

Temario Preliminar

Curso de Programación Paralela: MPI, OpenMP y GPUs

El temario pretende abordar los diferentes aspectos involucrados en el desarrollo de aplicaciones paralelas, desde una revisión a las diferentes arquitecturas disponibles hasta las metodologías de diseño y herramientas asociadas a la programación de aplicaciones para su ejecución en paralelo.



Este curso se dirige a diseñadores, desarrolladores y directores de proyecto relacionados con sistemas de altas prestaciones. El temario está centrado en la programación con MPI, OpenMP y CUDA.

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. 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.

- [Semana 0](#): Computación paralela MPI, OpenMP, GPUs (3 días -24hs).
- [Semana 1](#): Programación Avanzada GPGPU en CUDA (2 días – 16 hs).
- [Semana 2](#): Programación Avanzada MPI/OPENMP (2 días – 16 hs).

La información actualizada de estos cursos en www.electratraining.org

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 para procesamiento en paralelo.
 - Clasificación de Flynn
 - Clasificación 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 rendimiento.
- 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/Esclavo
 - Modos de comunicación
 - Comunicaciones Punto a Punto
 - Comunicaciones Colectivas.
 - Tipos de datos derivados
 - Topologías

Programación de sistemas con memoria compartida: Modelo de programación 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 muchos 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).

1.- Introducción

- GPUs como sistemas de computación paralela
- Arquitectura de las GPUs actuales
- Modelos y lenguajes de programación

2- Introducción a CUDA

- Paralelismo de datos
- Estructura de un programa en CUDA
- Declaración de funciones
- Kernels
- API RunTime

3- Threads en CUDA

- Organización de los threads
- Sincronización y escalabilidad
- Planificación de threads

4- Memorias en CUDA

- Tipos de memorias
- Eficiencia de acceso a memoria

5- Consideraciones de rendimiento

- Ejecución
- Ancho de banda
- Pre-captura de datos
- Granularidad
- Medición de rendimiento

Semana 2: Programación Avanzada 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
- Directivas 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
- Algoritmos para ejecución por paso de mensajes

Programación con el paradigma de paso de mensajes: MPI

- Características generales de MPI
- Comunicaciones punto a punto
- Comunicaciones colectivas
- 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 los contadores.

Programación en sistemas computacionales de altas prestaciones

- Computación en sistemas multicomputadores/multiprocesador.
- Computación en cluster
- Computación heterogénea