UNIVERSIDAD TÉCNICA DEL NORTE

FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES

GRADE WORK PRIOR TO OBTAINING THE TITLE OF ENGINEERING IN COMPUTER SYSTEMS

THEME:

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

SCIENTIFIC ARTICLE

AUTHOR: GONZAGA NOGUERA ROSA MERCEDES

DIRECTOR: ENG. PABLO LANDETA

IBARRA-ECUADOR

2015

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

Srta. Rosa GONZAGA[1], Ing. Pablo LANDETA [2]

[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 objetive 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. Introduction

The computers mostly contain processors that are mono-core, so have a single brain to execute all processes, for it born multicore processors, they can distribute processes across multiple brains for better performance, thus improving the processing capacity.

With the advancement of technology in the area of computing, nowadays you have a wide range of programming languages that enjoy a high popularity, with them you can develop from small applications to large systems that are required for actual society.

The Parallel Programming has taken a wide range in application development, It has the first division problem large into smaller problems, so it’s can execute multiple instructions simultaneously , expanding the use of Brains processor.

The present research want to establish the characteristics of Parallel paradigms Programming that exist for a better use of computer resources, especially of the processor cores.

2. Materials and Methods

The present research was realized initially through exploratory research, where it was acquiring the basic knowledge about parallel programming, and the respective paradigms in which we have: threads management, message passing and hybrid. Basics for each paradigm were analyzed and their use. After having a certain information in a descriptive research paradigm which analyzes in depth each of the paradigms with a respective programming language for each was made, so take the advantages and disadvantages that these present.

2.1 Explorative Research

"Exploratory studies allow us to approach unknown phenomena, in order to increase the degree of familiarity and contribute ideas about the proper way to address a particular research. For the purpose of these studies do not constitute waste of time and resources, it is essential to approach them with an adequate review of the literature. "(Grajales, 2000)

2.2 Descriptive Research

"Descriptive studies seek to develop an image or faithful representation (description) of the studied phenomenon from its features. Describe in this case is synonymous with measuring. Measured variables or concepts to specify the important properties of communities, people, groups or phenomenon under analysis. "(Grajales, 2000)

3. Results

According to the research realized it has been found that reduces costs parallelism in the instructions that are typically performed in a sequential manner. The exploitation of new architectures leads to a future where time becomes valuable when performing large calculations, and according to the applications made it's found the following results:

Runtimes tables

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

Tabla 1 HelloWorld Runtimes

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

Tabla 2 PI Calculation Runtimes

4. Conclusiones

From the results it is concluded that the paradigm that best suits architectures is the hybrid parallel programming, because the advantage of a better way physical resources of architecture, in the research for the development of this paradigm was used CUDA programming language, with the disadvantage that was developed only for Nvidia architectures, making for a better use of the paradigm the better language to use is OpenCL, as it provides greater compatibility with new technologies.

Thanks

To the Técnica del Norte University for their support and to the Faculty of Engineering of Applied Science, to the teachers who through the race they established the necessary knowledge for the preparation of this work.

References

[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.eblib.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-mei, 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] Nesmachnow, 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).

About the authors...

Rosa GONZAGA Engineering in Computer Systems, hers high school studies were in the technical school "Mariano Suárez Veintimilla", obtaining a bachelor's degree in commerce and administration, specializing in informatics applications, he began his university studies in 2010 at the Technical University North .

Eng. Pablo LANDETA Currently he works as a teacher's career in Computer Systems Engineering from the Technical University of the North (UTN) in Ibarra - Ecuador. He earned his engineering degree in Computer Systems at the same university in 2004. He is currently culminating his graduate studies in the Master's program in Systems Management from the University of the Armed Forces (ESPE) in Sangolquí - Ecuador.