TALLER de PROGRAMACIÓN sobre GPUs

Año 2015

Carrera/ Plan:
Licenciatura en Sistemas
Planes 2003/07/12/15
Licenciatura en Informática
Planes 2003/07/12/15

Año: Optativa

Regimen de Cursada: *Semestral*

Carácter: Optativa

Correlativas: Programación Concurrente

Profesor: Adrian Pousa

Docentes Auxiliares: Victoria Sanz

Hs. semanales: 9 hs.

FUNDAMENTACIÓN

Los procesadores gráficos (GPUs) han surgido como una alternativa dentro de los procesadores con múltiples núcleos, por sus características de perfomance y consumo energético.

El uso de GPUs, tanto en computación de alto desempeño como en aplicaciones de propósito general comienza a ser una alternativa de bajo costo para el desarrollo de aplicaciones de muy alto rendimiento que tradicionalmente han sido exclusivas de los clusters de multicores y supercomputadoras.

En este contexto, la metodología e implementación de aplicaciones es un tema de gran interés actual.

Son objetivos de este curso: profundizar el conocimiento de las arquitecturas tipo GPU y su programación, comparar su perfomance con los multicores convencionales, analizar los modelos de resolución de problemas específicos e introducir conceptos de consumo y green computing a partir de la utilización de GPUs.

OBJETIVOS GENERALES:

Son objetivos de este curso:

- Profundizar el conocimiento de las arquitecturas tipo GPU y su programación.
- Comparar su perfomance con los multicores convencionales.
- Analizar los modelos de resolución de problemas específicos.
- Introducir conceptos de consumo y green computing a partir de la utilización de GPUs.

CONTENIDOS MINIMOS:

- GPU: Introducción a GPGPU
- Arquitecturas GPU - Modelo GPU-CPU.
- Modelo y jerarquía de Memoria de GPU.
- Modelo de Programación GPU - Resolución de aplicaciones.
- Medidas de performance y consumo en GPU.
- Multi-GPUs y Arquitecturas Híbridas.

Programa Analítico

Unidad 1: GPU: Introducción a HPC y GPGPU

- Introducción al cómputo de altas prestaciones (HPC).
- Paradigmas de Computación Paralela en GPUs: Modelo de Memoria Compartida, Modelo de Memoria Distribuida. Paralelismo de Datos y Paralelismo Funcional.
- Taxonomía de Flynn.
- Arquitecturas y herramientas para HPC.
- Introducción a la arquitectura GPU y su uso en HPC.
- GPGPU: Computación de Propósito General en GPU.

Unidad 2: Arquitecturas GPU - Modelo GPU-CPU

- Evolución de las GPUs.
- Arquitecturas Nvidia.
- Arquitecturas ATI-AMD.
- Arquitecturas Xeon-Phi.
- Modelo de interacción GPU-CPU.
- Introducción a la planificación de hilos en GPU Nvidia - Concepto de Grid, Bloque, Thread y Warp.
- Rendimiento y consumo de las arquitecturas GPU según Top500 y Green500.

Unidad 3: Modelo de Programación GPU - Resolución de aplicaciones

- Modelo de programación en GPU.
- Relación con SIMD, modelo SIMT.
- Modelo de programación CUDA.
    - Concepto de Host y Device. Identificadores.
    - Tipos de datos.
    - Definición de Constantes.
    - Variables: alcance y tiempo de vida.
    - Gestión de memoria, copia explicita CPU-GPU y GPU-CPU, Síncrona y Asíncrona.
    - Gestión de Hilos: Grid, Bloques, Threads. Dimensiones: 1D, 2D y 3D.
    - Kernel, llamados Síncronos y Asíncronos.
    - Funciones.
    - Identificadores de Threads y Bloques.
    - Planificación de Threads.
    - Sincronización de Threads.
- Modelo de programación OpenCL.
    - Arquitecturas con soporte OpenCL.

  - Conceptos básicos de OpenCL Context, WorkQueue, WorkItems, Kernels.
  - Equivalencias OpenCL – CUDA.
- Diseño de programas en GPU.
- Estudio experimental de casos.
- Métricas de rendimiento: speedup y eficiencia.
- Métricas de consumo y eficiencia energética: Watt/flop.
- Análisis de perfomance. Aceleración en GPU con respecto a CPU.

Unidad 4: Modelo y jerarquía de Memoria de GPU
- Modelo de Memoria de GPU.
- Jerarquía de Memoria: Registros, Memoria Compartida, Memoria de constantes, Memoria de Texturas, Memoria Global. Memorias Cache: Constantes, Texturas, Nivel 1, Nivel 2.
- Patrones de Acceso a Memoria Global, relación entre segmentos y cantidad de transacciones.
- Patrones de Acceso a Memoria Compartida, bancos de memoria, conflicto de bancos, accesos sin conflictos.
- Concepto de Acceso Coalescente.
- El problema de la latencia.

Unidad 5: Optimizaciones
- Divergencia.
- Coalescencia y prefetching.
- Mezcla y granularidad de instrucciones.
- Asignación de recursos.

Unidad 6: Multi-GPUs y Arquitecturas Híbridas.
- Maquinas con más de una GPU.
- Arquitectura Híbrida Multicore-GPU: Integración de herramientas CUDA – OpenMP/Pthreads.
- Arquitectura Híbrida Cluster-GPU: Integración de herramientas CUDA - MPI.
- Arquitectura Híbrida Cluster-Multicore-GPU: Integracion de herramientas CUDA - OpenMP/Pthreads – MPI.
- Heterogeneidad – Balance de carga.
- Análisis de perfomance. Aceleración en GPU con respecto a Arquitecturas Multicores y Clusters.
- Casos de Estudio. Programación de aplicaciones.

METODOLOGÍA DE ENSEÑANZA Modalidad presencial

La asignatura se estructura con clases teórico-prácticas y prácticas experimentales.

- Las clases teórico-prácticas son dictadas por los Profesores de la asignatura y son obligatorias para la promoción.
- Las explicaciones de práctica son introductorias al trabajo en Laboratorio, para facilitar la utilización del equipamiento y software por los alumnos. Se desarrollan en las clases teórico-prácticas.
- El Taller propone el desarrollo de trabajos concretos con arquitecturas GPU y combinaciones de multicores y GPUs. Las actividades de Taller se hacen en máquina, en el contexto de las clases teórico-prácticas.
- Las consultas y correcciones son realizadas en forma presencial.
- En principio se utilizará la Sala de Cómputo de Postgrado (por la disponibilidad de placas GPU) y equipamiento especial del III-LIDI.

METODOLOGÍA DE ENSEÑANZA Modalidad semi presencial

Se hace notar que por la característica de las tareas experimentales, el alumno deberá tener acceso a algún modelo de arquitectura paralela y contar con alguna GPU para poder realizar los trabajos que se solicitan en el curso.

El alumno puede seguir los temas por el entorno WEB-UNLP y asistir a las consultas que se fijen para los alumnos presenciales.

EVALUACIÓN Modalidad presencial:

Para obtener la aprobación de cursada de la asignatura los alumnos deben aprobar todas las entregas de los diferentes trabajos experimentales, estas entregas pueden ser en grupos de 2 personas. Los trabajos no tienen reentregas. Además de las entregas los alumnos deben aprobar un examen parcial para el que se dispone de una fecha y dos recuperatorios.

Para la aprobación final de la asignatura se les propondrá un trabajo final experimental que deberán defender en un coloquio en una fecha de examen final.

EVALUACIÓN Modalidad semi presencial

Deben cumplir con los mismos requisitos que los alumnos en modalidad presencial.

BIBLIOGRAFÍA OBLIGATORIA

M. F. Piccoli, “Computación de Alto Desempeño utilizando GPU”. XV Escuela Internacional de Informática. Editorial Edulp, 2011.

Guil N. y Ujaldón M. “La GPU como arquitectura emergente para supercomputación”. In *XIX Jornadas de Paralelismo de Castellon*. 2008.

Kirk, D.,Hwu, W.. “Programming Massively Parallel Processors: A Hands-on Approach”. ISBN: 978-0-12-381472-2. Elsevier. 2010.

Luebke D. H.G. “How GPUs work”. *EEE Computer*, 40(2), 2007.

Sanders, J., Kandrot, E.. “Cuda by Example: An Introduction to General- Purpose Gpu Programming”. ISBN: 0131387685. Addison-Wesley Professional. 2010.

General-Purpose Computation on Graphics Processing Units. http://gpgpu.org.

Kerr A.and Diamos G. y Yalamanchili S. “Modeling GPU-CPU workloads and systems”. In *3rd Workshop on GP Computation on Graphics Processing Units*. ACM, 2010.

Grama A, Gupta A, Karypis G, Kumar V. “Introduction to parallel computing”. Second Edition. Pearson Addison Wesley, 2003.

Kindratenko, V.V el al "GPU clusters for high-performance computing," Cluster Computing and Workshops, 2009. CLUSTER '09. IEEE International Conference on , vol., no., pp.1,8, Aug. 31 2009-Sept. 4 2009

http://www.cs.caltech.edu/courses/cs101gpu/

Adrian Pousa, Victoria Sanz, Armando De Giusti “Performance Analysis of a Symmetric Cryptographic Algorithm on Multicore Architectures”. CACIC (XVII Congreso Argentino de Ciencias de la Computación). ISBN: 978-950-34-0756-1. Universidad de La Plata, La Plata, Argentina. 10 al 14 de Octubre de 2011. Publicado en el libro “Computer Science & Technology Series. XVII Argentine Congress of Computer Science Selected Papers” Capítulo “XI Distributed and Parallel Processing Workshop - Performance Analysis of a Symmetric Cryptographic Algorithm on Multicore Architectures” ISBN:978-950-34-0885-8.

Fernando Romero, Adrian Pousa, Victoria Sanz, Armando De Giusti “Consumo energético en arquitecturas multicore. Análisis sobre un algoritmo de criptografía simétrica”. CACIC (XVIII Congreso Argentino de Ciencias de la Computación). ISBN: 978-987-1648-34-4. Universidad Nacional del Sur, Bahia Blanca, Argentina. 8 al 12 de Octubre de 2012.

Adrian Pousa, Victoria Sanz, Armando De Giust “Performance Analysis of a Symmetric Cryptography Algorithm on GPU and GPU Cluster”. HPCLatam 2013 (VI Latin American Symposium on High Performance Computing). Páginas 113-121. Instituto de Ciencias Básicas, Universidad Nacional de Cuyo, Mendoza, Argentina. 22 al 26 de Julio de 2013.

Montes de Oca E., De Giusti L., De Giusti A., Naiouf M. “Comparación del uso de GPU y cluster de multicore en problemas con alta demanda computacional”. XII Workshop de Procesamiento Distribuido y Paralelo. CACIC2012. ISBN: 978987-1648-34-4. Pág. 267-275. Bahía Blanca, Buenos Aires, Argentina, Octubre 2012.

Montes de Oca E., Naiouf M., De Giusti L., Chichizola F., Giacomantone J., De Giusti A. “Una implementación paralela de las Transformadas DCT y DST en GPU. Análisis de performance”. XII Workshop de Procesamiento Distribuido y Paralelo. CACIC2012. ISBN: 978987-1648-34-4. Pág. 276-285. Bahía Blanca, Buenos Aires, Argentina, Octubre 2012.

BIBLIOGRAFÍA COMPLEMENTARIA

Joselli M., Zamith M., Clua E., Montenegro A., Conci A., Leal-Toledo R., Valente L., Feijo B., Dórnellas M., y Pozzer C. "Automatic dynamic task distribution between CPU and GPU for real-time systems". In *11th IEEE International Conference on Computational Science and Engineering*. 2008.

OpenGL Red Book – General resource for OpenGL/graphics programming

OpenGL Orange Book GPU/GLSL version of the Red Book

The CUDA Zone: http://www.nvidia.com/cuda Examples, documentation, drivers, etc.

NVIDIA. "Nvidia cuda compute unified device architecture, programming guide version 2.0". In *NVIDIA*. 2008a.

NVIDIA. "Nvidia geforce 8800 gpu architecture overview". In *NVIDIA*. 2006.

NVIDIA. Nvidia geforce gtx 200 gpu architectural overview. In *NVIDIA*. 2008b.

W. Hwu)Buck I. "Gpu computing with Nvidia Cuda". *ACM SIGGRAPH 2007 courses ACM*, 2007. New York, NY, USA.

Chen W. y Hang H. "H.264/avc motion estimation implementation on compute unified device architecture (cuda)". In IEEE, editor, *IEEE International Conference on Multimedia*. 2008.

Goyal N., Ormont J., Smith R., Sankaralingam K., y Estan C. "Signature matching in network processing using simd-gpu architectures". In *University of Wisconsin*. 2008.

Lieberman M., Sankaranarayanan J., y Samet H. "A fast similarity join algorithm using graphics processing units". In *ICDE 2008. IEEE 24th International Conference on Data Engineering 2008*. 2008.

Lloyd D., Boyd C., y Govindaraju N. "Fast computation of general fourier transforms on gpuS". In *IEEE International Conference on Multimedia and Expo*. 2008.

Luebke D. "Cuda: Scalable parallel programming for high-performance scientific computing". In *5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro, ISBI 2008*. 2008.

Nottingham A. y Irwin B. "Gpu packet classification using opencl: a consideration of viable classification methods". In *Research Conf. of the South African Inst. of Comp. Sc. and Inf. Technologists*. ACM, 2009.

Ryoo S., Rodrigues C., Baghsorkhi S., Stone S., Kirk D., y Hwu W. Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In *ACM*. ACM, 2008.

Mc. Cool M. "Programming models for scalable multicore programming". 2007. http://www.hpcwire.com/features/17902939.html

Rucci E., De Giusti A., Chichizola F., Naiouf M., De Giusti L. "DNA Sequence Alignment: hybrid parallel programming on multicore cluster". Proceedings of the International Conference on Computers, Digital Communications and Computing (ICDCCC '11), Vol. 1, Nikos Mastorakis, Valeri Mladenov, Badea Lepadatescu, Hamid Reza Karimi, Costas G. Helmis (Editors), WSEAS Press, September 15-17, 2011, Barcelona, ISBN: 978-1-61804-030-5, pp. 183-190.

Feng, W.C., "The importance of being low power in high-performance computing". Cyberinfrastructure Technology Watch Quarterly (CTWatch Quarterly). 2005.

Muresano Cáceres R. "Metodología para la aplicación eficiente de aplicaciones SPMD en clústers con procesadores multicore" Ph.D. Thesis, Universidad Autónoma de Barcelona, Barcelona, España, Julio 2011.

Sinha, R.; Prakash, A.; Patel, H.D., "Parallel simulation of mixed-abstraction SystemC models on GPUs and multicore CPUs," Design Automation Conference (ASP-DAC), 2012 17th Asia and South Pacific , vol., no., pp.455,460, Jan. 30 2012-Feb. 2 2012.

Lingyuan Wang, Miaoqing Huang, and Tarek El-Ghazawi. "Towards efficient GPU sharing on multicore processors". In Proceedings of the second international workshop on Performance modeling, benchmarking and simulation of high performance computing systems (PMBS '11). ACM, New York, NY, USA, 23-24.

Chao-Tung Yang, Chih-Lin Huang, Cheng-Fang Lin, "Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU Clusters", Computer Physics Communications 182 (2011) 266–269, Elsevier.

Alexandra Fedorova, Juan Carlos Saez, Daniel Shelepov and Manuel Prieto. Maximizing Power Efficiency with Asymmetric Multicore Systems. Communications of the ACM, Vol. 52 (12), pp 48-57. December 2009.

Nottingham A. y Irwin B. "Gpu packet classification using opencl: a consideration of viable classification methods". In Research Conf. of the South African Inst. of Comp. Sc. and Inf. Technologists. ACM,

CRONOGRAMA TENTATIVO DE CLASES Y EVALUACIONES
El cronograma detallado se pone en conocimiento de los alumnos al inicio del curso.

Clase	Contenidos/Actividades
1- Semana del 10/08	Unidad 1
2- Semana del 17/08	Unidad 2
3- Semana del 24/08	Unidad 3
4- Semana del 31/08	Actividades y consultas Prácticas.
5- Semana del 07/09	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
6- Semana del 14/09	Entrega y corrección de Trabajos Prácticos.
7- Semana del 21/09	Unidad 4
8- Semana del 28/09	Unidad 5
9- Semana del 05/10	Actividades y consultas Prácticas.
10- Semana del 12/10	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
11- Semana del 19/10	Entrega y corrección de Trabajos Prácticos.
12- Semana del 26/10	Unidad 6
13- Semana del 02/11	Actividades y consultas Prácticas.
14- Semana del 09/11	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
15- Semana del 16/11	Entrega y corrección de Trabajos Prácticos.
16- Semana del 23/11	Parcial 1ra Fecha
17- Semana del 30/11	Consultas Prácticas. 1er Recuperatorio del parcial
18- Semana del 07/12	Consultas Prácticas. 2do Recuperatorio del parcial
19- Semana del 14/12	Definición de trabajos finales

Contacto de la cátedra (mail, página, plataforma virtual de gestión de cursos):
Plataforma virtual: webunlp.unlp.edu.ar
Web: http://weblidi.info.unlp.edu.ar/catedras/tallerGPU/
Mail: apousa@lidi.info.unlp.edu.ar

Firmas del profesor responsable: