each of the problems a sequential solution will be provided, and the teams will then develop their own faster solutions using a set of given technologies such as [OpenMP](#), [MPI](#) or [CUDA](#).

#### Languages

 Español

 English

#### History

Records

- 2016 - Salamanca
- 2015 - Córdoba
- 2014 - Valladolid
- 2013 - Madrid
- 2012 - Elche
- 2011 - La Laguna

#### Related links

- Workshop EuroEDUPAR
- ACM Programming Contest
- Maraton of Parallel programming

# Web del Concurso - Envíos

11:52 3 users Logout 18:58

**Parallel Programming Contest 2015**

Contest running

**Admin**
**Judge**

[Warning]

**View:**

- Pending
- Questions
- Submissions
- Printouts
- Balloons
- Evolution
- Statistics
- Ranking

**Problems: Teams:**

All
  All
  Grouped

A AramayoRivero  
 B BelmonteHuescar  
 C CaballerMartinez  
 D CalinSanchez  
 E CaroLopez  
 F ChuecosMartinez  
   ConesaGarre  
   CorbalanNavarro  
   Domingo

**Update:**  
 every  minutes  
 with  lines

[Logout]

#	Contest Time	Country	Team	Problem	Language	Result	Time (ms)	State
<a href="#">S508</a>	5282:42:47		MPP1516 CalinSanchez	<a href="#">A</a>	cpp_wrapper	0 Accepted	8937	final
<a href="#">S507</a>	5282:26:23		MPP1516 CalinSanchez	<a href="#">A</a>	cpp_wrapper	0 Wrong Answer	9443	final
<a href="#">S506</a>	5212:49:22		MPP1516 CalinSanchez	<a href="#">A</a>	cpp_wrapper	0 Wrong Answer	18283	final
<a href="#">S505</a>	3147:41:49		Organization records	<a href="#">F</a>	cpp_wrapper	0 Accepted	72	final
<a href="#">S504</a>	3147:38:22		Organization records	<a href="#">E</a>	cpp_wrapper	0 Accepted	89	final
<a href="#">S503</a>	2967:35:49		MPP1516 CorbalanNavarro	<a href="#">F</a>	cpp_wrapper	0 Accepted	69	final
<a href="#">S502</a>	2967:34:31		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	100	final
<a href="#">S501</a>	2967:33:50		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	85	final
<a href="#">S500</a>	2957:56:29		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	71	final
<a href="#">S499</a>	2957:53:44		MPP1516 CorbalanNavarro	<a href="#">F</a>	cpp_wrapper	0 Accepted	199	final
<a href="#">S498</a>	2954:15:32		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	87	final
<a href="#">S497</a>	2954:13:48		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	114	final
<a href="#">S496</a>	2954:12:51		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	103	final
<a href="#">S495</a>	2954:12:24		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	114	final
<a href="#">S494</a>	2954:03:55		MPP1516 CorbalanNavarro	<a href="#">E</a>	cpp_wrapper	0 Accepted	87	final

Page 1 of 34

Comment to problem  A  B  C  D  E  F

Subject:

# Web del Concurso - Clasificación

View 1.1.1 | 3 users | 18:38

Parallel Programming Contest 2015
Contest running

Admin
Judge

View:

- Pending
- Questions
- Submissions
- Printouts
- Balloons
- Evolution
- Statistics
- Ranking

Problems: Teams:

All  A  B  C  D  E  F

A AramayoRivero

B BelmonteHuescar

C CaballerMartinez

D CalinSanchez

E Carolopez

F ChuecosMartinez

ConesaGarre

CorbalanNavarro

Domingo

Update:

every  minutes

with  lines

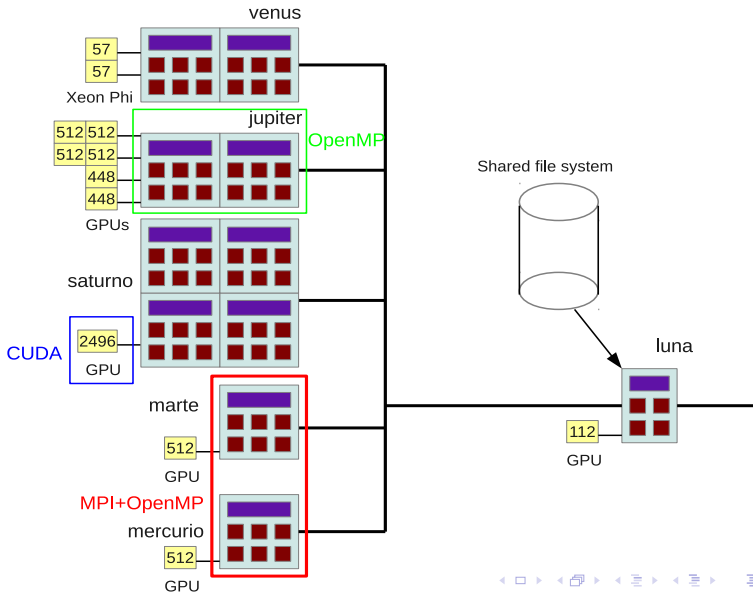
[Logout]

#	Country	Team	Problems						Solved	Points
			A	B	C	D	E	F		
1		Organization records	9.923670 (823 21.114216 1 1)	10.000000 (253 25.877470 1 1)	10.000000 (2716 10.045655 1 1)	7.506076 (1399 7.506076 1 1)	9.549455 (89 387.764045 2 2)	9.614054 (72 519.833333 2 2)	6	56.593254
2		exhibition UDIGO	9.898473 (825 21.060606 6 9)		9.991909 (2718 10.037528 7 10)		4.131926 (205 167.780488 2 3)	2.285637 (301 123.584718 1 1)	4	26.307944
3		teams UAB	4.168824 (1844 8.869848 1 1)	8.998451 (280 23.285714 2 2)	9.899713 (2741 9.944911 2 2)		0.249799 (3105 10.143317 3 4)	0.089534 (6420 4.841121 2 2)	5	23.406321
4		MPP1516 CorbalanNavarro	0.000000 (18446 0.000000 2 2)				10.000000 (85 406.058824 8 8)	10.000000 (67 540.701493 28 29)	3	20.000000
5		teams UCM1	1.778683 (3804 3.784437 4 5)	1.978792 (1111 5.120612 7 10)	9.923615 (2735 9.968921 1 1)		0.241652 (3200 9.812500 6 8)	0.076642 (7290 4.144033 3 3)	5	13.999384
6		teams UEX	1.999397 (3464 4.254042 1 1)		9.871960 (2748 9.917031 3 3)		0.249358 (3110 10.125402 2 2)	0.076655 (7289 4.144739 5 5)	4	12.197370
7		teams UCLM2	1.968424 (3508 4.188141 2 2)		9.903689 (2740 9.948905 1 1)		0.239261 (3229 9.715392 1 1)	0.076629 (7291 4.143327 1 1)	4	12.188003
8		teams UCLM1	1.964953 (3513 4.180757 2 2)	0.362215 (3510 0.937322 1 3)	7.749393 (3415 7.784773 6 6)	1.457711 (1407 1.457711 16 16)	0.236671 (3261 9.610242 8 8)	0.162871 (3824 8.806485 8 9)	6	11.933816
9		MPP1516 SanchezMunoz	10.000000 (817 21.276622 5 5)						1	10.000000
10		teams UALM	1.872917 (3651 3.984936 2 2)		7.762215 (3410 7.797654 2 2)		0.250331 (3099 10.164892 4 4)	0.078234 (7170 4.230126 3 5)	4	9.963697
11		MPP1516 SanchezLopez	0.000000 (0 0.000000 5 10)		9.923615 (2735 9.968921 9 9)				1	9.923615
12		MPP1516 PerezAguilar	1.995838 (3469 4.246469 3 5)						1	1.995838
13		MPP1516 GarciaCamascos	1.980298 (3491 4.213406 8 17)						1	1.980298
14		MPP1516 ChuecosMartinez	1.520873 (3476 3.235903 11 24)						1	1.520873

Comment to problem  A  B  C  D  E  F

Subject:

# Sistema computacional



# Herramientas y entornos de paralelismo

Cinco sesiones: trabajo con entornos diversos y el sistema del concurso.

- Entornos de paralelismo.  
Ejemplos básicos de paralelismo con algoritmos de ordenación en C++, Java y pthreads. Generalizar a más hilos y procesos, y comparar prestaciones. Experimentos con un problema de un proyecto de introducción del paralelismo en estudios de formación profesional.
- OpenMP.  
El mismo ejemplo y una multiplicación de matrices: experimentar con paralelización de bucles a tres niveles y con la cláusula collapse. Hacer versiones con secciones y tareas.
- MPI.  
Para el ejemplo de ordenación, usar comunicaciones colectivas. En la multiplicación, usar comunicaciones asíncronas.
- Computación híbrida y heterogénea.  
Versión MPI+OpenMP de la multiplicación de matrices. Uso eficiente del clúster heterogéneo.
- CUDA.  
Multiplicación de matrices: experimentos variando el número de hilos por bloque.

# Paradigmas algorítmicos

Cinco sesiones: trabajo con paradigmas de algoritmos paralelos.

Prob	Paralelismo de datos	Particionado de datos	Algoritmos relajados	Paralelismo síncrono	Algoritmos en árbol	Pipeline	Divide y vencerás	Programación dinámica	Búsqueda en árbol	Trabajadores replicados
11A	X	X								
11B	X			9				X		
11C							X			
11D	X	X	X						X	X
11E										
12A	8	8	X					X		
12B	X	X	X				X			
12C	X	X			10		X			
12D									X	X
12E								11		
12A-CUDA	X	X		X						
13A	X	X								
13D	X	X		X						
13F									12	12
14A	X	X	X							
14D	X	X	X						X	X
14F										
15A	X			X	X	10				X
15C	X	X	X	X						
15E	X	X	X							

## Algoritmos en árbol y *Pipeline*

- Problema A de 2015: Submatrices de tamaño  $m \times m$  en la diagonal de una matriz de tamaño  $n \times n$ , con solapamiento del subbloque  $l \times l$  inferior de una submatriz con el superior de la siguiente. Se dan varias iteraciones de cálculo de los cuadrados de las submatrices, empezando por la superior-izquierda hacia abajo.  
Resolverlo con un esquema *Pipeline* en OpenMP.  
Los alumnos pueden dar soluciones válidas para 4 y 6 hilos o generales, en cuyo caso la puntuación es mayor.
- Problema C de 2012: sucesivas ordenaciones de los datos de una matriz y cuadrados de la matriz.  
Paralelizar la ordenación en MPI con una estructura de comunicación en árbol.  
Los alumnos pueden dar soluciones válidas para 2, 4 y 8 procesos o generales, con las que se obtiene mayor puntuación.



# Práctica final (25%)

- Trabajo autónomo e individual.
- Cada alumno trabaja con un problema del Concurso.
- Partes que se evalúan:
  - Explicación del método secuencial y propuestas de mejora (2%).
  - OpenMP. Explicación del funcionamiento de las soluciones en la tabla de récords, y propuesta de modificaciones para mejorar el tiempo de ejecución (4%).
  - MPI. Explicación del funcionamiento de las soluciones en la tabla de récords, y propuesta de modificaciones para mejorar el tiempo de ejecución (4%).
  - MPI+OpenMP. Explicación del funcionamiento de las soluciones en la tabla de récords, y propuesta de modificaciones para mejorar el tiempo de ejecución (2%).
  - Análisis teórico: tiempo de ejecución, *speed-up*, eficiencia e isoeficiencia (4%).
  - Estudio experimental: tiempo de ejecución, *speed-up*, eficiencia e isoeficiencia (5%).
  - Validación (2%).
  - Contraste teórico / experimental (2%).
- Opcional:
  - Implementación CUDA.
  - Nuevos récords.

# Objetivo

Incrementar el número de alumnos que alcanzan el nivel mínimo para aprobar.

Número de estudiantes en el curso, y número y porcentaje de alumnos que aprueban.

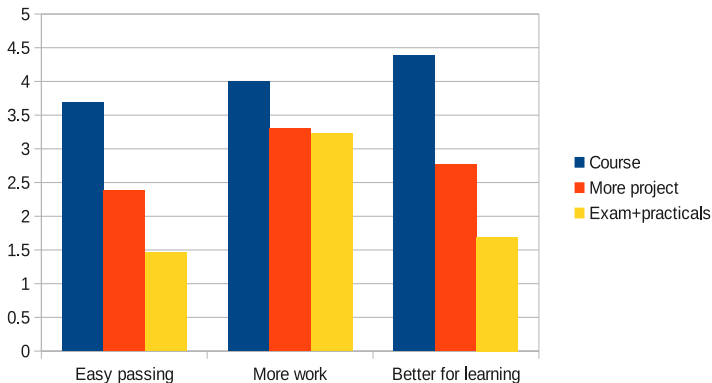
Curso	03	04	05	06	07	08	09	10	11	12	13	14	15
Alumnos	10	8	13	18	10	4	13	12	19	29	26	41	35
#Aprob.	6	5	9	7	7	4	8	7	10	12	9	19	16-22
%Aprob.	60.0	62.5	69.2	38.9	70.0	100	61.5	58.3	52.6	41.4	34.6	46.3	46-63

Número de alumnos que siguen el sistema de evaluación continua y no continua, número de soluciones enviadas, y puntuación media, después de cada sesión práctica.

Sesión:	1	2	3	4	5	6	7	8	9	10
No continua	3	3	7	7	7	7	7	7	7	8
Continua	28	28	24	24	24	24	24	24	24	23
Soluciones	27	27	24	23	24	24	24	24	19	13
Puntuación media	7.2	6.8	5.1	5.6	9.9	6.1	6.2	3.4	4.9	3.4

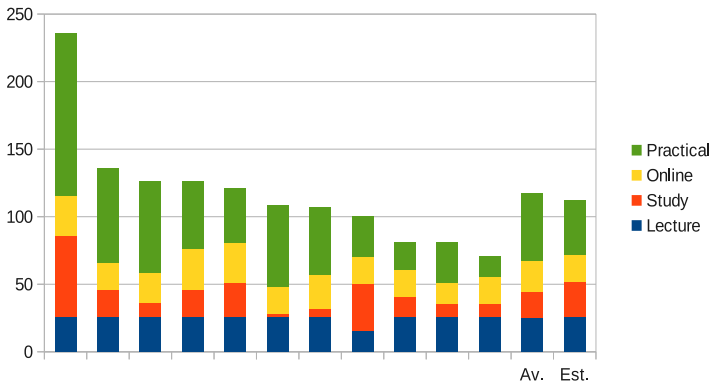
## Opinión sobre la estructura del curso

Prefieren la organización del curso este año a la de años anteriores.  
Pero piensan que supone un ligero aumento de la carga de trabajo



## Carga de trabajo

La carga de trabajo en las distintas parte es similar a la estimada por el profesor,  
con menos tiempo dedicado a estudio de teoría y algo de sobrecarga en la realización de las prácticas.



# Un curso práctico de Programación Paralela basado en problemas del Concurso de Programación Paralela

Domingo Giménez

Departamento de Informática y Sistemas, Universidad de Murcia



JENUI, Almería, 8 de julio de 2016

¿Preguntas? ¿Comentarios? ¿Recomendaciones?

Inscripción hasta el 8 de septiembre  
(<http://luna.inf.um.es>)