

Programación Paralela y Distribuida

Licenciatura en
Ciencias de la Computación
UNCuyo – Facultad de Ingeniería



Programación Paralela y Distribuida

- Docente
 - Dra. Paola Caymes Scutari



Docente e Investigadora en el LICPaD
Laboratorio de Investigación
en Cómputo Paralelo/Distribuido
UTN-FRM y CONICET



Docente en la carrera de
Lic. en Ciencias de la Computación
UNCuyo - FIng

Paralelismo

- ¿Por qué recurrir al Cómputo Paralelo?
- ¿No es suficiente con un solo procesador para resolver cualquier problema? **¡NO!**

- **Grandes Problemas:**
 - Cambio global
 - Genoma humano
 - Desastres ecológicos
 - Incendios forestales
 - Corrientes marinas
 - Modelos para superconductividad
 - Visión artificial
 - Servidores
 - Bases de datos
 - Gráficas online
 - ...

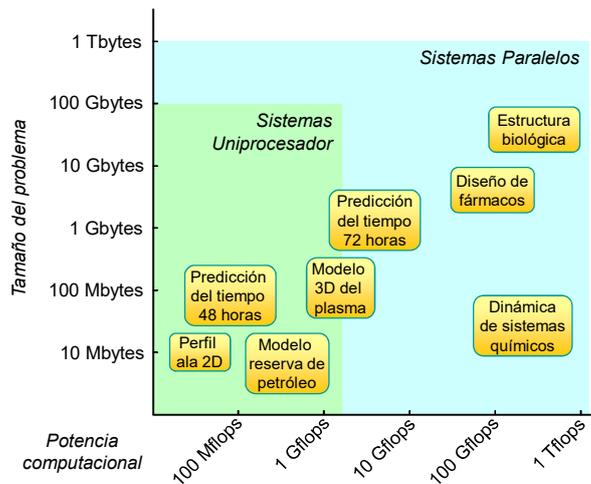
¡Constituyen necesidades reales!



3

Paralelismo

Grandes Problemas – Altos requerimientos



4



Programación Paralela y Distribuida

Introducción

Unidad 1



Contenido

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

6

[Contenido]

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

7

[Introducción]

- ¿Qué se entiende por ...
 - ...Cálculo Paralelo?
 - ...Cómputo Paralelo?
 - ...Procesamiento Paralelo?
- A grandes rasgos y en general, son términos utilizados indistintamente
- Puede definirse de la siguiente manera:

El cálculo paralelo es el método mediante el cual se divide un gran problema en componentes, tareas o cálculos más pequeños que pueden resolverse en paralelo, es decir ***“al mismo tiempo”***.

8

Introducción

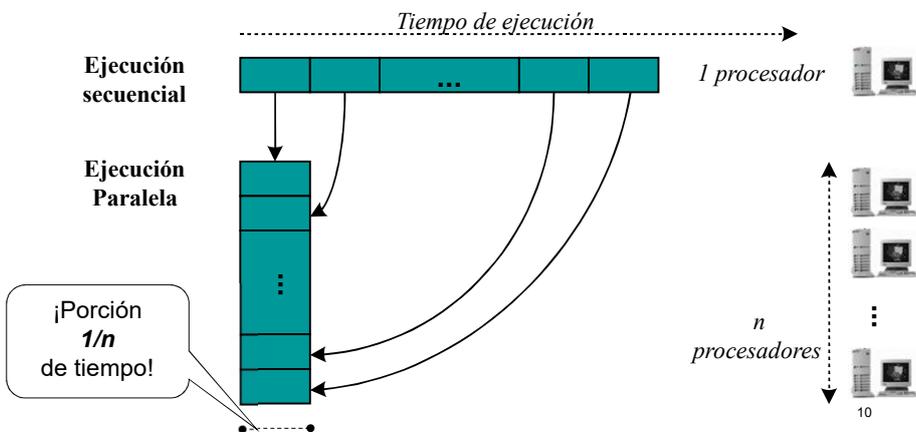
- La idea general y básica del cómputo paralelo es que n procesadores o nodos deberían proveer una velocidad computacional n veces mayor a la provista por un nodo simple, es decir:

El problema debería ser resuelto en un período $1/n$ del tiempo insumido por un uniprocador

9

Introducción

Gráficamente:



Introducción

- ¿Por qué surge el Cálculo Paralelo?
 - Demanda de rendimiento computacional
 - Cada vez deben resolverse problemas más complejos
 - El desempeño de las computadoras determina la clase de problemas que pueden resolverse. Existen problemas tan complejos que su solución podría tomar días, meses o años.
 - Disciplinas como la Biología, la Química, la Física, etc. constituyen los principales productores y usuarios del cálculo paralelo
 - Determinación del genoma humano
 - Evolución del Universo
 - Modelos de la naturaleza
 - Seguridad automotor
 - Aplicaciones multimedia
 - ...
 - Se requieren recursos proporcionales a:
 - el volumen de datos que debe procesarse
 - la complejidad de los cálculos que deben efectuarse sobre los datos

Introducción

- En resumen:
 - **Deben combinarse:**
 - sofisticadas aplicaciones
 - potentes sistemas
 - **...para resolver el problema**
 - en el menor tiempo posible
 - realizando un buen aprovechamiento de los recursos utilizados

Introducción

- La creciente necesidad de contar con sistemas y computación de alto rendimiento (*High Performance Computing - HPC*), ha orientado la atención de los científicos hacia los Sistemas Paralelo/Distribuidos
 - Las aplicaciones de gran envergadura necesitan más poder computacional del que una computadora secuencial puede ofrecer
 - La velocidad del procesador y demás componentes está limitada por la velocidad de la luz, leyes termodinámicas y el alto costo de fabricación
- Un sistema paralelo está formado por varios procesadores, y es capaz de realizar procesamiento paralelo
 - (El procesamiento paralelo es el método mediante el cual se divide un gran problema en componentes, tareas o cálculos más pequeños que pueden resolverse en paralelo, es decir "al mismo tiempo".)

13

Introducción

- Sistemas computacionales paralelos
 - Permiten que múltiples procesadores compartan una tarea computacional
 - Esta definición incluye diferentes clases de máquinas paralelas:
 - Una máquina con cientos o miles de procesadores (MPP)
 - Un conjunto de workstations o nodos conectados mediante una LAN (sistema distribuido)
 - Cualquiera de estas configuraciones debería proveer un **incremento significativo** en el rendimiento cuando se compara con un uniprosesor

La porción $1/n$ mencionada anteriormente

14

Introducción

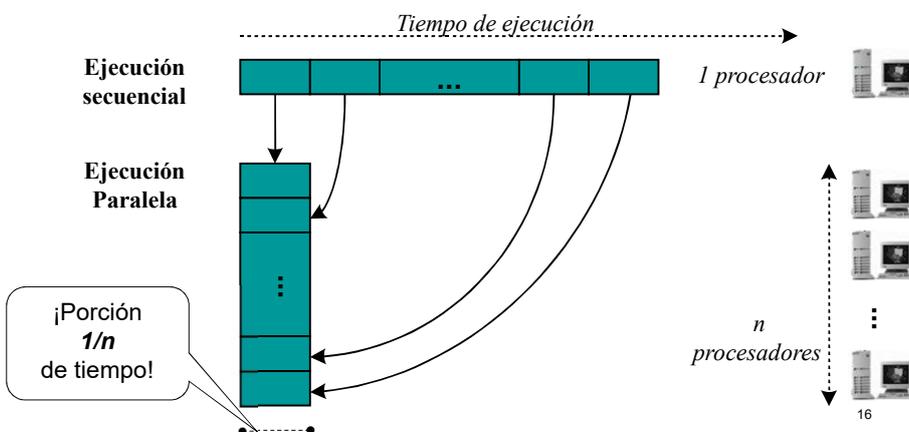
- La idea general y básica del cómputo paralelo es que n procesadores o nodos deberían proveer una velocidad computacional n veces mayor a la provista por un nodo simple, es decir:

El problema debería ser resuelto en un período $1/n$ del tiempo insumido por un uniprocador

15

Introducción

Gráficamente:



[Introducción]

- El auge del procesamiento paralelo se potenció con dos importantes desarrollos:
 - **Procesadores Masivamente Paralelos** (*massively parallel processors - MPP*)
 - Cientos o miles de CPUs en un gran gabinete
 - Cientos de Giga-bytes de memoria
 - **Cómputo Distribuido**
 - Alternativa viable y accesible
 - Conexión de múltiples procesadores mediante una red
 - Coordinación de su poder computacional

17

[Introducción]

- La fortaleza de los sistemas distribuidos se debe a su **costo**
 - Puede utilizarse un conjunto de máquinas de las que ya se dispone y es viable de ser extendido
 - Laboratorios o aulas de informática
 - PCs o workstations en una oficina o empresa
 - ...
 - Pueden utilizarse los recursos en los momentos del día en que están ociosos
 - Durante la noche
 - Durante los intervalos de tiempo que el usuario no hace uso del procesador.
 - ...
- En resumen, pueden configurarse y administrarse los recursos existentes para hacer uso de ellos y así lograr High Performance Computing a bajo costo.**

18

Introducción

- Nos referimos a “*sistemas distribuidos*” ya que consideramos el área de los sistemas distribuidos conocida como **Clustering**
 - Varias máquinas interconectadas trabajan colectivamente para procesar instrucciones y datos.
- Clustering también es referido como **Sistemas Paralelos**
 - Se utiliza un sistema distribuido para implementar un sistema paralelo. Ello conlleva:
 - Hardware
 - Sistema operativo
 - Middleware
 - Aplicaciones

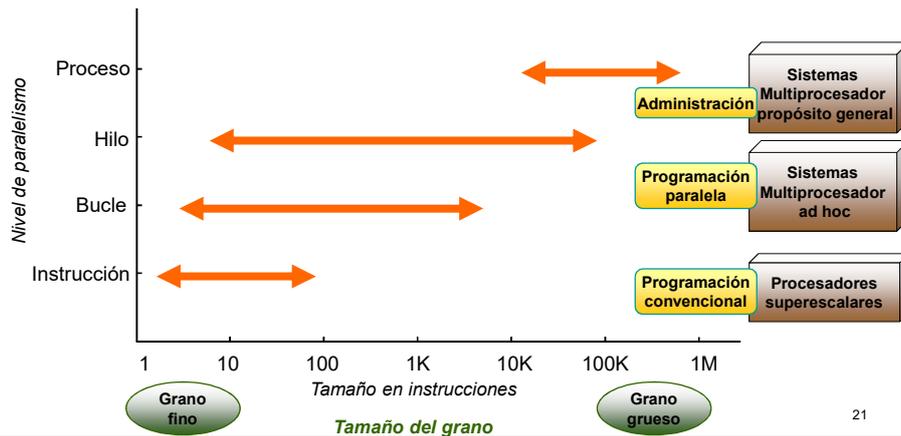
19

Introducción



Introducción

Niveles de paralelismo, instrucciones, granularidad, software y hardware



Contenido

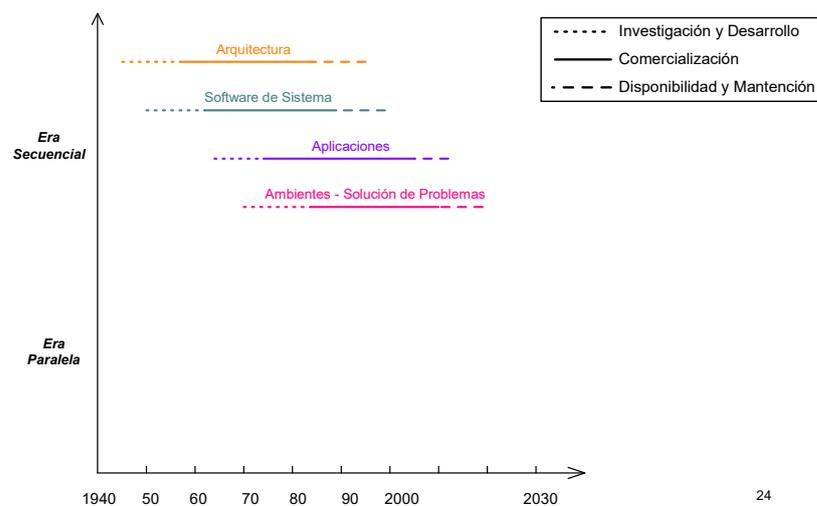
- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

[Un poco de Historia]

- **1945:** comienza la era de la computación
 - Computadoras grandes y muy costosas
 - Presentan limitaciones para la interconexión
- **1985:** los sistemas comienzan a evolucionar notoriamente
 - Desarrollo de microprocesadores más potentes
 - Invención de LAN (*Local Area Network*) de alta velocidad
 - Interconexión de varias máquinas (decenas o centenas)
 - Transferencia de porciones de información en milisegundos
- La “red” es el resultado de ambas tecnologías
 - Habilidad para crear redes de máquinas compuestas por un gran número de CPU's conectadas por una red de alta velocidad
- Estos sistemas se conocen como **Sistemas Distribuidos**
 - **Contrastan con los sistemas centralizados que constan de una sola CPU, memoria local y dispositivos de entrada/salida**

23

[Un poco de Historia]



24

Contenido

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

25

Motivación

- El desarrollo de software paralelo incluye
 - La coordinación de las tareas concurrentes y/o paralelas
 - La implementación de las acciones y comunicaciones paralelas
 - La dificultad de no contar con una fuerte base de
 - Algoritmos portables
 - Entornos estándares (hardware y software)
 - Herramientas de desarrollo
 - ...
- Por ello, constituye una tarea que presenta una demanda intensiva de **TIEMPO** y **ESFUERZO**.

26

[Motivación]

- Entonces...

¿Por qué emerge el Cómputo Paralelo?

- Veamos qué sugiere la Ley de Moore [Moo65] ...

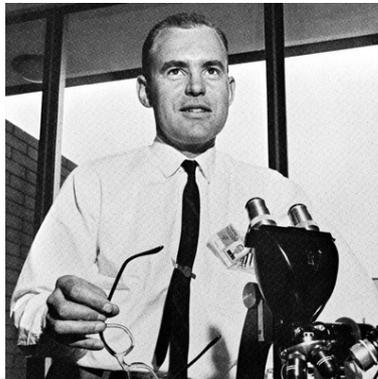
27

LICPaD (UTN-FRM)

Dr. Germán Bianchini - Dra. Paola Caymes Scutari

[Motivación]

- En 1965 Gordon Moore (uno de los fundadores de Intel) afirmó y publicó en la revista "*Electronics Magazine*" que :



LICPaD (UTN-FRM)

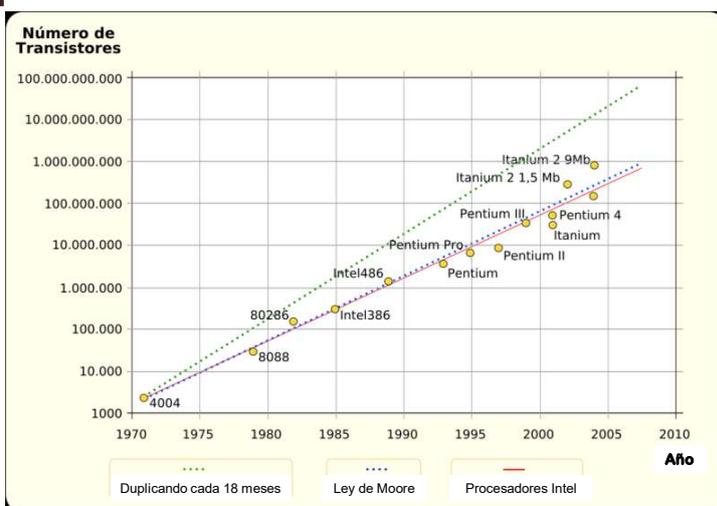
Dr. Germán Bianchini - Dra. Paola Caymes Scutari

Motivación

- La consecuencia directa de la Ley de Moore es que:
 - Los precios bajan al mismo tiempo que las prestaciones suben
 - la computadora que hoy vale 3000 dólares costará la mitad al año siguiente y estará obsoleta en dos años.
 - En 26 años el número de transistores en un chip se ha incrementado 3200 veces.

29

Motivación



30

[Motivación]

- Las limitaciones de esta ley fueron discutidas por el propio Moore y el famoso científico Stephen Hawking hace unos años, y éste último respondió que los límites dependían precisamente de los límites de la microelectrónica:
 - la velocidad de la luz
 - la naturaleza atómica de la materia.
- Esos dos límites no están muy lejos según Moore, que ha afirmado en 2007 que en 10 o 15 años su ley dejará de cumplirse [McD07].
- Según aseguró durante la conferencia en la que hizo su predicción afirmó, no obstante, que una nueva tecnología vendrá a suplir a la actual

31

[Motivación]

- Dadas las limitaciones de la Ley de Moore, en tanto no se utilice otra tecnología material, el paralelismo constituye un recurso para cubrir las crecientes demandas en cuanto a:
 - Poder computacional
 - Velocidad de disco y memoria
 - El solapamiento entre cómputo y operaciones de E/S permiten ocultar los límites intrínsecos de las operaciones que involucran dispositivos de E/S
 - Comunicación de datos
 - Especialmente en sistemas distribuidos, resulta de especial utilidad la paralelización de funcionalidades con el fin de evitar la centralización de los datos ya que puede consumir mucho tiempo dependiendo de la calidad de conexión.

32

[Motivación]

- Para ampliar la información:
 - Ver la siguiente demo:
 - http://www.intel.com/technology/product/demos/multi/demo.htm?iid=tech_multi-core+body_demo
 - Leer el material facilitado en web.

33

[Contenido]

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

34

Objetivos

- Sistemas computacionales paralelos
 - Su objetivo es permitir que múltiples procesadores compartan una tarea computacional con los siguientes fines:
 - Proveer la **potencia computacional** necesaria para tratar y resolver el problema
 - Reducir los **tiempos de ejecución**, acelerar la obtención de soluciones.
 - Esta definición incluye diferentes clases de máquinas paralelas:
 - Una máquina con cientos o miles de procesadores (MPP)
 - Un conjunto de workstations o nodos conectados mediante una LAN (sistema distribuido)
 - Cualquiera de estas configuraciones debería proveer un **incremento significativo** en el rendimiento cuando se compara con un uniprocador

35

Objetivos

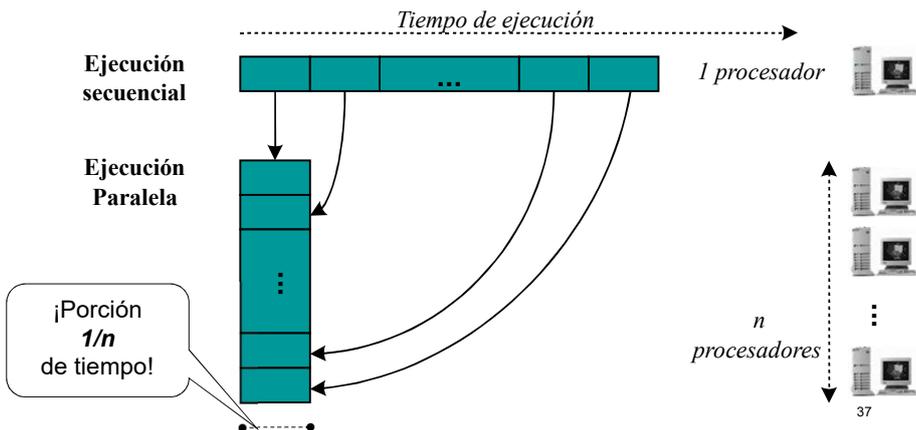
- La idea general y básica del cómputo paralelo es que n procesadores o nodos deberían proveer una velocidad computacional n veces mayor a la provista por un nodo simple, es decir:

El problema debería ser
resuelto en un período
 $1/n$
del tiempo insumido por un uniprocador

36

Objetivos

Gráficamente:



LICPaD (UTN-FRM)

Dr. Germán Bianchini - Dra. Paola Caymes Scutari

Objetivos

- Sin embargo, ésta es una situación ideal que en la práctica no siempre se cumple.
 - Límites de rendimiento teóricos o prácticos
 - No obstante son superiores a los límites de los uniprosesores.
 - Por ello, constituyen una plataforma ideal para aplicaciones que requieren altas prestaciones.
- Factores que limitan a las aplicaciones paralelas:
 - **Períodos** en los que algunos procesadores permanecen ociosos
 - **Cálculos redundantes** en cada nodo (recálculo de constantes, por ejemplo)
 - **Comunicación** entre procesos
 - Regiones del programa no paralelizables, como **inicialización** y **finalización** de la aplicación.

38

LICPaD (UTN-FRM)

Dr. Germán Bianchini - Dra. Paola Caymes Scutari

Objetivos

- Sean
 - f la porción de tiempo requerida para procesar la región secuencial del programa
 - $T(1)$ el tiempo de ejecución secuencial de la aplicación, es decir en un solo procesador
- El tiempo de ejecución con n nodos en paralelo será:

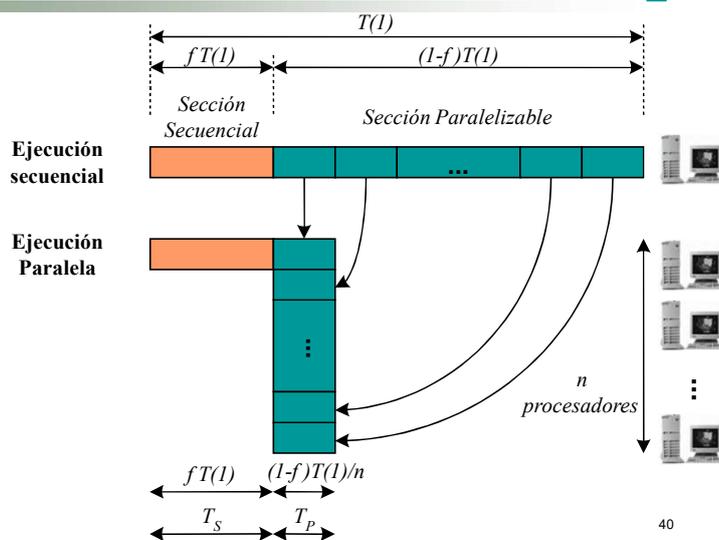
$$f * T(1) + (1 - f) \frac{T(1)}{n}$$

- Más intuitivamente...
 - T_S denota el tiempo insumido por la región secuencial
 - T_P denota el tiempo insumido por la región paralela
 - Y el tiempo de ejecución con n nodos será $T_S + T_P$

39

Objetivos

Gráficamente:



40

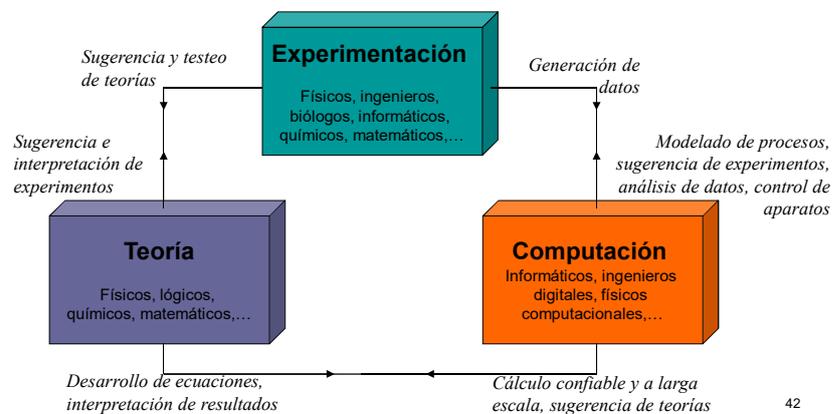
Contenido

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

41

Aplicaciones

- La resolución de problemas científicos de gran envergadura involucra tres disciplinas interactivas:

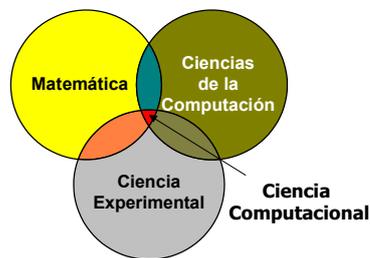


42

Aplicaciones

■ Ciencia Computacional

- Surge como producto de la interacción entre teoría, experimentación y computación



43

Aplicaciones

■ Ventajas de la ciencia computacional

- Los experimentos computacionales son más económicos y rápidos
 - Los experimentos físicos conllevan la utilización de materiales, equipos técnicos, etc.
 - En ocasiones se necesita mucho tiempo para el seguimiento de los experimentos (días, meses o años).
 - En algunas áreas de las ciencias, los experimentos son impracticables, ya sea por su dificultad, su peligrosidad o su costo.
- Puede resolver un rango mayor de problemas
 - En un laboratorio específico sólo podrán abordarse los problemas que puedan resolverse con el equipamiento existente
- Posee sólo límites de velocidad y de capacidad de memoria
 - Los experimentos físicos presentan diversas restricciones prácticas

44

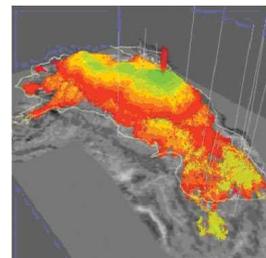
Aplicaciones

- De acuerdo a los requerimientos de procesamiento paralelo, pueden caracterizarse algunos grupos:
 - Modelado predictivo y simulación
 - Diseño industrial y robótica
 - Exploración de recursos energéticos
 - Investigación básica, médica y militar
 - Visualización

45

Aplicaciones

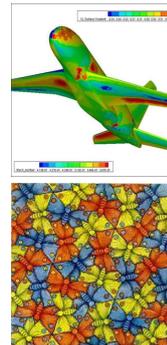
- Modelado predictivo y simulación
 - Características
 - Modelado multidimensional
 - Necesidad de incluir inteligencia computacional
 - precisión deseada
 - turnaround time
 - Ejemplos
 - Predicción de fenómenos climatológicos
 - Simulación de semiconductores
 - Oceanografía
 - Astrofísica (agujeros negros, evolución)
 - Determinación del genoma humano
 - Economía mundial
 - ...



46

Aplicaciones

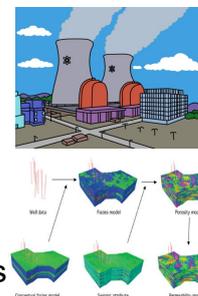
- Diseño industrial y robótica
 - Características
 - Problemas de diseño industrial (estructural o dinámico)
 - Necesidad de incluir inteligencia artificial a los sistemas
 - Robótica
 - Inteligencia artificial
 - Censado remoto
 - Ejemplos
 - Análisis de elementos finitos
 - Aerodinámica computacional
 - Aplicaciones de censado remoto
 - Inteligencia artificial y robótica
 - Procesamiento de imágenes
 - Reconocimiento de patrones
 - Visión por computador
 - Procesamiento del lenguaje
 - Sistemas computacionales expertos
 - Knowledge engineering
 - ...



47

Aplicaciones

- Exploración de recursos energéticos
 - Características
 - Menos costos de producción
 - Mayores medidas de seguridad
 - Ejemplos
 - Exploración sísmica
 - Modelado de yacimientos
 - Seguridad en reactores nucleares
 - ...



48

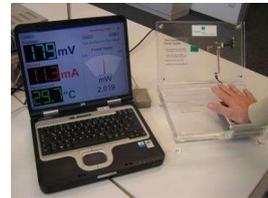
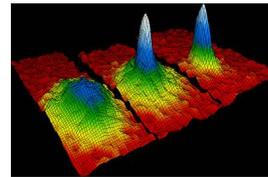
Aplicaciones

- Investigación básica, médica y militar

- Características
 - Gran capacidad y velocidad de procesamiento
 - Estimación de efectos

- Ejemplos

- Diagnóstico por imágenes
- Diseño de órganos artificiales
- Ingeniería genética
- Problemas de mecánica cuántica
- Química de polímeros
- Diseño de armas nucleares
- ...



Aplicaciones

- Visualización

- Características
 - Reducción de costos en la generación de escenarios
 - Habilidad para violar las reglas de la física

- Ejemplos

- Gráficos generados por computadora
- Películas y animaciones
 - The Matrix, Titanic, Toy Story, Shrek, ...
- Visualización de datos
- ...



Contenido

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

51

Dificultades

- Los sistemas paralelos presentan algunas **dificultades y debilidades**
 - La cantidad de software disponible es limitada
 - Dificulta la programación
 - Paralelismo no asegura rendimiento. Depende de:
 - la calidad de la programación
 - el estado entorno de ejecución (carga del sistema)
 - el conjunto de datos de entrada
 - Las comunicaciones pueden introducir problemas de confiabilidad o robustez
 - Pérdida de mensajes
 - Saturación de las comunicaciones
 - ...
 - Requiere soluciones de hardware (¡\$\$\$!)
 - Introduce problemas de seguridad
 - La disponibilidad de datos puede ser usada maliciosamente

52

[Dificultades]

- No obstante, el paralelismo ofrece más **ventajas** que desventajas
 - Alto rendimiento, alto throughput
 - Alta disponibilidad – tolerancia a fallos
 - Economía y flexibilidad
 - Ofrece mejor compromiso entre rendimiento y costo que la computación secuencial
 - El costo de los cálculos o experimentos es menor a la vez que pueden resolverse diferentes problemas con el mismo sistema computacional
 - Expansión y escalabilidad
 - Adición de nuevas máquinas. En computación secuencial, es muy elevado el costo de expandir un computador, y físicamente presenta límites
 - Tecnología

53

[Contenido]

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- **Bibliografía**
- Glosario básico

54

Bibliografía

- **Almeida D., Giménez D., Mantas J., Vidal A.**, “*Introducción a la Programación Paralela*”. Paraninfo Cengage Learning. 2008.
- **Jordan H.F., Alagband G.**, “*Fundamentals of Parallel Processing*”. Pearson Prentice Hall. 2003.
- **Dongarra J., Foster I., Fox G., Gropp W., Kennedy K., Torczon L., White A.**, “*Sourcebook of parallel computing*”. Morgan Kaufmann Publishers. 2003.
- **Foster Ian** “*Designing and Building Parallel Programs*”, Addison Wesley. 1995 (también disponible on-line en: <http://www-unix.mcs.anl.gov/dbpp/>)
- **Fox Geoffrey C., Williams Roy D., Messina Paul C.** “*Parallel Computing Works!*”, Morgan Kaufmann Publishers. 1994 (también disponible on-line en: <http://www.netlib.org/utk/lsi/pcwLSI/text/>)
- **Morrison R. S.** – “*Cluster Computing –Architectures, Operating Systems, Parallel Processing and Programming Languages*” – GNU General Public License 2002.
[http://static.schoolrack.com/20736/2_Cluster_Computing -
_Architectures, Operating Systems, Parallel Processing & Programmi
ng Languages \(v2.4\).pdf](http://static.schoolrack.com/20736/2_Cluster_Computing_-_Architectures,_Operating_Systems,_Parallel_Processing_&_Programming_Languages_(v2.4).pdf)

Bibliografía adicional

- **Gramma A., Gupta A., Karypis G., Kumar V.**, “*Introduction to Parallel Computing*”. Pearson Addison Wesley. Second Edition. 2003.
- **Dongarra J., Foster I., Fox G., Gropp W., Kennedy K., Torczon L., White A.**, “*Sourcebook of parallel computing*”. Morgan Kaufmann Publishers. 2003.
- **Wilkinson B., Allen M.**, “*Parallel Programming - Techniques and Applications Using Networked Workstations and Parallel Computers*”. Pearson Prentice Hall. Second Edition. 2005.

[Bibliografía]

- [Moo65] **Moore G.**, “*Cramming more components onto integrated circuits*”. Electronics Magazine, Vol. 38, No. 8, April 19, 1965.
- [McD07] **McDaniel C.** “*Semiconductor breakthroughs promise smaller, faster chips*”. Automation Notebook Magazine, Vol. 9, 2007.

[Contenido]

- Introducción
- Un poco de Historia
- Motivación
- Objetivos
- Aplicaciones
- Dificultades
- Bibliografía
- Glosario básico

[Glosario Básico]



59

[Glosario Básico]



- Programa
 - Conjunto de sentencias que ejecuta una computadora para resolver un problema
 - Incluye la definición de datos y ejecución
- Proceso
 - Programa en ejecución
 - Tiene asociado un conjunto de estructuras (sector de memoria, variables, tiempo de cpu, etc.)
- Thread o hilo de ejecución
 - Parte concurrente de un proceso
 - Comparte el ambiente de ejecución
- Nodo
 - Una unidad de procesamiento.
 - Dependiendo de la arquitectura puede tratarse de una computadora o de un procesador.

60

Glosario Básico



- **Throughput**
 - Cantidad de resultados que se produce por unidad de tiempo
- **Ancho de Banda**
 - Cantidad de datos que pueden transferirse en una unidad de tiempo
- **Latencia**
 - Tiempo que tarda un dato en viajar del origen al destino
- **Concurrencia**
 - Habilidad de un algoritmo para ejecutar múltiples operaciones en un momento

61

Glosario Básico



- **Descomposición**
 - Proceso de dividir el trabajo en partes más pequeñas
- **Tarea**
 - Cada una de las porciones en que se descompone el trabajo
- **Granularidad**
 - Cantidad y tamaño de las tareas o porciones en que se divide un problema
- **Acoplamiento**
 - Grado de interdependencia entre los procesos
- **Comunicación**
 - Proceso por el cual dos o más procesos intercambian información
- **Mensaje**
 - Paquete de información que es enviado a un determinado proceso

62

Glosario Básico



- **Eficiencia**
 - Porcentaje de tiempo durante el cual fueron utilizados los procesadores (no ociosos)
- **Escalabilidad**
 - Habilidad de un programa para utilizar eficientemente los procesadores de un sistema paralelo a medida que el tamaño del problema crece
- **Secuencial**
 - Que realiza las tareas una a una hasta completar el tratamiento de todo el problema
- **Paralelo/Distribuido**
 - Que realiza tareas en paralelo (al mismo tiempo)



63

Resumen

- **Computación Paralela**
 - Definición, Necesidad, Utilidad
 - Sistemas paralelos y paralelo/distribuidos
- **Marco histórico y evolutivo**
- **¿Por qué Cómputo Paralelo?**
- **Propósito de la Computación Paralela**
- **Casos de aplicación de paralelismo**
- **Obstáculos vs. Ventajas**
- **Bibliografía inicial**
- **Glosario Básico**



Programación Paralela y Distribuida

**Licenciatura en
Ciencias de la Computación**
UNCuyo – Facultad de Ingeniería

