UNIVERSIDAD CENTRAL DE VENEZUELA
Facultad de Ciencias
Escuela de Computación
Licenciatura en Computación

Plan de Estudios 2000
Componente Profesional
Opción: Sistemas Paralelos y Distribuídos
Actualización: Noviembre 2014 v1.0

Computación de Alto Rendimiento (CAR)

UC	HT	HP	HL	Modalidad	Código	Prerequisitos
5	4	2	0	Electiva	6543	Lenguajes de Programación y Comunicación de Datos

1. Motivación.

El auge tecnológico y los relativos bajos costos de los sistemas computacionales basados en procesadores de varios y muchos núcleos, así como los Clústers de cómputo, Grids, etc., ha hecho que el usuario común tenga a su alcance la tecnología idónea para resolver problemas complejos en diversas disciplinas del ámbito científico, industrial y organizacional. En este contexto, la Computación de Alto Rendimiento (CAR) es el área tecnológica que aborda las herramientas, métodos y técnicas secuenciales y paralelas, en hardware y software, que permiten acelerar soluciones de cómputo intensivo que resuelven problemas complejos. La CAR busca explotar eficientemente los recursos computacionales empleados y obtener drásticas reducciones de los tiempos de ejecución de los programas sin sacrificar la confiabilidad de los resultados.

De allí que el dominio y comprensión de los fundamentos de la arquitectura del computador y del procesador, de técnicas de programación secuencial y paralela, de optimización de software, entre otras; así como destrezas prácticas que permitan aprovechar eficientemente los recursos computacionales y obtener soluciones de alto desempeo es esencial en la formación del licenciado en computación. El propósito de esta asignatura es familiarizar al estudiante con conceptos, técnicas, métodos, tecnologías y herramientas específicas en el área de la optimización secuencial y paralela, en hardware y software, para acelerar la ejecucion de programas que resuelven problemas complejos.

2. Objetivos.

Al finalizar el curso el estudiante será capaz de:

- Conocer y describir los fundamentos teóricos y aspectos tecnológicos asociados a la computación de alto rendimiento.
- Conocer las áreas y problemas susceptibles de ser solucionados mediante la computación de alto rendimiento.
- Conocer, identificar y describir los factores que limitan el desempeño así como las herramientas para su análisis.
- Conocer y aplicar técnicas de optimización secuencial y paralela, en hardware y software, que mejoran el desempeño.
- Analizar problemas de cómputo intensivo y describir soluciones de alto desempeño aplicando técnicas secuenciales y paralelas.

3. Contenido temático.

TEMA 0. CHARLA INTRODUCTORIA: ▪ **0.1.** Presentacion del Profesor ▪ **0.2.** Presentación de la asignatura: entrega de la nota informativa, nombre de la materia y su significado ▪ **0.3.** Objetivos de la asignatura y su necesidad en la carrera ▪ **0.4.** Contenido de la asignatura ▪ **0.5.** Criterios de evaluación ▪ **0.6.** Bibliografia a usar ▪ **0.7.** Página web de la asignatura.

TEMA 1. INTRODUCCIÓN AL RENDIMIENTO DE COMPUTADORES: ▪ **1.1.** Introducción: Definición, motivación, objetivos y alcances de la Computación de Alto Rendimiento. Aspectos tecnológicos asociados a los sistemas computacionales de alto rendimiento. Áreas y problemas de aplicación. ▪ **1.2.** Conceptos básicos: Concepto de Problema, Algoritmo y Programa. ▪ **1.3.** Arquitectura del Computador: Modelo de Von Neumann. Modelo de Computación. Modelo de Memoria. Modelo de Programación. Clasificación de los Sistemas Computacionales según la Taxonomía

de Flynn (SISD, SIMD, MISD y MIMD). Otras propuestas de clasificación. ▪ **1.4.** Medición del rendimiento: Concepto de rendimiento. Rendimiento del CPU. Métricas de rendimiento (MIPS, MFLOPS). Programas para evaluar el rendimiento. Medición del tiempo de ejecución de programas. ▪ **1.5.** Cálculo del rendimiento: Ley de Amdahl. Aplicación de la Ley de Amdahl para el caso secuencial, paralelo y pipeline. ▪ **1.6.** Ejemplo práctico: Uso de programas de evaluación de dominio público. ▪ **1.7.**
Texto 1: ***Jhon Hennessy y David Patterson,*** **"Arquitectura de Computadores: Un Enfoque Cuantitativo".** Mc-Graw Hill. 1era. Edición, 1993.
Texto 2: ***David Patterson y Jhon Hennessy,*** **"Estructura y Diseño de Computadores: La Interfaz Hardware/Software".** Mc-Graw Hill. 4ta. Edición, 2011.

TEMA 2. ESTRATEGIAS PARA LA OPTIMIZACIÓN DE CÓDIGO SECUENCIAL: ▪ **2.1.** Introducción: Etapas en el Proceso de generación de código ejecutable. ▪ **2.2.** Mecanismos de optimización mediante hardware: Elementos de la Arquitectura del Procesador. Ciclo de Ejecución de Instrucciones. Pipeline de datos e instrucciones. Unidades Funcionales. Procesadores Superescalares. Procesadores Multinúcleo. Ejemplo práctico 1: Aplicacin de la Ley de Amdahl para el caso secuencial y pipeline. Jerarquía de Memoria: Registros, Cache, Memoria Principal y Secundaria. Localidad de memoria: temporal y espacial. Ejemplo práctico: Localidad de datos e instrucciones. ▪ **2.3.** Técnicas de Optimización mediante Software: Rol del compilador. Reglas de optimización manual (programador) y automática (compilador). Banderas (Flags) y Directivas del Compilador para la optimización y depuración de programas. Ejemplo práctico 2: Optimización de código de un programa mediante flags del compilador (compilador gcc). Perfil de ejecución (Profiling) de programas. Ejemplo práctico 3: Perfilando programas usando la herramienta de dominio público (gprof). ▪ **2.4.** Subtema 4 ▪ **2.5.** Subtema 5 ▪ **2.6.** Subtema 6 ▪ **2.7.** Subtema 7
Texto 1: ***Georg Hager and Gerhard Wellein,*** **"Introduction to high performance computing for scientists and engineers".** CRC Press. First Edition, 2011.

TEMA 3. FUNDAMENTOS DE PARALELISMO: ▪ **3.1.** Conceptos básicos: Concurrencia Real y Virtual. Paralelismo de Datos. Paralelismo de Control. Granularidad del paralelismo. Balance de procesamiento versus comunicación. Modelos de Computación Paralela. Redes de interconexión. ▪ **3.2.** Taxonomía de los Sistemas de Computación Paralela: Sistemas de Computación Paralela con Memoria Compartida o UMA. Sistemas de Computación Paralela con Memoria Distribuída o NUMA. Sistemas de Computación Paralela con Memoria Distribuída-Compartida o ccNUMA. Ejemplo práctico 1: Comparar el tiempo de ejecución de un programa corriendo en un núcleo versus varios núcleos. ▪ **3.3.** Programación paralela: Principios de diseño de algoritmos paralelos. Modelos de algoritmos paralelos. Metodología de desarrollo de programas paralelos. Ejemplo práctico 2: Programa paralelo ejemplo. ▪ **3.4.** Subtema 4 ▪ **3.5.** Subtema 5 ▪ **3.6.** Subtema 6 ▪ **3.7.** Subtema 7
Texto 1: ***Georg Hager and Gerhard Wellein,*** **"Introduction to high performance computing for scientists and engineers".** CRC Press. First Edition, 2011.

TEMA 4. PROGRAMACIÓN PARALELA USANDO MEMORIA COMPARTIDA: ▪ **4.1.** Sistemas de Computación Paralela con Memoria Compartida o UMA: Arquitectura. Coherencia de cache. Procesadores Multi-núcleos y Muchosnúcleos (GPUs). ▪ **4.2.** Lenguajes con soporte para la programación paralela usando memoria compartida: Directivas OpenMP, Librerías OpenCL/CUDA. Ejemplo práctico: Multiplicación de matrices en paralelo usando OpenMP, OpenCL/CUDA y Threads (Hilos). ▪ **4.3.** Subtema 3 ▪ **4.4.** Subtema 4 ▪ **4.5.** Subtema 5 ▪ **4.6.** Subtema 6 ▪ **4.7.** Subtema 7
Texto 1: ***Georg Hager and Gerhard Wellein,*** **"Introduction to high performance computing for scientists and engineers".** CRC Press. First Edition, 2011.

TEMA 5. PROGRAMACIÓN PARALELA USANDO MEMORIA DISTRIBUÍDA: ▪ **5.1.** Sistemas de Computación Paralela con Memoria Distribuída o NUMA: Arquitectura. Clúster de Computadores. ▪ **5.2.** Lenguajes con soporte para la programación paralela usando memoria distribuída: Librerías OpenMPI: Ejemplo práctico 1: Multiplicación de matrices en paralelo usando OpenMPI. Ejemplo práctico 2: Instalación del ?Clúster de un hombre pobre?. ▪ **5.3.** Sistemas de Computación Paralela con Memoria Distribuída-Compartida: Arquitectura. Clúster híbridos (paralelismo local UMA y global NUMA). Ejemplo práctico 3: Multiplicación de matrices en paralelo usando memoria distribuída-compartida en OpenMPI/OpenMP. ▪ **5.4.** Subtema 4 ▪ **5.5.** Subtema 5 ▪ **5.6.** Subtema 6 ▪ **5.7.** Subtema 7
Texto 1: ***Georg Hager and Gerhard Wellein,*** **"Introduction to high performance computing for scientists and engineers".** CRC Press. First Edition, 2011.

TEMA 6. SOLUCIONES COMUNES A PROBLEMAS USANDO CÓMPUTO DE ALTO RENDIMIENTO: ▪ **6.1.** Aspectos comunes en problemas de cómputo intensivo: Organización del cómputo o Procesamiento. Segmentación y distri-

bución de los datos. Formas de almacenamiento de los datos. ▪ **6.2.** Problemas comunes en: Dinámica de fluídos computacional. Dinamiza Molecular. Algebra lineal Numérica. Energía y ambiente. Modelado oceánico. Simulación Sísmica. Señales y procesamiento de imágenes y video. Otros. ▪ **6.3.** Aspectos a considerar en el Desarrollo de Software Numérico. ▪ **6.4.** Subtema 4 ▪ **6.5.** Subtema 5 ▪ **6.6.** Subtema 6 ▪ **6.7.** Subtema 7
Texto 1: ***Georg Hager and Gerhard Wellein,*** **"Introduction to high performance computing for scientists and engineers".** CRC Press. First Edition, 2011.

4. Criterios de Evaluación.

Tareas	Exámenes	Proyectos	Exposición	Total
2 x 10 %	aula:15 %, casa:10 %	openMP:15 %, MPI:15 %, CUDA:15 %	1 x 10 %	100 %

5. Horarios de Clase.

Tipo	Dia	Hora	Lugar
Teoría	Lunes y Miercoles	9-11am	Aula Postgrado, Piso 1, Escuela de Computación

6. Grupo Docente.

Profesor	Rol	Seccion	email	Centro
Carlos Acosta	Coordinador	Única	ccpd.ciens.ucv.edu@gmail.com	CCPD

8. Observaciones generales.

- Criterios de entrega de tareas y proyectos:

9. Planificación de clases.

Calendario de Clases, semestre lectivo: 2-2012.

Semanas	Clases	Fechas	Contenidos
Semana 1	Clase 0	Martes 02-10-2012	Tema 0: Charla Introductoria
	Clase 1	Jueves 04-10-2012	Tema 1: 1.1. Introducción. 1.2. Conceptos básicos.
Semana 2	Clase 2	Martes 09-10-2012	Tema 1: 1.3. Arquitectura del Computador.
	Clase 3	Jueves 11-10-2012	Tema 1: 1.4. Medición del rendimiento.
Semana 3	Clase 4	Martes 16-10-2012	Tema 1: 1.5. Cálculo del rendimiento.
	Clase 5	Jueves 18-10-2012	Tema 1: 1.6. Ejemplo práctico.
Semana 4	Clase 6	Martes 23-10-2012	Tema 2: 2.1. Introducción.
	Clase 7	Jueves 25-10-2012	Tema 2: 2.1. Introducción.
Semana 5	Clase 8	Martes 30-10-2012	Tema 2: 2.1. Introducción.
	Clase 9	Jueves 01-11-2012	Tema 2: 2.1. Introducción.
Semana 6	Clase 10	Martes 06-11-2012	Tema 2: 2.1. Introducción.
	Clase 11	Jueves 08-10-2012	Tema 2: 2.1. Introducción.
Semana 7	Clase 12	Martes 13-11-2012	Tema 2: 2.1. Introducción.
	Clase 13	Jueves 15-11-2012	Tema 2: 2.1. Introducción.
Semana 8	Clase 14	Martes 20-11-2012	Tema 2: 2.1. Introducción.
	Clase 15	Jueves 22-11-2012	Tema 2: 2.1. Introducción.
Semana 9	Clase 16	Martes 27-11-2012	Tema 2: 2.1. Introducción.
	Clase 17	Jueves 29-11-2012	Tema 2: 2.1. Introducción.
Semana 10	Clase 18	Martes 04-12-2012	Tema 2: 2.1. Introducción.
	Clase 19	Jueves 06-12-2012	Tema 2: 2.1. Introducción.
Semana 11	Clase 20	Martes 11-12-2012	Tema 2: 2.1. Introducción.
	Clase 21	Jueves 13-12-2012	Tema 2: 2.1. Introducción.
Semanas	No clases	Lunes 17-12-2012	Inicio de vacaciones de Diciembre
	No clases	Lunes 08-01-2013	Fin de vacaciones de Diciembre
Semana 12	Clase 22	Martes 08-01-2013	Tema 2: 2.1. Introducción.
	Clase 23	Jueves 10-01-2013	Tema 2: 2.1. Introducción.
Semana 13	Clase 24	Martes 15-01-2013	Tema 2: 2.1. Introducción.
	Clase 25	Jueves 17-01-2013	Tema 2: 2.1. Introducción.
Semana 14	Clase 26	Martes 22-01-2013	Tema 2: 2.1. Introducción.
	Clase 27	Jueves 24-01-2013	Tema 2: 2.1. Introducción.
Semana 15	Clase 28	Martes 29-01-2013	Tema 2: 2.1. Introducción.
	Clase 29	Jueves 31-01-2013	Tema 2: 2.1. Introducción.
Semana 16	Clase 30	Martes 05-02-2013	Tema 2: 2.1. Introducción.
	Clase 31	Jueves 07-02-2013	Tema 2: 2.1. Introducción.