

Información General

Conocimientos Previos:

Fundamentos de sistemas operativos y arquitectura de ordenadores. Conocimientos de programación C,

Horarios:

Semana 0: Miércoles 20, jueves 21 y viernes 22 de junio de 2012, de 9 a 18 hs.

Semana 1: Jueves 28 y viernes 29 de junio de 2012, de 9 a 18 hs.

Semana II: Jueves 5 y viernes 6 de julio de 2012, de 9 a 18 hs.

Matrícula:

Semana 0: 900 €.

Semana 1: 800 € (si asiste a semana 0: 640€).

Semana 2: 800 € (si asiste a semana 0 ó I 640€, si asiste a semana 0 y I: 600 €).

El precio incluye el material del curso, comidas y cafés.

Forma de pago:

Una vez recibida la confirmación de la reserva de plaza, se le enviará una hoja de inscripción para pago por tarjeta de crédito o transferencia bancaria. Información adicional sobre facturación y pagos: facturacion@electratraining.org

Lugar:

Escuela Politécnica Superior de la Universidad Autónoma de Madrid, Ctra. de Colmenar Km.15, 28049 Madrid. Puede encontrar información sobre como llegar a la EPS UAM en: <http://www.electratraining.org/contactar/como-llegar/>

La Escuela se encuentra comunicada por autovía (M40 + M607 Ctra. de Colmenar), Autobuses y Trenes de Cercanías. No existen problemas de aparcamiento en la zona.

Reserva de Plaza:

El curso tiene un cupo limitado a 16 asistentes. Puede reservar plaza vía la página web del curso (www.electratraining.org) o enviando los siguientes datos por correo electrónico.

Nombre y Apellido:

Empresa:

Teléfono de contacto:

e-mail:

Fax:

Página www:

Reservar de Plaza semana 0, I, 2 (tachar lo que no corresponda)



Información Adicional:

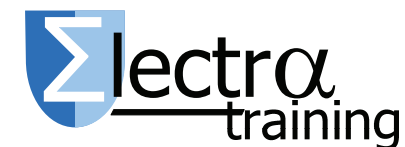
Gustavo Sutter (609 69 16 07)

Ctra. de Colmenar Km.15
28049 Madrid

Tlf: 91 497 6209

Correo: info@electratraining.org ó gustavo.sutter@uam.es

Internet: <http://www.electratraining.org>



Computación Paralela: MPI, OpenMP, GPUs

Ciclo 2012 de
Entrenamiento Intensivo en
Diseño de Sistemas Digitales



Escuela Politécnica Superior
Universidad Autónoma de Madrid



Curso 2012: Computación Paralela, MPI, OpenMP, GPUs

Presentación

Este curso se dirige a diseñadores, desarrolladores y directores de proyecto relacionados con sistemas de altas prestaciones. Dentro del amplio espectro de la programación paralela, el temario está centrado en la programación con MPI, OpenMP y CUDA.

El curso es eminentemente práctico. El trabajo de Laboratorio (un asistente por PC) constituye cerca del 60 % del total.

Por flexibilidad, el temario se ha separado en tres partes independientes. En la Semana 0, se presentan las nociones básicas de la programación paralela tanto a nivel arquitectural como de lenguajes de programación. Luego, en la Semana 1 se centra exclusivamente en los temas avanzados de la programación de GPGPU con CUDA. En la Semana 2 se concentran los temas avanzados de programación con MPI/OPENMP.

Los descriptores del curso para cada bloque son:

Semana 0: Computación paralela MPI, OpenMP, GPUs (3 días -24hs).

Objetivos y Metodología. Motivación y Aplicaciones. Desafíos y Claves de la programación paralela. Clasificación de Arquitecturas: Clasificación de Flynn y por Acceso a Memoria e Interconexión. Tipos de Paralelismo. Modelos de Programación. Actualidad y Futuro: Top 500.

Análisis de prestaciones en sistemas de computación paralela. Tiempo de ejecución, Aceleración (Speedup). Métricas de ren-

dimiento. Modelos de trabajo en Sistemas paralelos. Tiempo Fijo vs Carga de trabajo Fijo. Ley de Amdahl. Ley de Gustafson. Grado de paralelismo. Condiciones para abordar un programa en paralelo.

Programación de sistemas con paso de mensajes: Introducción a MPI. Paradigma de paso mensajes. Principios y Características. Máquina Virtual Paralela. Introducción a MPI (Message Passing Interface). Conceptos y Funciones Básicas. Modelos de Programación con MPI. Paradigma Maestro/Escavo. Modos de comunicación: Punto a Punto; Colectivas. Tipos de datos derivados. Topologías

Programación de sistemas con memoria compartida: OpenMP. Programación paralela basada en directivas. Ejecución con múltiples Threads. Variables de entorno. Dependencia de datos e Interferencia. Variables privadas. Reducción. Planificación y Transformación de bucles. Rendimiento en OpenMP

Programación paralela con procesadores gráficos GPU. Sistemas con coprocesadores gráficos GPU. Paralelismo SPMD en una arquitectura con múltiples cores. Arquitectura y Modelo de Ejecución. Programación de propósito general con GPU (GPGPU). Entorno de programación CUDA. Modelo de Programación. API de CUDA. Ejemplos de aplicación.

Semana 1: Programación Avanzada GPGPU en CUDA (2 días - 16 hs).

Introducción: GPUs como sistemas de computación paralela. Arquitectura de las GPUs actuales. Modelos y lenguajes de programación

Introducción a CUDA. Paralelismo de datos. Estructura de un programa en CUDA. Declaración de funciones. Kernels. API RunTime.

Threads en CUDA: Organización y planificación de los threads, Sincronización y escalabilidad.

Memorias en CUDA: Tipos de memorias, Eficiencia de acceso a memoria.

Consideraciones de rendimiento: Ejecución, Ancho de banda, Pre-captura de datos, Granularidad, Medición de rendimiento

Semana 2: Programación Avanzada de MPI/OPENMP (2 días - 16 hs).

Paradigmas de programación paralela.

Paralelización desde las opciones de compilación: Autoparalelización, Desenrollamiento de bucles, Instrucciones SSE, Optimizaciones especulativas.

Programación para sistemas con memoria compartida: OpenMP. Paralelismo basado en Threads. Directivas para la construcción de paralelismo y de sincronización. Biblioteca de rutinas OpenMP. Variables de entorno. Herramientas de monitorización TAU. Evaluación de prestaciones

Análisis de Algoritmos paralelos y revisión de librerías de cálculo científico: Algoritmos para ejecución en memoria compartida y por paso de mensajes.

Programación con el paradigma de paso de mensajes MPI: Características generales. Tipo Comunicaciones. Tipos de datos derivados. Empaquetado de datos. Topologías virtuales y comunicadores. Esquemas de balanceo de carga. Evaluación de prestaciones. Nuevas funcionalidades de MPI-2. Programación híbrida: MPI + OpenMP

Herramientas de profiling: Valgrind. Evaluación dinámica de prestaciones. Acceso a contadores.

Programación en sistemas computacionales de altas prestaciones: en multicomputadores/multiprocesador y en cluster. Computación heterogénea.