

UNIVERSIDAD CATÓLICA DEL MAULE

Facultad de Ciencias de la Ingeniería

Escuela de Ingeniería Civil Informática

Profesor Guía

Nombre_profesor_Guía

TÍTULO DE LA TESIS

NOMBRE(S) ALUMNO(S) TESISISTA(S)

Tesis para optar al
Título Profesional de Ingeniero Civil Informático

Talca, MES 20XX

UNIVERSIDAD CATÓLICA DEL MAULE
FACULTAD DE CIENCIAS DE LA INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL INFORMÁTICA

TESIS PARA OPTAR AL
TÍTULO PROFESIONAL DE INGENIERO CIVIL INFORMÁTICO

“TÍTULO DE LA TESIS”
NOMBRE(S) ALUMNO(S) TESISISTA(S)

COMISIÓN EXAMINADORA

FIRMA

PROFESOR GUÍA

NOMBRE_PROFESOR_GUÍA

PROFESOR COMISIÓN

NOMBRE_PROFESOR_COMISIÓN

PROFESOR COMISIÓN

NOMBRE_PROFESOR_COMISIÓN

NOTA FINAL EXAMEN DE TÍTULO

TALCA, MES DE 20XX

Sumario

resumen trabajo.... (máximo 3 páginas)

A continuación algunos ejemplos de figuras, referencia, cita, tabla y algoritmo:

Una referencia a la Figura 1 y a las subfiguras 2(a) y 2(b). Una cita a libro de Pthreads [?] y OpenMP ([?]).

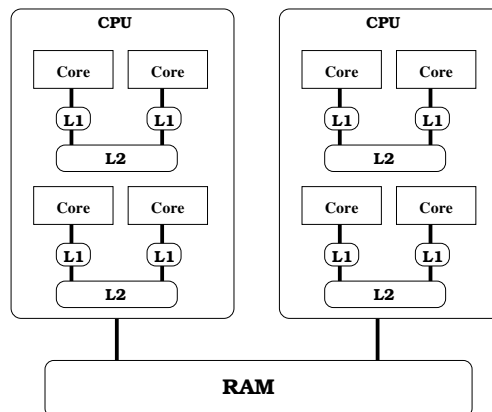


Figura 1: Plataforma multi-core.

```

int tid = IDThread
...
int ref1 = myArray[tid] * 1;
__syncthreads();
myArray[tid + 1] = 2;
__syncthreads();
int ref2 = myArray[tid] * 1;
result[tid] = ref1 * ref2;
...

```

(a) Caption Subfigura 1

```

int tid = IDThread
...
if (tid < warpSize) {
  int ref1 = myArray[tid] * 1;
  myArray[tid + 1] = 2;
  int ref2 = myArray[tid] * 1;
  result[tid] = ref1 * ref2;
}
...

```

(b) Caption Subfigura 2

Figura 2: Ejemplos para ilustrar la sincronización de los threads de un *warp*.**Algoritmo 1** *EGNAT*: búsqueda por rango r para la consulta q .

busquedarango(Nodo P , Consulta q , Rango r)

```

1: {Sea  $R$  el conjunto resultado}
2:  $R \leftarrow \emptyset$ 
3:  $d \leftarrow dist(p_0, q)$ 
4: if  $d \leq r$  then
5:   se reporta  $p_0$ 
6: end if
7:  $range(p_0, q) \leftarrow [d - r, d + r]$ 
8: for all  $x \in P$  do
9:   if  $range(p_0, q) \cap range(p_0, D_{p_x}) \neq \emptyset$  then
10:    se agrega  $x$  a  $R$ 
11:    if  $dist(x, q) \leq r$  then
12:      se reporta  $x$ 
13:    end if
14:  end if
15: end for
16: for all  $p_i \in R$  do
17:   busquedarango( $D_{p_i}, q, r$ )
18: end for

```

Tabla 1: Características Generales

Processor	2xIntel Quad-Xeon (2.66 GHz)
L1 Cache	8x32KB + 8x32KB (inst.+data) 8-way associative, 64byte per line
L2 Unifed Cache	4x4MB (4MB shared per 2 procs) 16-way associative, 64 byte per line
Memory	16GBytes (4x4GB) 667MHz DIMM memory 1333 MHz system bus
Operating System	GNU Debian System Linux kernel 2.6.22-SMP for 64 bits

Índice general

1. Introducción	1
1.1. Objetivos	1
2. Estado del Arte	2
2.1. Tipos de paralelismo	2
2.1.1. bit	2
3. Marco Teórico	3
4. Desarrollo	4
4.1. Introducción	4
5. Experimentos	5
6. Conclusiones	6
6.1. Trabajos Futuros	6
6.2. Contribuciones de la Tesis	6

Índice de figuras

1.	Plataforma multi-core.	3
2.	Ejemplos para ilustrar la sincronización de los threads de un <i>warp</i>	4

Índice de tablas

1.	Características Generales	4
----	-------------------------------------	---

Índice de Algoritmos

1.	<i>EGNAT</i> : búsqueda por rango r para la consulta q	4
----	--	---

Capítulo 1

Introducción

texto...

1.1. Objetivos

El objetivo general de esta tesis es

Los objetivos específicos son los siguientes:

- objetivo-especifico-1
- objetivo-especifico-2
- objetivo-especifico-N

Capítulo 2

Estado del Arte

El paralelismo en computación es una técnica de programación la cual permite que varias instrucciones se puedan ejecutar simultáneamente. Se base en el principio de la división de grandes problemas en varios problemas mas pequeños, que son resueltos de forma concurrente.

2.1. Tipos de paralelismo

Existen diversos tipos de niveles de paralelismo que se describiran a continuación:

2.1.1. Nivel de bit

En la decada de 1970 hasta alrededor de 1986, la aceleración en la arquitectura de computadores lo logro aumentar el tamaño de una palabra, reduciendo asi el numero de instrucciones que un procesador debia realizar, de este modo se logro por ejemplo reducir el proceso de instrucciones en un procesador de 8 bits al sumar dos números de 16 bits.

2.1.2. Nivel de instrucción

Los avances en esta área se realizaron alrededor de 1980 y 1990, donde esto constaba con el reordenamiento y combinación de instrucciones en grupos que luego son ejecutadas en paralelo sin cambiar el resultado del programa. Actualmente los procesadores cuentan con "pipeline" de instrucciones de varias etapas, Cada etapa en el pipeline corresponde a una acción diferente que

el procesador realiza en la instrucción correspondiente a la etapa; un procesador con un pipeline de N etapas puede tener hasta n instrucciones diferentes en diferentes etapas de finalización.

2.1.3. Datos

2.1.4. Tareas

Capítulo 3

Marco Teórico

bla bla....

Capítulo 4

Desarrollo

4.1. Introducción

texto...

Capítulo 5

Experimentos

texto...

Capítulo 6

Conclusiones

texto...

6.1. Trabajos Futuros

- uno...
- dos...
- N...

6.2. Contribuciones de la Tesis

- uno...
- dos...
- N...