

WICC 2014 XVI Workshop de Investigadores en Ciencias de la Computación

“Análisis de Eficiencia en Sistemas Paralelos”

Jorge Osio¹, Diego Montezanti^{1,2}, Martin Daniel Morales¹

¹Instituto de Ingeniería y Agronomía - Universidad Nacional Arturo Jauretche

²Instituto de Investigación en Informática LIDI (III-LIDI) - Facultad de Informática
- UNLP

josio@unaj.edu.ar, dmontezanti@lidi.unlp.edu.ar, martinmorales@unaj.edu.ar

Resumen

El objetivo de esta línea de investigación es estudiar problemas de consumo y eficiencia energética en arquitecturas de procesamiento paralelo, analizando la relación que guarda con los paradigmas de programación de algoritmos paralelos. Por otro lado se pretende investigar las consecuencias de las fallas que puedan presentarse en arquitecturas paralelas y distribuidas y proponer estrategias de detección y recuperación.

Finalmente, resulta necesario estudiar casos concretos sobre arquitecturas con procesadores de múltiples núcleos. Específicamente se pretende analizar la factibilidad del procesamiento paralelo en algoritmos de procesamiento de imágenes, evaluando la eficiencia sobre sistemas multicore basados en plataformas FPGAs.

Palabras clave: *Procesamiento paralelo, eficiencia energética, tolerancia a fallos, arquitecturas paralelas, sistemas multicore, platadormas FPGAs, procesamiento de imágenes.*

Contexto

Las líneas de Investigación aquí desarrolladas son parte del Proyecto de Investigación Científico-Tecnológico “Búsqueda de Eficiencia en el Procesamiento Digital de Imágenes y Aplicaciones de Telemedicina” de la Universidad Nacional Arturo Jauretche (UNAJ).

Por otra parte, se tiene financiamiento en el marco del programa “Universidad, Diseño y Desarrollo Productivo” del Ministerio de Educación a través de varios proyectos aprobados en la UNAJ.

Parte de las líneas de investigación desarrolladas se encuentran enmarcadas en los convenios de colaboración en Actividades de Investigación firmados por la UNAJ con el Laboratorio CeTAD y el Instituto III – LIDI de la UNLP.

Introducción

La demanda de mayor poder de cómputo ha llevado a un límite en cuanto a la frecuencia de funcionamiento de los procesadores. Por otro lado, se han implementado una innumerable cantidad de técnicas para obtener una alta eficiencia en la ejecución de

instrucciones. Finalmente, la búsqueda de explotar al máximo el paralelismo brindado por las arquitecturas multicore, requiere el agregado de complejidad que permita controlar el consumo energético y lograr eficiencia en dichos sistemas [1] [2].

Eficiencia energética en Arquitecturas de Procesamiento paralelo

La creciente preocupación por el consumo de los recursos energéticos ha llevado a intentar hacer un uso responsable de la energía, debido al coste económico y a las consecuencias ambientales que conlleva.

Particularmente en los sistemas de cómputo de altas prestaciones, el tiempo de uso tiene un alto coste asociado al consumo de energético, por lo que se debe evaluar la eficiencia del consumo, evitando la mejora de las prestaciones a cualquier precio. Los sistemas comenzaron a ser valorados bajo el paraguas denominado “Green Computing” que intenta evitar los daños resultantes del alto consumo de energía de las computadoras.

En el contexto de la utilización, cada vez más difundida, de las nuevas arquitecturas multiprocesador, surge la necesidad de analizar no sólo su rendimiento al usar diferentes modelos de programación paralela, sino también establecer el grado de eficiencia en el uso de la energía [3] [4].

Factores de Fallas en Arquitecturas Paralelas

El aumento en la escala de integración, con el objetivo de mejorar las prestaciones de los procesadores, sumado al crecimiento de los sistemas de cómputo, ha producido que las

arquitecturas multicore presenten una creciente vulnerabilidad a fallos causados por interferencias provenientes del ambiente. Estos fallos se vuelven significativos a causa de su capacidad de corromper los resultados de las aplicaciones. Su impacto es mayor en los sistemas de cómputo de altas prestaciones, donde se ejecutan aplicaciones paralelas de cómputo intensivo y de gran duración. El alto costo de relanzar una ejecución desde el comienzo, en caso de que un fallo produzca la finalización de la aplicación con resultados incorrectos, provoca la necesidad de contar con estrategias para mejorar la fiabilidad de estos sistemas [5].

Plataformas FPGAs para procesamiento paralelo.

La implementación de paralelismo en plataformas FPGAs consiste en el uso de procesadores embebidos para ejecutar la aplicación, y en la concurrencia que provee la lógica programable para manejar las porciones de código que se pueden ejecutar de dicha forma [6].

La posibilidad de implementar procesadores embebidos en forma rápida, junto con la posibilidad de obtener concurrencia mediante la programación de HW permiten combinar el paralelismo con los sistemas multicore en virtud de buscar la máxima eficiencia, lo cual aparece como un gran desafío para lograr la optimización en el procesamiento de imágenes. Esto se debe a que cada algoritmo es un caso de estudio, en el que se debe decidir qué parte se implementa de manera concurrente y cuál de forma secuencial en el sistema multicore. Para la implementación de estos sistemas complejos lo ideal es combinar la programación en VHDL con lenguajes de alto nivel y aplicaciones como el core-

generator [7]. Con las mejoras constantes que aporta la evolución de la tecnología sobre las FPGAs, se logran diseños de gran magnitud, a tal punto que la tendencia actual es implementar microprocesadores de propósito general y todo el hardware de propósito específico que requiere la aplicación.

Paralelismo en el Procesamiento de Imágenes.

En el espacio frecuencial existen dos métodos muy potentes para buscar eficiencia en el procesamiento de imágenes mediante el procesamiento paralelo.

La paralelización mediante el *método de FFT de vector múltiple* [8], calcula la DFT de los vectores columna de una matriz de forma simultánea. La suma de las multiplicaciones de todas las iteraciones demuestra que gran parte de las multiplicaciones totales son triviales. Esto hace que este algoritmo sea extremadamente eficiente. Además, este método admite un grado de paralelización, implementando rutinas separadas de *butterfly* en procesadores diferentes para los casos especiales de la multiplicación realizada en el cálculo del algoritmo.

Por otro lado, la estructura de procesamiento paralelo más importante en aplicaciones de PDI es la *Red Neuronal (RN)*[9]. El cálculo de la DFT mediante una RN se basa necesariamente en la definición de la DFT mediante matrices. Esto es debido a que la estructura neuronal no puede ejecutar un algoritmo (p. e. la FFT) sobre la señal, dado que, por definición, un algoritmo es de carácter secuencial, mientras que la RN es una estructura de procesamiento en paralelo.

El principal problema de calcular una DFT directa mediante una RN se centra

en que los coeficientes para cada una de las muestras (píxeles en la imagen) son números complejos, por lo que su almacenamiento en los pesos de la red no puede realizarse de manera conjunta. En lugar de ello, hay que dividir la red para calcular la parte real y la parte imaginaria por separado, y esto para cada una de las dimensiones que se traten. Entre los métodos más conocidos basados en RN se tiene la DFT de Hartley (DHT) y la red de programación Lineal de Hopfield.

Por otro lado en el plano espacial se tienen algoritmos basados en convolución, como la detección de bordes, que son propicios para la implementación en HW de manera concurrente. Además, gran parte de los algoritmos implementados en hardware para etiquetar objetos se basan en procesamiento paralelo (por ejemplo, los que realizan el procesamiento local de los píxeles generalmente asignan unidades de procesamiento a cada píxel); esta arquitectura garantiza un desempeño veloz del algoritmo pero demanda una alta complejidad computacional y un consumo costoso de recursos dentro del dispositivo FPGA [10].

Líneas de Investigación, Desarrollo e Innovación

Temas de Estudio e Investigación

- Estudio de problemas de consumo y eficiencia energética en arquitecturas de procesamiento paralelo
- Estudio de las posibles fallas que pueden presentarse en arquitecturas paralelas y distribuidas.
- Implementación de un sistema multicore en Dispositivos Lógicos Programables.

- Estudio de las características paralelizables de algoritmos de procesamiento de imágenes en el plano espacial y frecuencial.
- Búsqueda de eficiencia en el procesamiento de imágenes mediante su paralelización.

Resultados y Objetivos

Investigación experimental

- Lograr paralelizar eficientemente determinados algoritmos de procesamiento de imágenes mediante su implementación sobre plataformas multicore basadas en FPGAs.
- Verificar la eficiencia comparando tiempos de cómputo entre algoritmos de procesamiento de imágenes implementados de forma secuencial y sus paralelizaciones en sistemas multicore.

Formación de Recursos Humanos

Dentro de la temática de la línea de I/D se participa en el dictado de asignaturas de la carrera de Ingeniería Informática de la UNAJ.

En esta línea de I/D existe cooperación a nivel nacional. Hay 2 investigadores realizando su Doctorado y 1 realizando una Maestría.

Referencias

1. Grama A, Gupta A, Karypis G, Kumar V. "Introduction to parallel computing". Second Edition. Pearson Addison Wesley, 2003.
2. H Hennessy, J. L., Patterson, and D. A., Computer Architecture, Fourth Edition: A Quantitative Approach. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.,

- 2006.
3. Mc.Cool Michael, "Programming models for scalable multicore programming", <http://www.hpcwire.com/features/17902939.html> (2007)
4. Brown D. J., "Toward Energy-Efficient Computing", Magazine Communications of the ACM Volume 53 Issue 3, March 2010.
5. Rexachs, D., Luque, E.: High Availability for Parallel Computers. JCS&T Vol. 10 No. 3, 110 – 116 (2010).
6. C. Rupp, The NAPA Adaptive Processing Architecture. IEEE Symposium on FPGAs Custom Computing Machines. 1998.
7. "Model-Based Design with Simulink, HDL Coder, and Xilinx System Generator for DSP", Kiran Kintali and Yongfeng Gu, MathWorks, White Paper (<http://www.mathworks.com/fpga-design/simulink-with-xilinx-system-generator-for-dsp.html>).
8. Laurence R Johnson, Anil K. Jain, "An Efficient Two-Dimensional FFT Algorithm", IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. PAMI-3, NO. 6, NOVEMBER 1981, pag. 698-701.
9. Fernández Muñoz Alvaro, "Estudio comparativo de las técnicas de procesamiento Digital de Imágenes", 1999.
10. Crookes, D; Benkrid, K. An FPGA I Implementation of Image Component Labeling. The Queen's University of Belfast. Suzuki, Kenji; Horiba, Isao; Sugie, Noboru.