CARACTERIZACION DE PROBLEMAS A SOLUCIONAR MEDIANTE PROGRAMACIÓN MULTIPROCESO DE MEMORIA COMPARTIDA

Autor
ERIKA VIVIANA RIAÑO BEJAR

Director
JOSE ORLANDO MALDONADO BAUTISTA

DEPARTAMENTO DE INGENIERÍAS ELÉCTRICA ELECTRÓNICA SISTEMAS Y TELECOMUNICACIONES
FACULTAD DE INGENIERÍAS Y ARQUITECTURA
INGENIERIA DE SISTEMAS

UNIVERSIDAD DE PAMPLONA
PAMPLONA, FEBRERO de 2016

1. INFORMACIÓN GENERAL DEL TRABAJO DE GRADO

Título: CARACTERIZACION DE PROBLEMAS A SOLUCIONAR MEDIANTE PROGRAMACIÓN MULTIPROCESO DE MEMORIA COMPARTIDA							
Nombre Autor: **ERIKA VIVIANA RIAÑO BEJAR**						C.C. : 1093140884 FIRMA:	
E-mail: **e.vrb12@hotmail.com**						Teléfono: 3104715585	
Lugar de Ejecución del Proyecto: Pamplona, Norte de Santander							
Duración de Proyecto (en semanas): 16							
Modalidad:							
Investigación	X	Práctica Empresarial*		Diplomado		Docencia	
**Deberá adjuntar carta de aceptación de la empresa o copia del convenio o contrato*							

Objetivo General:

- Identificar las características propias de los problemas que pueden ser optimizados mediante la programación multiproceso de memoria compartida

Objetivos específicos:

- Estudiar el paradigma de programación multiproceso de memoria compartida mediante la API OpenMP
- Analizar y caracterizar los problemas que son susceptibles de ser optimizados mediantes este paradigma.
- Seleccionar problemas tipo dentro de la caracterización realizada y analizar el rendimiento de la implementación en paralelo frente al serial tradicional.

Valor Total del Proyecto:	
Aceptación director de la tesis: **NOMBRE:** **FIRMA:** *Acepto y certifico que revisé el presente anteproyecto antes de firmarlo.*	**Autorización director del programa: SI NO** **FIRMA:**
Jurado 1:	**Firma Jurado 1:**
Jurado 2:	**Firma Jurado 2:**
Jurado 3:	**Firma Jurado 3:**

Sustentación Anteproyecto:
Lugar : _____ **Fecha:** _____ **Hora**_____
Aprobado ☐ **Incompleto** ☐ **Rechaza** ☐

Observaciones del jurado:

Firma Jurado 1: _____ **Firma Jurado 2:** _____ **Firma Jurado 3:** _____

Realización de Correcciones
Lugar : _____ **Fecha:** _____ **Hora**_____
Aprobado ☐ **Rechazado** ☐
Firma Jurado 1: _____ **Firma Jurado 2:** _____ **Firma Jurado 3:** _____

2. RESUMEN DEL PROYECTO

Con este proyecto se pretende estudiar el paradigma de programación multiproceso de memoria compartida mediante la API openMP con el fin de analizar y caracterizar diferentes problemas que son susceptibles de ser optimizados a dicho paradigma y partiendo de dicha caracterización observar el rendimiento de la implementación en paralelo frente a la tradicional.

2.1. Palabras claves:

Computación de alto rendimiento, programación en paralelo, OpenMP

3. PLANTEAMIENTO DEL PROBLEMA.

Actualmente se ha venido desaprovechando gran parte del potencial de las arquitecturas modernas que nos ofrecen las diferentes marcas de pc, ya que a pesar de que estas cuenten con varios procesadores y/o núcleos el algoritmo programado se ejecuta como si fuese un equipo monoprocesador, es por ello que se hace necesario la utilización de herramientas tales como openMP que provee directivos de compilador para la programación de multihilos, variables de ambiente y rutinas de biblioteca que controlan paralelismo en nivel de cores y nivel de procesadores, logrando con ello la caracterización de problemas computacionales susceptibles a solucionar mediante programación multiproceso de memoria compartida.

3.1. JUSTIFICACION.

En la actualidad la Universidad de Pamplona no cuenta con una infraestructura bien definida la que permita realizar procesos de alta complejidad, por tal motivo se desea analizar la optimización de problemas caracterizados mediante programación paralela en openMP.

4. MARCO TEÓRICO

Computación de Alto Rendimiento (High Performance Computing)

Es aquella que proporciona una mayor capacidad de cómputo que la que se puede obtener de computadores individuales.

- Sistemas de memoria compartida.
- Sistemas distribuidos.
- Máquinas paralelas.
    - Sistemas Masivamente Paralelos
    - Clúster

High Performance Computing o HPC. Introducción a la Computación de Alto Rendimiento Una computador paralelo es una máquina que tiene más de un procesador y por lo tanto tiene la capacidad de ejecutar más de un programa al mismo tiempo Centro de Cálculo (Diaz)

- Técnicas, investigación e infraestructura entorno a problemas con alta demanda de recursos computacionales.

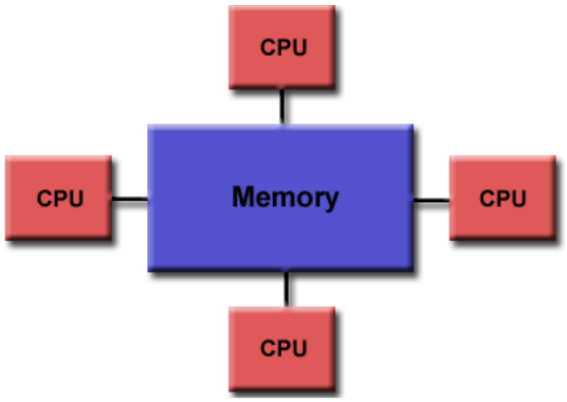
Comprende:

- Hardware: almacenamiento, procesamiento, redes
- Software: sistemas de archivos, programación paralela
- Aplicaciones: cálculos complejos, simulación, minería.

Maquinas Paralelas:

Memoria compartida:

- SMP (*SymmetricMulti-Processor*)
- Cada procesador posee su propia memoria caché, y está interconectado junto con otros procesadores a la memoria principal
- Direcciones de memoria comunes para todos los procesadores
- Mismos tiempos de respuesta
- Escrituras deben sincronizarse
- Cantidad de procesadores limitada (Roman, 2013)



Memoria distribuida:

- Direcciones de memoria separadas físicamente
- Lectura de un bloque de memoria local en forma inmediata
- Comunicación para leer memoria remota
- Bus de datos
- Red
- Tiempos de acceso a memoria son diferentes (local vs remoto)
- Permite conectar muchas máquinas y tener muchos procesadores (Roman, 2013)

Híbridas:

- Fusión de los enfoques anteriores

  Por ejemplo, clúster SMP:

  - Múltiples máquinas de memoria compartida
  - Conexión de red de alta velocidad
  - Permiten utilizar varios paradigmas de programación

  (Roman, 2013)

Hilos

- Múltiples rutas de ejecución concurrentes
- Cada hilo:
  - Comparte todos los recursos del hilo principal
  - tiene datos locales
- Requieren sincronización
- Los hilos pueden crearse o destruirse
- El hilo principal debe permanecer activo durante toda la ejecución
- Asociado a máquinas de memoria compartida (Roman, 2013)

Paradigmas de programación: Es una clase de algoritmos que resuelven problemas diferentes pero tiene la misma estructura de control. Estos programas son llamados esqueletos algoritmos o programas genéricos, a partir de estos esqueletos se pueden derivar programas modelos que ilustren como el paradigma resuelve problemas específicos. (M. Rossainz López)

Paradigmas de programación paralela

Existen varias formas de programar en paralelo:

- Paso de mensajes
- Memoria compartida
- Tareas Paralelismo de datos
- Operaciones remotas en memoria
- Modelos combinados

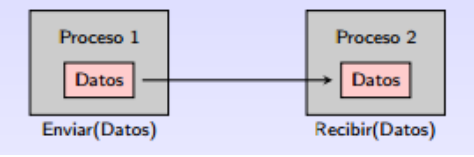
Todas estas formas:

- NO son excluyentes entre sı
- NO tienen nada que ver con el ordenador (Hermanns, 2007)

Modelo de paso de mensajes

Características del paradigma de programación:

- Un conjunto de procesos que disponen de memoria local
- Los procesos intercambian datos mediante mensajes
- El emisor y receptor del mensaje tienen que colaborar

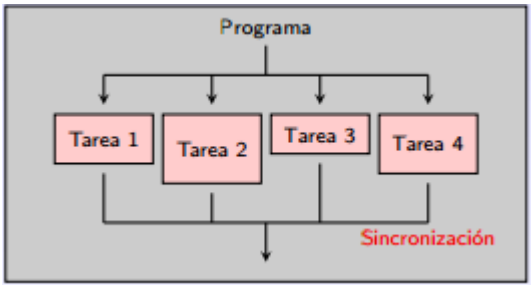


El programador es responsable del envío y la recepción de mensajes, que típicamente se hacen mediante llamadas a una librería

Modelo de tareas:

Un programa serial define un conjunto de tareas

- Cada tarea dispone de su memoria local
- Cada tarea tiene acceso a una memoria conjunta
- Las tareas son realizadas simultáneamente



El programador es responsable de la sincronización

El estándar OpenMP: Lo define el OpenMP Architecture Review Board
En 1997 se publicó la versión 1.0
En 2000 se publicó la versión 2.0
En 2005 se publicó la versión 2.5 (Hermanns, 2007)

Lo incorpora la mayoría de compiladores actuales

Selección del paradigma de programación:

- El tipo de ordenador paralelo que se vaya a utilizar
- Las características del problema/programa a paralelizar
- El tiempo disponible para la implementación paralela

Se selecciona el paradigma de programación apropiado:

- **Paradigma de paso de mensajes: MPI**
- **Paradigma de tareas: OpenMP (Hermanns, 2007)**

Paralelismo de datos:

- Divide el problema en subdominios
- Cada procesador trabaja sobre datos independientes
- Podría mejorar la escalabilidad (Roman, 2013)

Paralelismo de tareas:

- Identifica regiones o tareas del programa
- Se ejecutan en paralelo aquellas que son independientes
- Diferentes procesadores realizan diferentes funciones (Roman, 2013)

OpenMP:

Es una especificación para implementaciones portables de paralelismo en FORTRAN y C/C++.

Las especificación contiene provee directivos de compilador para programación de multihilos, variables de ambiente y rutinas de biblioteca que controlan paralelismo

- Nivel de Cores
- Nivel de Procesadores

Soporta el modelo de paralelismo de datos
Paralelismo Incremental
Combina código serial y paralelo en un solo código fuente. (The OpenMP® , s.f.)

5. DELIMITACION

5.1 OBJETIVO GENERAL

- Identificar las características propias de los problemas que pueden ser optimizados mediante la programación multiproceso de memoria compartida

5.2 OBJETIVOS ESPECÍFICOS

- Estudiar el paradigma de programación multiproceso de memoria compartida mediante la API OpenMP
- Analizar y caracterizar los problemas que son susceptibles de ser optimizados mediantes este paradigma.
- Seleccionar problemas tipo dentro de la caracterización realizada y analizar el rendimiento de la implementación en paralelo frente al serial tradicional.

5.3. ACOTACIONES

Los problemas tipos serán seleccionados de otras áreas de afines a las líneas de los investigadores del grupo CICOM: tratamiento de imágenes, modelado y simulación, tratamiento de imágenes, etc.

6. TAREAS Y CRONOGRAMA DE ACTIVIDADES

1) Estudio de los conceptos básicos de Computación de alto rendimiento

2) Análisis de los conceptos básicos en programación multiproceso de memoria compartida con OpenMP. Estudio y compilación de de problemas básicos.
3) Exploración de herramientas de desarrollo, compiladores, IDEs.
4) Estudio en profundidad de OpenMP: Directivas, Clausulas, Parámetros
5) Identificación de características de los problemas susceptibles de optimizar
6) Selección de problemas tipo y ejemplos de prueba. codificación, pruebas y análisis de rendimiento.
7) Documentación y compilación del informe.
8) Socialización de resultados

6.1 Cronograma y descripción de Actividades.

ACTIVIDADES	SEMANAS																	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1																		
2																		
3																		
4																		
5																		
6																		
7																		
8																		

Tabla 6.1. Cronograma y descripción de Actividades

7 RESULTADOS/PRODUCTOS ESPERADOS Y POTENCIALES BENEFICIARIOS

7.1. Impacto esperados a partir del uso de los resultados:

Tabla 7.1. Impactos esperados después de finalizar el proyecto

Impactos	Plazo	Indicador	Supuestos
Sociales	Corto	Socialización de resultados	Los resultados serán socializados ante el semillero de investigación, los estudiantes del problema y miembros de grupos a fines a grupo CICOM, con el objeto de incentivar el uso y profundización en la computación de alto rendimiento
	Mediano	Publicación de resultados	La documentación generada y el reporte técnico estará disponible para consulta en linea desde la web del programa, con el fin de facilitar la continuidad del trabajo.
	Mediano	Desarrollo de proyectos de aplicación	Con la caracterización de problemas y la socialización de los mismos a la comunidad se espera determinar problemas de aplicación en los diferentes grupos de investigación que puedan ser abordados desde el paradigma

Bibliografía

Diaz, G. (s.f.). *Plataformas Actuales para computacion de alto rendimiento.* Merida,Venezuela.

Hermanns, M. (2007). *Introduccion a la Programacion Paralela.* España: Universidad Politecnica de Madrid.

M. Rossainz López, J. V. (s.f.). *Implementación de Patrones de Diseño Paralelos En Java Como Una Bibliooteca De Clases De Objetos Paralelos.* Granada, España.

MÚNERA, A. M. (2009). *TESIS MÉTODO PARA EL MANEJO DEL BALANCEO DE CARGA EN SISTEMAS DE CÓMPUTO DISTRIBUIDO DE ALTO DESEMPEÑO.* Medellin, Colombia: UNIVERSIDAD NACIONAL .

Roman, R. (Abril de 2013). *CENTRO NACIONAL DE ALTA TECNOLOGIA*. Obtenido de FUNDAMENTOS DE COMPUTACION PARA CIENTIFICOS: http://www.cenat.ac.cr/descargas/TallerCNCA-CIEMIC_2013-04/6-Computacion%20de%20alto%20rendimiento.pdf

Saez Fernando, P. F. (s.f.). *Paradigmas De Programacion Paralela.* San Luis,Argentina: Universidad Nacional De San Luis.

Sánchez, C. R. (2010). *Tesis Algoritmos Paralelos para la resolución de problemas de mínimos cuadrados basados en transformaciones ortogonales sobre GPUs y multiprocesadores.* España: Universidad Politecnica De Valencia.

The OpenMP® . (s.f.). Obtenido de API specification for parallel programming: http://openmp.org/wp/