Especialización en Cómputo de Altas Prestaciones – Modalidad a distancia

Taller de Programación sobre Arquitecturas Paralelas Avanzadas

Año 2021

Duración: 70 hs. Totales.

Cantidad de horas presenciales/VC: 20 hs.

Cantidad de horas de actividades en línea y de trabajo final: 50 hs.

OBJETIVOS GENERALES:

A la aparición de arquitecturas many-core (como las GPU o los procesadores MIC), se ha sumado el uso de FPGAs debido a su potencia de cómputo y rendimiento energético. Su combinación en sistemas HPC da lugar a plataformas híbridas con diferentes características Lógicamente, esto trae aparejado una revisión de los conceptos del diseño de algoritmos paralelos (incluyendo los lenguajes mismos de programación y el software de base), así como la evaluación de las soluciones que éstos implementan. También resulta necesario investigar las estrategias de distribución de datos y de procesos a fin de optimizar la performance. Además, el estudio del consumo y la eficiencia energética de los nuevos sistemas paralelos se vuelve tan importante como el de métricas clásicas (speedup, eficiencia, escalabilidad) debido a los costos económicos y los problemas operativos asociados.
En este Taller se analizan los problemas de procesamiento paralelo desde el punto de vista del sotware, teniendo especialmente en cuenta las nuevas arquitecturas sobre las que se implementa esta software.
Además se tienen en cuenta las posibilidades de empleo de placas de bajo costo conformando una arquitectura utilizable en HPC.
Por último se tratan especialmente los temas de resiliencia (dada la incidencia de las fallas al crecer la complejidad de las arquitecturas) y de consumo energético (dada la importancia actual de esta métrica de perfomance).

COMPETENCIAS A DESARROLLAR EN RELACION CON EL OBJETIVO DE LA CARRERA

C.2- Conocer los fundamentos para el desarrollo de Sistemas Paralelos (incluyendo la relación entre hardware y software).

C.3- Tener capacidad de análisis, diseño, implementación y optimización de algoritmos distribuidos y paralelos, aplicables a problemas numéricos y no numéricos en diferentes áreas

del conocimiento, incluyendo su análisis de rendimiento y eficiencia.

C.4- Conocer y analizar arquitecturas dedicadas para procesamiento paralelo. Tener capacidad de configurar arquitecturas y desarrollar programación en la nube (Cloud Computing).

C.7- Estar capacitado para el desarrollo de aplicaciones paralelas sobre diferentes arquitecturas.

CONTENIDOS MINIMOS:

- Programación paralela sobre arquitecturas GPU y Cluster de GPUs.
- Programación paralela sobre arquitecturas MIC (Many Integrated Core Architectures).
- Programación paralela sobre arquitecturas FPGA (Field Programmable Gate Arrays).
- Programación paralela sobre arquitecturas TPU (Tensor Processor Unit).
- Programación paralela sobre arquitecturas que integran multicores y alguna de las arquitecturas expuestas.
- Métricas de perfomance en cada caso, incluyendo rendimiento y consumo energético.
- Resiliencia y diferentes tipos de fallas en las diferentes arquitecturas. Impacto sobre la programación de aplicaciones.
- E/S paralela y su influencia en el rendimiento de aplicaciones paralelas.
- Casos de estudio experimental con cada tipo de arquitectura.

PROGRAMA

Programación paralela sobre arquitecturas GPU y Cluster de GPUs.

- Características de las GPUs y mecanismos de programación paralela sobre ellas.
- Lenguajes / entornos orientados a GPUs. OpenMP y CUDA.
- Máquinas multicore que incluyen una o más GPUs y programación sobre ellas.
- Programación sobre Cluster de máquinas multicore cada una con una o más placas de GPU, lo que permite combinar OpenMP/MPI/CUDA o Pthread/MPI/CUDA

Programación paralela sobre arquitecturas MIC (Many Integrated Core Architectures).

- Características de los MIC y diferencias con la programación sobre GPUs.
- Placas disponibles tipo Xeon PHI y KNL y lenguajes / entornos orientados a su programación eficiente.
- Análisis de algoritmos que emplean MIC y comparación con soluciones sobre multicores / GPUs.

Programación paralela sobre arquitecturas FPGA (Field Programmable Gate Arrays).

- Características de los FPGA y sus áreas tradicionales de aplicación.
- Modelos y herramientas de programación clásica sobre FPGA.
- Evolución de la programación sobre FPGA a partir de arquitecturas híbridas tales como las de INTEL (ALTERA) o de IBM (Xilinx). Herramientas nuevas para la programación de FPGA.
- Implementación y análisis de algoritmos paralelos sobre FPGA.
- Métricas de rendimiento/consumo comparativas con otras soluciones ya vistas.

Programación paralela sobre arquitecturas TPU (Tensor Processor Unit).

- Características de las TPU y su posible aplicación como aceleradores en programación paralela.
- Herramientas de programación sobre TPUs.
- Análisis del framework Tensor Flow y su empleo en programación sobre TPUs.
- Implementación y análisis de algoritmos paralelos sobre TPUs.
- Tasas de aceleración y eficiencia energética provistas por esta nueva arquitectura, en comparación con otras ya estudiadas.

Métricas de perfomance. Eficiencia energética.

- Importancia de la eficiencia energética en arquitecturas con gran número de procesadores.
- Análisis de las metodologías y herramientas para medir y optimizar el consumo energético.
- Técnicas de reducción del consumo energético considerando la aplicación y el soporte de la arquitectura sobre la que se ejecuta el algoritmo paralelo.
- Evaluación de eficiencia energética en diferentes algoritmos y plataformas paralelas.
- Empleo de los contadores de hardware para estimar consumo y ajustar dinámicamente la ejecución de algoritmos paralelos.

Temas actuales a considerar en la Programación de Arquitecturas Paralelas avanzadas.

- Resiliencia y tolerancia a fallas en cada caso.
- Análisis de la E/S paralela en diferentes modelos de arquitectura y su impacto en la perfomance de los algoritmos paralelos.
- Posibilidad de empleo de placas de bajo costo (Tipo Raspberry Pi / Odroid) en aplicaciones paralelas. Análisis de rendimiento / costo / consumo.

ACTIVIDADES EXPERIMENTALES y DE INVESTIGACION

Tareas en Laboratorio (presencial o remoto)

Tal como se explica en el item relacionado con la metodología, ésta se basa en clases sincrónicas (presenciales o remotas) combinadas con actividades demostrativas en el laboratorio para aplicar los conceptos teóricos y que así el alumno adquiera las competencias y habilidades sobre cada uno de los temas que forman parte del contenido de la asignatura.

Además el alumno debe analizar un proyecto/desarrollo relacionado con los temas dictados en la teoría, cuya implementación concreta se realiza sobre máquinas / placas específicas y/o una infraestructura virtualizada que los alumnos pueden acceder en forma remota (en el Laboratorio dedicado a Paralelismo en el Postgrado).

Investigación/ Estudios adicionales:

Los alumnos analizarán papers relacionados con los problemas de paralelización de algoritmos y el proceso de análisis / diseño e implementación de los mismos, en particular sobre arquitecturas avanzadas explicadas en clase.

Se les propondrán temas de I+D orientados al estudio comparativo de métricas de rendimiento en algoritmos sobre los diferentes tipos de arquitecturas analizadas en el curso, de modo de potenciar el conocimiento trasmitido en la teoría.

METODOLOGIA Y MODALIDAD DE EVALUACION

La metodología se basa en clases sincrónicas a través del sistema de videoconferencias adoptado por el Postgrado de Informática combinadas con sesiones en el laboratorio remoto para aplicar los conceptos teóricos y que así el alumno adquiera las competencias y habilidades sobre cada uno de los temas que forman parte del contenido de la asignatura.

Se requiere un 80% de asistencia a los encuentros sincrónicos, incluyendo el encuentro inicial de presentación de la materia, y el encuentro final de integración, ambos de asistencia obligatoria.

El trabajo se complementa con un proyecto experimental que debe desarrollar el alumno para cumplimentar las horas asignadas con soporte tutorizado por el profesor (*on-line*) y seguimiento a través del Entorno Virtual IDEAS contemplado en el SIED de la Facultad de Informática de la UNLP.

El despliegue práctico se realizará sobre una infraestructura virtualizada accesible al alumno en la que se puede ejecutar aplicaciones y medir su rendimiento con las técnicas explicadas en el curso.
La evaluación se realizará mediante un examen escrito al final de las sesiones sincrónicas para evaluar el grado de conocimientos del alumno (20%), el proyecto/desarrollo experimental que deberá entregar el alumno al final de las horas programadas (70%) y la participación y aportaciones de calidad/excelencia a las soluciones propuestas (10%).

RECURSOS Y MATERIALES DE ESTUDIO

Como materiales de estudio, se dispone de:

- Presentaciones multimedia desarrolladas ad-hoc para introducir cada uno de los diferentes ejes temáticos.
- Píldoras formativas con la explicación de algunos temas
- Ejemplos donde se aplican los conceptos teóricos
- Ejercicios prácticos que son desarrollados en clase
- Material de lectura para estudiar y profundizar conceptos abordados en las clases
- Enlaces a artículos de actualidad de repositorios reconocidos en el área
- Acceso a equipamiento remoto situado en la Facultad de Informática de la UNLP y también en la nube (Cloud) de acuerdo a la disponibilidad del Postgrado de la Facultad de Informática para sus cursos.

- Software específico para determinadas actividades de laboratorio que se detallan en este programa.

ACTIVIDADES EXPERIMENTALES Y APROPIACIÓN DE SABERES

Los trabajos experimentales pueden desarrollarse en cada clase o continuarse en más de una clase. Parten de una especificación/consigna del docente (explicada en la clase) y un trabajo individual o en grupos que interactúan en el que los alumnos resuelven un problema experimental concreto relacionado con la temática.

Los trabajos podrán ser individuales o grupales. Para esto último se configura el entorno virtual para que los alumnos del mismo grupo se encuentren en un espacio virtual diferente del resto. Durante el desarrollo del trabajo, el docente estará conectado respondiendo dudas y consultas.

Estos trabajos pretenden desarrollar y/o fortalecer las aptitudes de opinión crítica en los temas

relativos del curso. Los alumnos deberán sintetizar su comprensión de los temas, al realizar correctamente la tarea experimental propuesta.

También se pretende desarrollar la capacidad de poder comunicar y transmitir los resultados, en presentaciones pautadas a lo largo del curso.

En general, finalizada una actividad, hay una sesión de discusión conjunta donde los participantes comunicarán sus opiniones e intercambiarán los distintos puntos de vista.

BIBLIOGRAFÍA

Introduction to Parallel Computing.
Grama, Gupta, Karypis,Kumar. Addison Wesley 2003

Foundations of Multithreaded, Parallel and Distributed Programming
Andrews. Addison Wesley 2000.

Parallel Programming
Wilkinson, Allen. Prentice Hall 2005.

Sourcebook of Parallel Computing
Dongarra, Foster, Fox, Gropp, Kennedy, Torczon,White. Morgan Kauffman 2003.

MPI: The complete Reference
Snir, Otto, Huss-Lederman , Walker , Dongarra , Cambridge, MA: MIT Press, 1996.

Cloud Computing: A Hands-On Approach
Arshdeep Bahga, Vijay Madisetti. CreateSpace Independent Publishing Platform, 2013. ISBN-13: 978-1494435141

Programming Massively Parallel Processors: A Hands-on Approach (Applications of GPU Computing Series)
David B. Kirk, Wen-mei W. Hwu. Morgan Kaufmann, 2010. ISBN-13: 978-0123814722

CUDA Programming: A Developer's Guide to Parallel Computing with GPUs (Applications of Gpu Computing)
Shane Cook. Morgan Kaufmann, 2012. ISBN-13: 978-0124159334

“Cloud Tensor Processing Unit (TPU)”
Google Inc. Disponible en https://cloud.google.com/tpu/docs/tpus?hl=es-419

“Towards a Malleable Tensorflow Implementation”
L. A. Libutti, F. D. Igual, L. Piñuel, L. D. Giusti, and M. Naiouf,, Cloud Computing, Big Data & Emerging Topics. 8th Conference, JCC-BD&ET 2020, La Plata, Argentina, September 8-10, 2020, Proceedings, págs. 30-40, doi. 10.1007/978-3-030- 61218-4_3, 2020. [21] Costanzo, M., R

“Intel Xeon Phi Processor High Performance Programming Knights Landing Edition”.
Reinders, J., Jeffers, J., Sodani, A.. Morgan Kaufmann Publishers Inc., Boston, MA, USA, 2016

“SDAccel Development Environment”.
Xilinx Inc. [Online]. Disponible en http://www.xilinx.com/products/design-tools/softwarezone/sdaccel.html

“PMCTrack: Delivering performance monitoring counter support to the OS scheduler”.
Saez, J.C., Pousa, A., Rodríguez-Rodríguez, R., Castro, F., Prieto-Matias, M. The computer journal Volume 60, Issue 1 January 2017.

Raspberry PI. https://www.raspberrypi.org/

“Accelerating Pattern Matching on Intel Xeon Phi Processors”
V. Sanz, A. Pousa, M. Naiouf, and A. De Giusti, , Algorithms and Architectures for Parallel Processing. ICA3PP 2020., ISBN: 978-3-030-60245-1, págs. 262-274, doi. 10.1007/978-3-030-60245-1_18, 2020.

Odroid http://www.hardkernel.com

“ACFS: A Completely Fair Scheduler for Asymmetric Single-ISA Multicore Systems”.
Juan Carlos Saez, Adrian Pousa, Daniel Chaver, Fernando Castro, Manuel Prieto Matias: In: ACM SAC 2015 (The 30TH ACM/SIGAPP Symposium on applied computing).

“Soft errors detection and automatic recovery based on replication combined with different levels of checkpointing”.
D. Montezanti, E. Rucci, A. D. De Giusti, M. Naiouf, D. Rexachs, and E. Luque, Future generation computer systems (ISSN 0167- 739X), vol. 113, págs. 240-254, doi. https://doi.org/10.1016/j.future.2020.07.003, 2020.

“Unified Power Modeling Design for Various Raspberry Pi Generations Analyzing Different Statistical Methods”.
J. M. Paniego, L. Libutti, M. P. Puig, F. Chichizola, L. De Giusti, M. Naiouf, and A. De Giusti, En: Computer Science – CACIC 2019. communications in Computer and Information Science., ISBN: 978-3-030-48325-8, Springer International Publishing, págs. 53-65, 2020.

Intel. “Intel Acquisition of Altera”.
Disponible en http://intelacquiresaltera.transactionannouncement.com [6] Sean Settle: “High-performance Dynamic Programming on FPGAs with OpenCL”. In: IEEE High Performance Extreme Computing Conference. 2013.

“Collaborative, distributed and scalable platform based on mobile, cloud, micro services and containers for intensive computing tasks”.
D. Petrocelli, A. E. De Giusti, and M. Naiouf, . Short papers of the 8th Conference on Cloud Computing, Big Data & Emerging Topics (JCC-BD&ET 2020), ISBN: 978-950- 34-1927-4, págs. 10-13, 2020.

“Towards Management of Energy Consumption in HPC Systems with Fault Tolerance”
M. Morán, J. Balladini, D. Rexachs, E. Rucci. Proceedings of the IV IEEE ARGENCON 2020 CONGRESS, En prensa, 2020.

“Power characterisation of shared-memory HPC systems”
Balladini, J., Rucci, E., De Giusti, A., Naiouf, M., Suppi, R., Rexachs, D., Luque, E. Computer Science & Technology Series – XVIII Argentine Congress of Computer Science Selected Papers. ISBN 978-987-1985-20-3. Págs. 53-65. 2013.

“On the Calibration, Verification and Validation of an Agent-Based Model of the HPC Input/Output System”.
D. Encinas, M. Naiouf, A. De Giusti, S. Mendez, D. Rexachs, and E. Luque. Proceedings from The Eleventh International Conference on Advances in System Simulation (SIMUL 2019), November 24 - 28, 2019.