



UNIVERSIDAD TÉCNICA DEL NORTE
FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS
CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

TRABAJO DE GRADO PREVIO APROBACIÓN PARA LA OBTENCIÓN DEL TÍTULO
DE INGENIERA EN SISTEMAS COMPUTACIONALES

TEMA:

COMPARATIVA DE LOS PARADIGMAS DE PROGRAMACIÓN PARALELA: POR
MANEJO DE THREADS, POR PASO DE MENSAJES E HÍBRIDA.

ARTÍCULO CIENTÍFICO

AUTORA: GONZAGA NOGUERA ROSA MERCEDES

DIRECTOR: ING. PABLO LANDETA

IBARRA-ECUADOR

2015

Comparativa de los paradigmas de programación paralela: por manejo de threads, por paso de mensajes e híbrida

Srta. Rosa GONZAGA¹, Ing. Pablo LANDETA²

¹ Carrera de Ingeniería en Sistemas Computacionales, Tumbabiro, Calle Sucre y Marieta Veintimilla esq. San Miguel de Urcuquí, Imbabura

² Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ingeniería en Ciencias Aplicadas, Universidad Técnica del Norte, Av. 17 de Julio, 5-21 Ibarra, Imbabura

rositamg1993@gmail.com, palandeta@utn.edu.ec

Resumen. *El presente trabajo tiene como fin realizar una comparación entre los diferentes paradigmas de programación paralela, en los cuales se encuentran: por manejo de threads, por paso de mensajes, e híbrida; esto se lleva a cabo a través de una investigación explorativa y luego descriptiva. La programación paralela ha ido tomado fuerza con la aparición de las nuevas arquitecturas de multiprocesadores, es por eso que varias empresas han desarrollado lenguajes que ayuden al aprovechamiento de estos nuevos recursos entre los que se encuentran también las GPUs o Tarjetas Gráficas.*

Cada lenguaje de programación paralela presenta diferentes características que indican la compatibilidad que tienen con las arquitecturas existentes, en este documento se detalla la descripción de cada paradigma y un lenguaje referencial que se adapta a cada uno de ellos y las ventajas que se tiene al trabajar con las diferentes librerías.

Al final encontramos un análisis construido a partir de la investigación realizada, detallando cuál de los paradigmas y el lenguaje es más conveniente a utilizar, tomando en cuenta los tiempos de ejecución y la interacción con el usuario, basado en la arquitectura utilizada. Por favor, tenga en cuenta que existen diferentes encabezados de las páginas pares e impares. El encabezado de las páginas pares se abrevia si es necesario para que no exceda de una línea. El tipo de letra del título del artículo es igual a este modelo, así como el nombre, Universidad, dirección y correos electrónicos del autor y coautor.

Palabras Claves

Paralelismo, GPUs, comparativa.

Abstract. *The present work has as objective realize a comparison between different parallel paradigms of the programming, in which include: by handling of threads, message passing, and hybrid; this is accomplished through an exploratory research and then descriptive. Parallel programming has gone taking strength with the appearance, of new multiprocessor architectures, so several companies*

have developed languages to assist the use of these new resources among which are also the GPU or Graphics Cards.

Each parallel programming language has different characteristics that indicate the compatibility with the existing architectures, in this document describe the description of each paradigm and referential language that adapts to each of them and the advantages you have when working with detailed different libraries.

To the end we found an analysis constructed from the investigation realized, detailing which of the paradigms and language is more convenient to use, taking into account the execution times and interaction with the user, based on the architecture used.

Keywords

parallelism, GPUs, comparative.

1. Introducción

Las computadoras en su mayoría contienen procesadores que son mono-núcleo, es decir tienen un solo cerebro para ejecutar todos los procesos, para ello nacen los procesadores multicore, estos pueden repartir los procesos a través de varios cerebros para una mejor ejecución, mejorando así la capacidad de procesamiento.

Con el avance de la tecnología en el área de computación, hoy en día se tiene una amplia gama de lenguajes de programación que gozan de una alta popularidad, con ellos se puede desarrollar desde pequeñas aplicaciones hasta amplios sistemas que son necesarios para la sociedad actual.

La Programación Paralela se ha tomado un amplio campo en el desarrollo de aplicaciones, está tiene como principio la división de un problema de gran tamaño en problemas más pequeños, así de esta forma se puedan

ejecutar varias instrucciones de una manera simultánea, ampliando el uso de los cerebros de un procesador.

La presente investigación pretende establecer las características de los paradigmas de Programación Paralela que existen para un mejor aprovechamiento de los recursos del computador, especialmente los núcleos del procesador.

2. Materiales y Métodos

La presente investigación se realizó inicialmente a través de la investigación exploratoria, en donde se fue adquiriendo los conocimientos básicos sobre la programación paralela, y los respectivos paradigmas en los que tenemos: manejo de threads, paso de mensajes e híbrida. Se analizó conceptos básicos con respecto a cada paradigma y su uso. Después de tener una información determinada de cada paradigma se realizó una investigación descriptiva en donde se analiza a fondo cada uno de los paradigmas con un respectivo lenguaje de programación para cada uno, y de esta forma sacar las ventajas y desventajas que estos presenten.

2.1 Investigación Explorativa

“Los estudios exploratorios nos permiten aproximarnos a fenómenos desconocidos, con el fin de aumentar el grado de familiaridad y contribuyen con ideas respecto a la forma correcta de abordar una investigación en particular. Con el propósito de que estos estudios no se constituyan en pérdida de tiempo y recursos, es indispensable aproximarnos a ellos, con una adecuada revisión de la literatura.” (Grajales, 2000)

2.2 Investigación Descriptiva

“Los estudios descriptivos buscan desarrollar una imagen o fiel representación (descripción) del fenómeno estudiado a partir de sus características. Describir en este caso es sinónimo de medir. Miden variables o conceptos con el fin de especificar las propiedades importantes de comunidades, personas, grupos o fenómeno bajo análisis.” (Grajales, 2000)

3. Resultados

De acuerdo a la investigación realizada se ha encontrado que el paralelismo reduce los costos en las instrucciones que normalmente se realizan de una manera secuencial. El aprovechamiento de las nuevas arquitecturas lleva a un futuro donde el tiempo se vuelve valioso al momento de realizar grandes cálculos, y de acuerdo a las aplicaciones realizadas se encontró los siguientes resultados:

Tablas de tiempos de ejecución

Código Secuencial	MPI	OpenCL	CUDA
0.016s	0.097s	1.072s	0.063s

Tabla 1 Tiempos de ejecución de Código de Hello World

Código Secuencial	Código MPI	Código OpenCL	Código CUDA
0.016s	0.016s	0.052s	0.027s

Tabla 2 Tiempos de ejecución de Cálculo de PI

4. Conclusiones

A partir de los resultados obtenidos se concluye que el paradigma que mejor se adapta a las arquitecturas es la programación paralela híbrida, pues esta aprovecha de una mejor manera los recursos físicos de la arquitectura, en la investigación realizada para el desarrollo de este paradigma se utilizó el lenguaje de programación CUDA, con la desventaja de que fue desarrollado solo para arquitecturas Nvidia, por lo cual para un mejor aprovechamiento del paradigma el lenguaje mejor adaptado es OpenCL, ya que brinda una mayor compatibilidad con las nuevas tecnologías.

Agradecimientos

A la Universidad Técnica del Norte por su apoyo así como a la Facultad de Ingeniería en Ciencias Aplicadas, a sus docentes que a través de la carrera establecieron los conocimientos necesarios para la elaboración del presente trabajo.

Referencias Bibliográficas

- [1] Barker, B. (2015). Message passing interface (mpi). Paper presented at the Workshop: High Performance Computing on Stampede.
- [2] Blair-Chappell, S., & Stokes, A. (2012). Parallel Programming with Intel Parallel Studio XE Retrieved from

- <http://utn.ebib.com/patron/FullRecord.aspx?p=817946>
- [3] Flynn, M. J., & Rudd, K. W. (1996). Parallel architectures. *ACM Computing Surveys (CSUR)*, 28(1), 67-70.
- [4] Grajales, T. (2000). Tipos de investigación. On line)(27/03/2.000). Revisado el.
- [5] Jorba i Esteve, J., Margalef, T., & Morajko, A. (2006). Análisis automático de prestaciones de aplicaciones paralelas basadas en paso de mensajes: Universitat Autònoma de Barcelona.
- [6] Kirk, D. B., & Wen-me, W. H. (2012). Programming massively parallel processors: a hands-on approach: Newnes.
- [7] Leibovich, F., Gallo, S., De Giusti, A. E., De Giusti, L. C., Chichizola, F., & Naiouf, M. (2011). Comparación de paradigmas de programación paralela en cluster de multicores: pasaje de mensajes e híbrido. Paper presented at the XVII Congreso Argentino de Ciencias de la Computación.
- [8] Munshi, A., Gaster, B., Mattson, T. G., & Ginsburg, D. (2011). *OpenCL programming guide*: Pearson Education.
- [9] Nasmachnow, S. (2004). Algoritmos genéticos paralelos y su aplicación al diseño de redes de comunicaciones confiables.
- [10] OpenCL, K. The open standard for parallel programming of heterogeneous systems.
- [11] Piccoli, M. F. (2011). Computación de alto desempeño en GPU.
- [12] Tello, A. A., & Huesca, J. d. J. T. (2012). *OpenCL, Programación concurrente y paralela*. Editor: Prof. Ariel Ortiz Ramírez, 75.
- [13] Yang, C.-T., Huang, C.-L., & Lin, C.-F. (2011). Hybrid CUDA, OpenMP, and MPI parallel programming on multicore GPU clusters. *Computer Physics Communications*, 182(1), 266-269.
- [14] Guim, F., & Rodero, I. Arquitecturas basadas en computación gráfica (GPU).

Sobre los Autores...



Rosa GONZAGA Egresada de la Carrera de Ingeniería en Sistemas Computacionales, sus estudios secundarios fueron en el colegio técnico "Mariano Suárez Veintimilla", obteniendo el título de bachiller en comercio y administración, especialidad en aplicaciones informáticas, inicia sus estudios universitarios en el año 2010 en la Universidad Técnica del Norte.



Ing. Pablo LANDETA Actualmente trabaja como docente de la carrera de Ingeniería en Sistemas Computacionales de la Universidad Técnica del Norte (UTN) en Ibarra - Ecuador. Obtuvo su título de Ingeniero en Sistemas Computacionales en la misma universidad en el año 2004. Actualmente se encuentra culminando sus estudios de postgrado en el programa Maestría en Gerencia de Sistemas de la Universidad de las Fuerzas Armadas (ESPE) en Sangolquí - Ecuador.