

Financian:



CESGA Computational Science Summer School

Inscripción
1-30 Abril 2009

Centro de Supercomputación de Galicia (CESGA)
Avda. de Vigo, s/n
Campus Sur
Santiago de Compostela
<http://www.cesga.es/SummerSchool2009>

Plazas limitadas



Colaboran:



Índice:

1	Objetivos.....	2
2	Destinatarios	2
3	Fechas y lugar de celebración	2
4	Organización y financiación.....	2
5	Temática de los cursos.....	3
6	Calendario	5
7	Solicitud de plaza	6
8	Programas de los cursos de formación.....	7



1 Objetivos

CESGA Computational Science Summer School 2009 pretende fomentar la actividad computacional dentro de la comunidad investigadora, especialmente en la resolución de problemas con grandes necesidades. Se compone tanto de tareas formativas como de actividades de investigación. Además, la formación se complementará con la presentación de aplicaciones avanzadas de la ciencia computacional en otras áreas. Los objetivos de la CESGA Computational Science Summer School 2009 son los siguientes:

- Formar a jóvenes licenciados, ingenieros o investigadores en técnicas de computación y nuevas herramientas de cálculo científico.
- Mostrar los usos actuales de la ciencia computacional en diversas aplicaciones como Matemáticas, Biomedicina, Nanotecnología, etc.

2 Destinatarios

Esta Summer School está destinada principalmente a jóvenes licenciados o ingenieros y a jóvenes investigadores que deseen iniciarse en la ciencia computacional, tanto en el campo de la investigación como en el de las aplicaciones industriales.

3 Fechas y lugar de celebración

Del 15 de junio al 30 de septiembre de 2009 en el Centro de Supercomputación de Galicia (CESGA), Av. de Vigo, s/n, Campus Sur, Santiago de Compostela.

4 Organización y financiación

Organiza: Centro de Supercomputación de Galicia

Contacto: cursos@cesga.es

Comité organizador:

Fernando Bouzas Sierra
Andrés Gómez Tato
María Teresa Sánchez Rúa

Financian:

- CESGA
- Nodo CESGA del Proyecto i-MATH.
- Red temática Gallega de Computación de Altas Prestaciones (Red G-HPC).
- Red Mathematica Consulting & Computing de Galicia.

Colaboran:

- HP.
- Dell.

5 Temática de los cursos

Los cursos que se impartirán incluyen formación en programación, optimización y depurado de aplicaciones científicas, tanto secuenciales como paralelas:

<i>Nombre del curso</i>	<i>Fechas</i>	<i>Acrónimo</i>
Acceso y utilización de recursos (FT, SVGD) en el CESGA	15/06	INTRO
Programación en Fortran	16/06-19/06	PRF
Programación en C	22/06-26/06	PRC
Matemática Computacional: Compilación, ejecución y optimización de programas	29/06-03/07	MTC
Introducción a algoritmos de resolución de aplicaciones científicas	06/07-10/07	ALG
Introducción a la programación en MPI	20/07-24/07	MPI
Programación paralela utilizando directivas OpenMP	27/07-31/07	OMP
Herramientas de desarrollo de aplicaciones paralelas: Debugging y análisis del rendimiento	07/09-11/09	DEB



Adicionalmente a los cursos de formación, se celebrarán cuatro seminarios de presentación de la necesidad de las matemáticas y la computación en diversas áreas científicas, impartidos por investigadores especialistas de prestigio internacional.

Formato de los cursos:

- Presencial, entre 15 y 20 horas teórico/prácticas en sesiones de 4 horas (más descansos)
- Cada alumno recibirá una copia en papel de la documentación y ejercicios a realizar.
- Tutorías personalizadas realizadas por el profesorado y personal técnico del CESGA.
- N° plazas: 15.

Profesorado:

- Grupo de Arquitectura de Computadores, Universidad de A Coruña.
- Grupo de Arquitectura de Computadores, Universidad de Santiago de Compostela.
- Centro de Supercomputación de Galicia.

6 Calendario

Lunes		Martes		Miércoles		Jueves		Viernes	
Junio 2009									
15:	Presentación INTRO	16:	PRF	17:	PRF	18:	PRF	19:	PRF
22:	PRC	23:	PRC	24:	PRC	25:	PRC	25:	PRC
29:	MTC	30:	MTC	01:		02:		03:	
Julio 2009									
29:		30:		01:	MTC	02:	MTC	03:	MTC
06:	ALG	07:	ALG	08:	ALG	09:	ALG	10:	ALG
13:		14:		15:		16:		17:	
20:	MPI	21:	MPI	22:	MPI	23:	MPI	24:	MPI
27:	OMP	28:	OMP	29:	OMP	30:	OMP	31:	OMP
Septiembre 2009									
29:		01:		02:		03:		04:	
07:	DEB	08:	DEB	09:	DEB	10:	DEB	11:	DEB

Observaciones:

- 1 alumno por PC.
- Los alumnos tendrán asignado un tutor durante la celebración de la Summer School.
- Además, dispondrán de cuentas de usuario en CESGA durante la

- celebración de la escuela para acceder a los distintos recursos del centro.
- Para los miembros del proyecto Ingenio Mathematica y de la Red temática Gallega de Computación de Altas Prestaciones (Red G-HPC), se ofertan cuatro estancias de investigación (2+2) durante la celebración de la Summer School. Más información en el apartado *Estancias de Investigación* que puede encontrarse en la siguiente página web <http://www.cesga.es/SummerSchool2009>.

7 Solicitud de plaza

Del 1 al 30 de abril de 2009 a través de la página web <http://www.cesga.es/SummerSchool2009>.

La concesión de plazas se realizará por parte del Comité Organizador. Se notificará la concesión de plaza a los solicitantes antes del 15 de mayo de 2009.

Cuotas de inscripción:

Las cuotas de inscripción para la Summer School son:

	<i>Cuota de inscripción</i>
Miembro i-MATH o Red G-HPC	Gratuita
Investigador Univ. Gallegas / CSIC	700€
Otros	1750€

La cuota de inscripción incluye toda la documentación correspondiente a los cursos de formación.

Para obtener el diploma acreditativo de asistencia a la Summer School será necesario justificar el 80% de horas de participación en los cursos de formación.



8 Programas de los cursos de formación

Programación en Fortran:

Del 16 al 19 de junio

El objetivo del curso consiste en formar al alumno en profundidad sobre la programación en el lenguaje Fortran 90/95. Su interés radica en su extendida utilización en el ámbito de la ciencia computacional y de la computación de altas prestaciones, tanto en lo que respecta a su uso en aplicaciones y bibliotecas científicas y de ingeniería, como a su elección como lenguaje base para lenguajes de programación paralela. El temario se complementará con la realización de prácticas dirigidas utilizando el compilador de Fortran 90/95 g95 (<http://g95.sourceforge.net>).

Temario:

1. Introducción al lenguaje Fortran 90/95
2. Tipos de datos:
 - Tipos de datos intrínsecos.
 - Parámetro KIND.
 - Tipos de datos derivados.
3. Control de flujo:
 - Bloque IF.
 - Bloque CASE.
 - Operadores lógicos y relacionales.
4. Bucles (DO, EXIT, CYCLE).
5. Arrays:
 - Notación de arrays.
 - Sentencia WHERE.
 - Arrays y tipos de datos derivados.
 - Arrays multidimensionales.
 - Funciones intrínsecas con arrays.
6. Entrada/Salida:
 - Formatos para E/S.



- Descriptores de formato.
 - Especificadores de E/S.
 - Ficheros
7. Procedimientos:
- Funciones.
 - Subrutinas.
 - Argumentos de procedimientos (atributo INTENT).
 - Procedimientos y arrays: arrays automáticos y assumed-shape.
 - Variables locales (SAVE).
 - Procedimientos internos.
 - Recursividad.
 - Paso de procedimientos como parámetro.
 - Argumentos opcionales.
8. Módulos e Interfaces:
- Bloque INTERFACE.
 - Definición y uso de módulos (MODULE).
 - Restricciones de acceso en módulos (PUBLIC/PRIVATE).
 - Definición de procedimientos genéricos.
 - Definición de operadores.
 - Extensión de procedimientos y operadores.
9. Memoria Dinámica: Punteros.
- Arrays dinámicos (ALLOCATABLE).
 - Punteros y estructuras de datos dinámicas.
10. Principales Novedades en Fortran 95:
- Construcción FORALL.
 - Procedimientos puros y elementales.
11. Introducción al Estándar Fortran 2003



Programación en C:

Del 22 al 26 de junio

El objetivo del curso es el de formar al alumno en profundidad sobre la programación en el lenguaje C. Su interés radica en su uso como herramienta de programación en ciencia computacional y de la computación de altas prestaciones, tanto en lo que respecta a su uso en aplicaciones y bibliotecas científicas y de ingeniería, como a su elección como lenguaje base para lenguajes de programación paralela como MPI y OpenMP. Se hará, por tanto, especial hincapié en los aspectos del lenguaje más relevantes para su uso en el ámbito de la computación de altas prestaciones. La orientación del curso es eminentemente práctica, por lo que se irán proponiendo ejercicios a los alumnos a medida que se vayan explicando los diferentes conceptos. Así mismo, se intercalarán laboratorios prácticos más extensos al final de algunas lecciones.

Temario:

1. Introducción:

- Qué es y cómo funciona ANSI C.
- Evolución histórica de C.
- Estructura básica de un programa en C.
- Entornos de compilación y ejecución en Linux. icc y gcc.
- Portabilidad del código

2. Declaraciones y expresiones:

- Tipos de datos y variables.
- Ámbito y visibilidad de las variables.
- Constantes.
- Operadores y expresiones.

3. Sentencias básicas en C:

- Sentencias condicionales.
- Sentencias de control de flujo.

4. Punteros y arrays:

- Concepto de puntero y array.
- Inicialización y acceso.



- Arrays multidimensionales.
 - Relación entre punteros y arrays.
 - Gestión dinámica de memoria.
5. Funciones:
- Estructura de una función.
 - Argumentos y valor de retorno.
 - La función main().
 - Funciones de manejo de caracteres.
 - Punteros a funciones.
6. Tipos avanzados:
- Estructuras.
 - Uniones.
 - Typedef.
 - Casting.
7. Uso de librerías y códigos objeto externos:
- Invocación de funciones en ficheros externos.
 - Librerías estándar en C.
 - Invocación de funciones en otras librerías.
 - Creación de una librería propia.
8. El preprocesador de C:
- Sustitución de strings.
 - Inclusión de ficheros fuente.
 - Macros.
9. Entrada/salida:
- Lectura y escritura por entrada/salida estándar.
 - Lectura y escritura de ficheros.
10. Cálculo científico en C:
- Precisión y estabilidad.
 - Minimización de errores en punto flotante.

- Ejemplos de códigos numéricos en C.



Matemática Computacional: Compilación, ejecución y optimización de programas

Del 29 de junio al 3 de julio

El objetivo de este curso es mejorar la eficiencia de códigos computacionales.

Temario:

1. Compilación/linkado de programas:
 - Descripción del proceso de compilación/linkado.
 - Compiladores C y Fortran.
 - Descripción y uso de las opciones de compilación.
 - Definición y uso de las opciones de optimización.
 - Llamadas a subrutinas Fortran desde C y viceversa.
2. Aritmética del computador y sus implicaciones en computación matemática:
 - Formato de números enteros.
 - Formato en punto flotante IEEE 754.
 - Rango, precisión y redondeo.
 - Excepciones en punto flotante: overflow, underflow, NaN,...
 - Big endian-little endian.
 - Ejemplos aplicados a computación matemática.
3. Utilización de librerías matemáticas:
 - Definición y uso de librerías.
 - Librerías estáticas y dinámicas.
 - Librerías compartidas.
 - Librerías matemáticas (BLAS, LAPACK ...).
 - Creación y manipulación de librerías propias.
4. Optimización del rendimiento de la jerarquía de memoria:
 - La jerarquía de memoria: principio de localidad.
 - Fundamentos de funcionamiento de la memoria caché: carga, ubicación y reemplazo de líneas.
 - Técnicas de optimización software: intercambio de bucles, fusión de



bucles, partición en bloques (blocking),...

5. Depuración y evaluación del rendimiento:

- Herramientas de depuración.
- Herramientas para la evaluación del rendimiento.

6. Introducción a la computación paralela:

- Clasificación de las arquitecturas paralelas.
- Paradigmas de programación paralela.
- Elección del paradigma de programación paralela (memoria distribuida vs memoria compartida).
- Ejemplos de aplicaciones de Introducción a la Computación, paralelización y optimización.



Introducción a algoritmos de resolución de aplicaciones científicas (provisional):

Del 6 al 10 de julio

El objetivo de este curso es mostrar a los usuarios los métodos numéricos más apropiados para cada problema que pretendan resolver.

Temario:

1. Derivación e integración numérica.
2. Resolución de E.D.O: Métodos de Euler, Crank-Nicholson, Runge-Kutta.
3. Resolución de sistemas de ecuaciones lineales:
 - Métodos directos.
 - Métodos iterativos clásicos.
 - Métodos iterativos de descenso: Método de gradiente conjugado y GMRES.
 - Introducción a las técnicas de preconditionamiento.
 - Formatos de almacenamiento de matrices en el ordenador. Elección del formato en función del método empleado.
4. Resolución de sistemas de ecuaciones no lineales:
 - Métodos de punto fijo.
 - Métodos de Newton y quasi-Newton.
 - Método de Broyden.
5. Resolución de E.D.P.: método de diferencias finitas, método de elementos finitos.
6. Cálculo de autovalores y autovectores:
 - Método de la potencia.
 - Métodos iterativos: método de Rayleigh-Ritz, algoritmo de Lanczos.



Introducción a la programación en MPI:

Del 20 al 24 de julio

El estándar de facto para la programación paralela según el paradigma de paso de mensajes es MPI. En este curso se pretende introducir al alumno en los detalles de este modelo de programación y ofrecerle una visión completa de las diferentes posibilidades que ofrece MPI. Se hará un especial hincapié en la componente práctica del aprendizaje, de modo que se introducirán ejemplos asociados a los conceptos y funciones más importantes. Finalmente se abordará la cuestión de la evaluación, depurado y optimización de los códigos paralelos desarrollados. El objetivo final es que el alumno comprenda el paradigma de programación de pase de mensajes, y adquiera conocimientos suficientes para desarrollar códigos paralelos eficientes usando MPI.

Temario:

1. Introducción.
2. MPI: conceptos generales.
3. Funciones de MPI básicas.
4. Comunicaciones colectivas.
5. Tipos de datos derivados.
6. Comunicadores.
7. Topologías.
8. Otros modos de comunicación.
9. Depurado de programas.
10. Introducción a MPI-2



Programación paralela utilizando directivas OpenMP:

Del 27 al 31 de julio

El objetivo del curso es proporcionar una visión completa y práctica de OpenMP (www.openmp.org), un conjunto de directivas estándar para la programación paralela de supercomputadores de memoria compartida.

Temario:

1. Introducción a la programación paralela:
 - Niveles de paralelismo.
 - Medidas de rendimiento.
 - Tipos de arquitecturas paralelas.
 - Paradigmas de programación paralela.
 - Programación paralela en el SVGD.
 - Compilación y ejecución de programas en el SVGD
2. Especificación OpenMP:
 - Introducción.
 - Compilación y ejecución de programas OpenMP.
 - Características principales del estándar OpenMP.
 - Directivas para la construcción de paralelismo.
 - Directiva THREADPRIVATE.
 - Directivas de sincronización.
 - Biblioteca de rutinas OpenMP.
 - Variables de entorno
3. Paralelización a nivel de lazo mediante OpenMP:
 - Pasos en la paralelización de un programa.
 - Lazos potencialmente paralelos.
 - Técnicas de reestructuración de código.
 - Consideraciones caché.
 - Sobrecarga de la paralelización.
 - Casos de estudio.



- Análisis de eficiencia

4. Más información:

- Benchmarking.
- Otros compiladores.
- OpenMP 3.0.
- Directiva task.
- Cláusula collapse.
- Nuevo soporte para el anidamiento de paralelismo.
- Extensión de las formas de planificar un lazo.
- Control portable de threads.
- Referencias



Herramientas de desarrollo de aplicaciones paralelas: Debugging y análisis del rendimiento

Del 7 al 11 de septiembre

Debido al incremento del paralelismo implícito en el desarrollo de los actuales procesadores, la detección de fallos y la ejecución eficiente de los códigos paralelos se han convertido en un objetivo fundamental para satisfacer la creciente demanda de potencia de cómputo. Existen herramientas de análisis de rendimiento y debugging que ayudan a detectar los problemas existentes y proporcionan la información necesaria para resolverlos. En este curso se proporcionará una introducción a los conceptos de debugging y análisis de rendimiento de aplicaciones paralelas, mostrando al usuario las diferentes técnicas utilizadas. Así mismo, se mostrará el uso de herramientas representativas para mostrar las capacidades de este tipo de aproximaciones. Con los conocimientos adquiridos, el alumno/a tendrá la capacidad para depurar correctamente una aplicación paralela, así como para realizar un análisis del rendimiento e interpretar los resultados obtenidos.

Temario:

1. Visión general y organización del curso.
2. Debugging de aplicaciones paralelas:
 - Aproximación sistemática al proceso de debugging.
 - Tipos de fallos de aplicaciones: fallos comunes, problemas de difícil detección, fallos de memoria.
 - Problemas específicos de aplicaciones paralelas: race conditions, interbloqueo.
 - Uso de herramientas de debugging en aplicaciones paralelas: debugging de código, de memoria y multithread.
 - Ejemplos prácticos.
2. Análisis de rendimiento de aplicaciones paralelas:
 - Análisis de rendimiento: motivaciones y concepto.
 - Técnicas de instrumentación.
 - Técnicas de medida y análisis de rendimiento.
 - Herramientas para análisis de rendimiento de aplicaciones paralelas.
 - Ejemplos prácticos