

TABLA DE CONTENIDOS.

	página
Dedicatoria.	I
Agradecimientos.	II
Tabla de Contenidos.	III
Índice de Figuras.	VI
Índice de Tablas.	VIII
Resumen.	IX
Abstract.	x
1. Introducción.	1
2. El Problema.	3
2.1. Introducción.	3
2.2. Descripción del Problema.	4
2.3. Descripción del Contexto.	4
2.4. Objetivos.	6
2.4.1. Objetivo General.	6
2.4.2. Objetivos Específicos.	6
2.5. Resumen.	7
3. Marco Teórico.	8
3.1. Introducción.	8
3.2. Problemas de Alto Desempeño Computacional.	8
3.3. Cluster.	9
3.4. Computación Grid.	9
3.5. Computación Paralela y Distribuida.	10
3.5.1. Análisis de Desempeño.	15
3.5.2. Metodología de programación paralela.	19

3.6.	La Unidad de Procesamiento Central.	28
3.7.	La Unidad de Procesamiento Gráfico.	30
3.7.1.	Especificaciones generales de la GF100.	33
3.8.	Programación en CUDA C.	37
3.8.1.	Términos clave.	37
3.8.2.	Funciones claves.	40
3.8.3.	Programación de la GPU.	43
3.8.4.	Limitaciones.	52
3.8.5.	Funciones Matemáticas.	54
3.9.	Trabajos Actuales.	54
3.10.	Daisyworld.	55
3.11.	Resumen.	57
4.	Metodología.	58
4.1.	Introducción.	58
4.2.	Metodología de Programación Paralela.	58
4.3.	Metodología de desarrollo.	58
4.3.1.	Estudio de casos prácticos.	59
4.3.2.	Obtención de las mediciones.	59
4.3.3.	Plan de Pruebas.	60
4.3.4.	Evaluación de métricas.	61
4.3.5.	Comparaciones de implementaciones.	62
4.3.6.	Herramientas.	62
4.4.	Resumen.	64
5.	Implementación.	65
5.1.	Introducción.	65
5.2.	Selección de un problema de HPC.	65
5.3.	Formato de la función kernel.	67
5.4.	Llamado al kernel.	72
5.5.	Resumen.	76
6.	Resultados.	77
6.1.	Introducción.	77
6.2.	Problemas de Prueba.	77

6.3. Plan de Pruebas.	78
6.4. Mejoras en Implementaciones.	88
6.5. Resumen.	97
7. Conclusiones.	98
Bibliografía.	100
Anexos.	
A: Errores CUDA.	104

ÍNDICE DE FIGURAS.

	página
3.1. Programa Secuencial	11
3.2. Programa paralelo v/s programa secuencial	13
3.3. Ley de Amdahl	17
3.4. Metodología de descomposición de un problema para ser paralelizado.	20
3.5. Problemas en método de descomposición exploratorio.	24
3.6. Árbol de dependencia entre tareas.	26
3.7. Problemas de ociosidad producida por una mala elección de las tareas.	26
3.8. Una mejor solución al problema de la ociosidad de las tareas.	27
3.9. Diagrama general de una CPU.	29
3.10. Crecimiento de las capacidades de procesamiento de las CPU's y GPU's.	31
3.11. Jerarquía de memorias.	32
3.12. Modelo general de arquitectura GF100.	34
3.13. Streaming Multiprocessor en GF100.	35
3.14. Cuda Core en GF100.	36
3.15. Flujo de datos en CUDA.	38
3.16. Composición de un ejecutable compilado con nvcc.	44
3.17. Esquemas generarles de CPU y GPU.	46
3.18. Kernel. Grid con bloques de threads.	49
3.19. Diferentes estados del planeta.	56
4.1. Ejemplo de ejecución de una rutina Cuda C.	63
5.1. Distribución de temperaturas y albedos.	67
6.1. Tiempo de ejecución de la implementación secuencial en función del tamaño del problema.	79
6.2. Tiempo de Ejecución de la implementación paralela con Pthreads en función del número de threads.	80
6.3. Tiempos de Ejecución del programa paralelo con Pthreads con 1, 2 y 4 threads.	81
6.4. Tiempos de Ejecución del programa paralelo con Pthreads con 1, 2 y 4 threads para matrices de hasta 124x124.	81

6.5. Tiempo de ejecución de la implementación con Pthreads en función del tamaño del problema.	82
6.6. Tiempo de Ejecución de la implementación paralela con la GPU en función del número de threads.	84
6.7. Tiempos de Ejecución del programa paralelo con la GPU con 128, 256 y 512 threads.	85
6.8. Tiempo de ejecución de la implementación paralela con la GPU. . . .	85
6.9. Comparación del tiempo de ejecución de las implementaciones paralelas con la configuración óptima de threads.	86
6.10. Comparación del tiempo de ejecución de las implementaciones paralelas con la configuración óptima de threads.	87
6.11. Matriz de temperaturas de ejemplo, de dimensión 4x4.	88
6.12. Comparación de los tiempos de ejecución de la implementación paralela con Pthreads en función del tamaño del problema.	91
6.13. Comparación de los tiempos de ejecución de la implementación paralela con la GPU en función del tamaño del problema.	91
6.14. Comparación del porcentaje de reducción del tiempo de ejecución, de las implementaciones paralelas tras las mejoras.	92
6.15. Aceleraciones de las implementaciones paralelas	95
6.16. Eficiencia experimentada por la implementación paralela con Pthreads.	95
6.17. Eficiencia experimentada por la implementación paralela con la GPU.	96
6.18. Aceleración de la implementación paralela con la GPU.	96

ÍNDICE DE TABLAS.

	página
3.1. Especificaciones técnicas de las tarjetas Nvidia.	21
3.2. Matriz de términos claves en funciones.	39
3.3. Enumeradores para la transferencia de datos entre memorias.	42
3.4. Dimensiones de los parámetros de configuración del kernel Cuda C. .	51
3.5. Funciones Matemáticas Cuda Afectadas por flag de compilación. . . .	54
3.6. Proyectos actualmente desarrollados.	55
4.1. Características del equipo.	60
6.1. Tiempos de Ejecución del programa secuencial.	78
6.2. Tiempos de Ejecución del programa paralelo con Pthreads.	80
6.3. Tiempos de Ejecución del programa paralelo con la GPU.	83
6.4. Constantes de la regresión polinomial de segundo orden.	86
6.5. Elementos comunes calcular promedio de las celdas vecinas de celdas contiguas.	89
6.6. Tiempos de Ejecución del programa paralelo con Pthreads con mejoras aplicadas versus el tiempo inicial de ejecución. Expresado en segundos.	90
6.7. Tiempos de Ejecución del programa paralelo con la GPU con mejoras aplicadas versus el tiempo inicial de ejecución.	90
6.8. Constantes de la regresión polinomial de segundo orden, tras las mejoras.	93
6.9. Matriz de comparación de resultados del problema.	93
6.10. Comparación de las aceleraciones de las implementaciones paralelas. .	94
A.1. Lista de posibles errores cuda	105