Taller de Programación sobre GPUs

Año 2017

Carrera/ Plan:
Licenciatura en Informática Plan 2015
Licenciatura en Sistemas Plan 2015
Licenciatura en Informática Plan 2003-07/Plan 2012
Licenciatura en Sistemas Plan 2003-07/Plan 2012

Año: 4
Régimen de Cursada: *Semestral*
Carácter: Optativa
Correlativas: Programación Concurrente
Profesor/es: Adrian Pousa
Hs. semanales : 9hs

FUNDAMENTACIÓN

Los procesadores gráficos (GPUs) han surgido como una alternativa dentro de los procesadores con múltiples núcleos, por sus características de rendimiento 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 rendimiento con arquitecturas 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 rendimiento con arquitecturas 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 (de acuerdo al Plan de Estudios)

- *GPU: Introducción a GPGPU*
- *Arquitecturas GPU - Modelo GPU-CPU.*
- *Modelo de Programación GPU - Resolución de aplicaciones - Métricas.*

- *Modelo y jerarquía de Memoria de GPU.*
- *Optimizaciones sobre GPU.*
- *MultiGPUs. Integración con otras arquitecturas. Arquitecturas Híbridas.*

PROGRAMA ANALÍTICO

Unidad 1: *GPU: Introducción a HPC y GPGPU*

Introducción al cómputo de altas prestaciones (HPC). Concepto de sistema paralelo. Hardware y Software paralelo. Clasificación de Hardware paralelo: Taxonomía de Flynn y Clasificación según el modelo de memoria. Jerarquías de memoria. Software paralelo: paralelismo funcional o de control y paralelismo de datos, características de las aplicaciones (CPU, Memoria, Entrada/Salida, Fases), herramientas de desarrollo. Introducción a las arquitecturas GPU y su uso en HPC. Concepto 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. Arquitectura Pezy-SC. 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. Diseño de programas en GPU. Estudio experimental de casos. Métrica de rendimiento: speedup. Análisis de rendimiento. 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.*

Máquinas multi-GPUs. Arquitectura Híbridas: Modelo Multicore-GPU, Modelo Cluster-GPU y cluster de GPUs, Modelo Multicore-Cluster-GPUs. Heterogeneidad. Distribución de carga.

BIBLIOGRAFÏA

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.

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,

METODOLOGÍA DE ENSEÑANZA

Modalidad presencial

La asignatura sigue una modalidad taller en la cual se alternan clases teóricas con prácticas experimentales.

Durante el desarrollo de la asignatura no se tomará asistencia. La existencia de un control de asistencia es a fines estadísticos.

Las clases teóricas introducen los conceptos teóricos de la asignatura aplicables en las prácticas experimentales.

Las prácticas experimentales se desarrollan la Sala de Cómputo de Postgrado (por la disponibilidad de arquitecturas paralelas tipo GPU y la configuración de estas en modo cluster) y equipamiento especial del III-LIDI. Estas clases consisten de trabajos que deben desarrollar los alumnos en las arquitecturas disponibles.

Las consultas y correcciones son realizadas en forma presencial por las dificultades que representan al realizarlas en otros medios como pueden ser vía WEB.

Modalidad semi-presencial

Dada la no obligatoriedad de las clases, los alumnos en modalidad semi-presencial pueden seguir los temas por el entorno WEB-UNLP y asistir a las consultas que se fijen para los alumnos presenciales.

Se hace notar que por la característica de las tareas experimentales, el alumno deberá tener acceso al menos a un modelo de arquitectura paralela que incluya GPUs para poder realizar los trabajos que se solicitan en el curso. Es recomendable que asistan a las clases prácticas experimentales para el desarrollo de los conocimientos que se adquieren en dichas clases.

Redictado

Dada las características de la materia y la experiencia de años anteriores la cátedra no ve necesidad alguna de realizar un redictado.

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. Cada trabajo es acompañado por un coloquio y puede tener solo una re-entrega.

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 del final se computa el promedio de la nota de los trabajos y sus respectivos coloquios con la nota del parcial.

La nota del final será válida por un semestre posterior a la finalización de la materia pasado ese período el alumno deberá revalidar la nota del final mediante un coloquio o examen escrito.

Modalidad semi-presencial y Evaluación por promoción

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

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1	**Semana del 14/08**	Unidad 1
2	**Semana del 21/08**	Unidad 2
3	**Semana del 28/08**	Unidad 3
4	**Semana del 04/09**	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
5	**Semana del 11/09**	Entrega y corrección de Trabajos Prácticos.
6	**Semana del 18/09**	Unidad 4
7	**Semana del 25/09**	Unidad 5
8	**Semana del 02/10**	Actividades y consultas Prácticas. Re- Entrega y corrección.
9	**Semana del 16/10**	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
10	**Semana del 23/10**	Entrega y corrección de Trabajos Prácticos.
11	**Semana del 30/10**	Unidad 6
12	**Semana del 06/11**	Actividades y consultas Prácticas. Re- Entrega y corrección.
13	**Semana del 13/11**	Actividades y consultas Prácticas. Presentación de Trabajo práctico a entregar.
14	**Semana del 20/11**	Entrega y corrección de Trabajos Prácticos.
15	**Semana del 27/11**	Parcial 1ra Fecha – Correción y muestra
16	**Semana del 04/12**	Consultas Prácticas. 1er Recuperatorio del parcial – Correción y muestra
17	**Semana del 11/12**	Consultas Prácticas. 2do Recuperatorio del parcial – Correción y muestra
18	**Semana del 05/12**	Re- Entrega y corrección de Trabajos Prácticos.

Evaluaciones previstas	Fecha
Parcial 1ra Fecha	**Semana del 27/11**
1er Recuperatorio del parcial	**Semana del 04/12**
2do Recuperatorio del parcial	**Semana del 11/12**

Contacto de la cátedra (mail, sitio WEB, 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

Firma del/los profesor/es