

CESGA Computational Science Summer School 2010



Del 21 de junio al 15 de septiembre de 2010
Centro de Supercomputación de Galicia, Santiago de Compostela.
Inscripción: 1-31 Marzo 2010

CURSOS

FECHAS

Inauguración/ Acceso y utilización de recursos (FinisTerae, SVGD) en el CESGA	21/06
Programación en Fortran	22/06-25/06
Programación en C	28/06-02/07
Matemática Computacional	05/07-09/07
Introducción a algoritmos de resolución de aplicaciones científicas	12/07-16/07
Programación paralela utilizando directivas OpenMP	19/07-22/07
Introducción a la programación en MPI	26/07-29/07
Herramientas de desarrollo de aplicaciones paralelas	02/08-05/08
Realización de prácticas/ Entrega	09/08-15/09



CESGA Computational Science Summer School 2010

Índice:

1	Introducción.....	2
2	Objetivos.....	2
3	Destinatarios.....	3
4	Fechas y lugar de celebración.....	3
5	Organización y financiación.....	4
6	Formación.....	5
7	Calendario.....	7
8	Solicitud de plaza.....	8
9	Programas de los cursos de formación.....	9

1 Introducción

Hoy en día, gran parte de la investigación científica implica el uso de ordenadores tanto como la teoría o la experimentación. La computación muchas veces supone una ayuda fundamental para la comprensión y profundización en el problema abordado, que no sería posible mediante otras herramientas. Con los ordenadores, los científicos pueden investigar problemas que antes eran imposibles de abordar por enormemente complejos, peligrosos, o por su consumo excesivo de tiempo, y obtener resultados casi instantáneamente.

La Ciencia Computacional es una disciplina científica interdisciplinar orientada a la solución de problemas científicos complejos mediante la utilización de modelos matemáticos, simulación numérica y computación.

El Centro de Supercomputación de Galicia (CESGA) es el centro de cálculo, comunicaciones de altas prestaciones y servicios avanzados de la Comunidad Científica Gallega, sistema académico universitario y del Consejo Superior de Investigaciones Científicas (CSIC). Es uno de los centros de referencia en el área de Ciencia Computacional a nivel europeo, y cuenta entre sus máquinas con uno de los mayores supercomputadores de memoria compartida en Europa, "Finis Terrae".

2 Objetivos

La "CESGA Computational Science Summer School 2010" pretende fomentar la actividad computacional dentro de la comunidad investigadora, especialmente en la resolución de problemas con grandes necesidades de computación. Se compone tanto de tareas formativas grupales y tutorizadas como de actividades de investigación voluntarias. Además, la formación se complementará con la presentación de aplicaciones avanzadas de la ciencia computacional en otras áreas como Matemáticas, Biomedicina, Nanotecnología, etc.

Los objetivos de la CESGA Computational Science Summer School 2010 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.



3 Destinatarios

La "CESGA Computational Science Summer School 2010" 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.

Hay un máximo de **12 plazas** disponibles para esta segunda edición de 2010.

4 Fechas y lugar de celebración

Fechas: Del 21 de junio al 15 de septiembre de 2010.

Lugar: Centro de Supercomputación de Galicia (CESGA), Av. de Vigo, s/n, Campus Sur, Santiago de Compostela (A Coruña), España.

Santiago de Compostela es una acogedora ciudad con más de 500 años de tradición universitaria. Declarada "Patrimonio de la Humanidad" en 1985 por la UNESCO, por su belleza urbana y su importancia como destino de una de las rutas culturales y espirituales más importantes del mundo, el Camino de Santiago. Este año, además, cuenta con una significación especial, por ser Año Santo, y la ciudad tiene planificadas innumerables actividades culturales y de ocio durante el año 2010.

5 Organización y financiación

Organiza: Centro de Supercomputación de Galicia

Contacto: ccsss10@cesga.es

Comité organizador:

- María Teresa Sánchez Rúa, Centro de Supercomputación de Galicia (Directora).
- Fernando Bouzas Sierra, Centro de Supercomputación de Galicia.
- Andrés Gómez Tato, Centro de Supercomputación de Galicia.
- María José Rodríguez Malmierca, Centro de Supercomputación de Galicia, CESGA.
- Juan Touriño Domínguez, Departamento de Electrónica y Sistemas, Universidad de A Coruña.
- Francisco Fernández Rivera, Departamento de Electrónica y Computación, Universidad de Santiago de Compostela.

Colabora:

- Nodo CESGA del Proyecto i-MATH.

6 Formación

Contenido de los cursos:

La parte formativa de la "CESGA Computational Science Summer School 2010" está compuesta por siete cursos orientados a proporcionar 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 (Finis Terrae, SVGD) en el CESGA	21/06	INTRO
Programación en Fortran	22/06-25/06	PRF
Programación en C	28/06-02/07	PRC
Matemática Computacional: Compilación, ejecución y optimización de programas	05/07-09/07	MTC
Introducción a algoritmos de resolución de aplicaciones científicas	12/07-16/07	ALG
Programación paralela utilizando directivas OpenMP	19/07-22/07	OMP
Introducción a la programación en MPI	26/07-29/07	MPI
Herramientas de desarrollo de aplicaciones paralelas: Debugging y análisis del rendimiento	02/08-05/08	DEB

Adicionalmente a los cursos de formación, se celebrarán seminarios temáticos de aplicación de soluciones computacionales en diversas áreas científicas, impartidos por investigadores especialistas de prestigio internacional.

Formato de los cursos:

- Presencial, de 20 horas teórico/prácticas en sesiones de 4/5 horas (más descansos), con acceso adicional a una plataforma de e-learning para facilitar la interacción e intercambio de materiales.
- Cada alumno contará con una copia en papel y en formato electrónico de la documentación y ejercicios a realizar.
- Tutorías personalizadas realizadas por el profesorado y personal técnico del CESGA.
- Se dispondrá de un PC para cada alumno. Opcionalmente, se podrá trabajar con un portátil propio.
- Además, dispondrán de cuentas de usuario en CESGA durante la celebración de la escuela para acceder a los distintos recursos del centro.

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 CESGA.

<i>Nombre del curso</i>	<i>Profesor(es)</i>
Acceso y utilización de recursos en el CESGA (INTRO)	Carlos Fernández Sánchez (CESGA)
Programación en Fortran (PRF)	Juan Touriño Domínguez (UDC)
Programación en C (PRC)	Juan Ángel Lorenzo del Castillo (USC)
Matemática Computacional (MTC)	Patricia González Gómez (UDC)
Introducción a algoritmos de resolución de aplicaciones científicas (ALG)	María Teresa Sánchez Rúa (CESGA)
Programación paralela utilizando directivas OpenMP (OMP)	María José Martín Santamaría (UDC)
Introducción a la programación en MPI (MPI)	Xulio López Albín (USC)
Herramientas de desarrollo de aplicaciones paralelas (DEB)	Gabriel Rodríguez Álvarez (UDC) Diego Rodríguez Martínez (USC)

7 Calendario

Lunes		Martes		Miércoles		Jueves		Viernes	
Junio 2010									
21:	Presentación	22:	PRF	23:	PRF	24:	PRF	25:	PRF
	INTRO								
28:	PRC	29:	PRC	30:	PRC	01:	PRC	02:	PRC
Julio 2010									
28:	PRC	29:	PRC	30:	PRC	01:	PRC	02:	PRC
05:	MTC	06:	MTC	07:	MTC	08:	MTC	09:	MTC
12:	ALG	13:	ALG	14:	ALG	15:	ALG	16:	ALG
19:	OMP	20:	OMP	21:	OMP	22:	OMP	23:	
26:	MPI	27:	MPI	28:	MPI	29:	MPI	30:	
Agosto 2010									
02:	DEB	03:	DEB	04:	DEB	05:	DEB	06:	
Septiembre 2010									
13:		14:		15:	Fecha final entrega prácticas				

8 Solicitud de plaza

La "CESGA Computational Science Summer School 2010" cuenta con 12 plazas. La inscripción se podrá realizar del **1 al 31 de marzo de 2010**, a través de la página web <http://www.cesga.es/SummerSchool2010>.

La concesión de plazas se realizará por parte del Comité Organizador. Se notificará la concesión de plaza a los solicitantes antes del **15 de abril de 2010**.

Cuotas de inscripción:

	<i>Cuota de inscripción</i>
Investigador i-MATH	700€
Investigador Univ. Gallegas / CSIC	700€
Investigador Red de Centros Tecnológicos	700€
Otros	1750€

La cuota de inscripción incluye todas las actividades formativas y 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 y entregar las prácticas correspondientes a cada uno de los cursos antes del 15 de septiembre de 2010.

Observaciones:

- La cuota de inscripción a la CESGA Computational Science Summer School 2010 será gratuita para los beneficiarios de ayudas de movilidad para accesos a ICTS que soliciten una estancia en el CESGA del 5 de julio al 6 de agosto de 2010. Más información en <http://icts.cesga.es>.
- Los alumnos tienen la posibilidad de realizar una estancia de investigación, no remunerada, en el CESGA durante la celebración de la Summer School.
- Para los miembros del proyecto Ingenio Mathematica, se oferta una estancia de investigación durante la celebración de la Summer School. Más información en la página web: http://mathematica.nodo.cesga.es/estancia_CCSSS10.

9 Programas de los cursos de formación

Programación en Fortran:

Del 22 al 25 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 28 de junio al 2 de julio

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 5 al 9 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:

Del 12 al 16 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.

Programación paralela utilizando directivas OpenMP:

Del 19 al 22 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 FT
 - Compilación y ejecución de programas en el FT
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



CESGA Computational Science Summer School 2010

- Análisis de eficiencia

4. Más información

- Benchmarking
- Otros compiladores
- Referencias

Introducción a la programación en MPI:

Del 26 al 29 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

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

Del 2 al 5 de agosto

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