NOCIONES GENERALES SOBRE COMPUTACIÓN PARALELA

Máster de Modelización e Investigación Matemática, Estadística y Computación

Programación científica

Parte I

Curso 2014/2015

Contenido

Contenido

1.1 Introducción

- Un desafio importante en el mundo actual es el uso eficiente de los computadores para alcanzar objetivos que de otra forma no serían realizables.
- La respuesta a este desafio pasa, en muchas ocasiones, por el uso del paralelismo (computación de altas prestaciones, HPC).
- La tecnología multinúcleo, la utilización de redes, la acumulación de computadores interconectados, o la utilización de recursos geográficamente distribuidos (computación grid) son claros ejemplos de cómo la computación paralela va imponiéndose en la organización de los computadores.

1.1.1 ¿Por qué es necesaria más potencia computacional?

- Actualmente existen problemas cuya solución conocemos de forma matemática, pero cuando tratamos de obtenerla se genera un número de calculos elevado.
- La teoría de la complejidad estudia la forma de calcular este número de operaciones para un problema algorítmico dado.
- Actualmente se están planteando en muchos campos de la ciencia (mundo real), problemas “tratables” pero con características que los hacen de difícil solución con los computadores secuenciales disponibles hoy en día (*problemas de gran dimensión, problemas de tiempo real, problemas de gran desafío*).

- Los computadores paralelos se han convertido en una herramienta importante en muchos campos de la actividad humana, haciendo posible la resolución de grandes problemas antaño irresolubles.

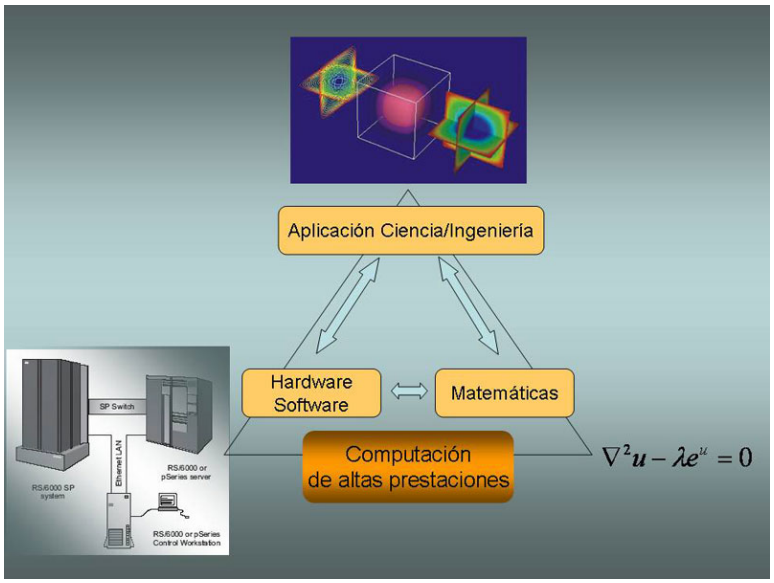

Figura 1: Esquema de actuación.

- Biociencia: Genoma humano, Investigación farmacos, Detección de enfermedades, Estudios de infecciones.
- Diseño asistido por ordenador: Automóviles, aviones, cohetes, estructuras, mecánica, etc.
- Ingeniería química: Diseño molecular.
- Contenidos digitales: Gráficos por computador, Digitalización de medios, etc.

Figura 2: Digitalización de medios.

- Diseño electrónico automatizado: Distribuir e interconectar los componentes de circuitos impresos.
- Geociencia: Petroleo y gas, Modelado de reservas, Modelos climáticos, Predicción meteorológica.
- Defensa.
- Energia: Gestión de la red eléctrica, etc.
- Transporte y logística: Planificación de rutas, Resolución de incidencias.
- Sector financiero: Análisis de riesgo, etc.
- Acádemico: Investigación básica y aplicada.

¡Computar eficientemente es siempre una ventaja!

- Fórmula 1:
  - Calcular el momento óptimo para el cambio de neumáticos.
  - Estado y temperatura del neumático.
  - Peso del coche.
  - Situación del tráfico: a la entrada, a la salida.
  - Túnel de viento.

Figura 3: Fórmula 1.

Contenido

1.2 Noción de computación paralela

- Un computador es un sistema digital complejo formado por un conjunto de circuitos electrónicos y elementos mecánicos que trabajan de forma conjunta para la consecución de un objetivo que, de forma general, se puede definir como el procesamiento de la información.
- Podemos considerar distintos niveles en una computadora que funcionan, se organizan y se estudian de forma independiente, pero que proporcionan de forma garantizada una serie de servicios de nivel superior.

- Aproximación por niveles:
  - Nivel físico: Microelectrónica, Nanoelectrónica, Transistores.
  - Nivel circuitos digitales: Circuitos digitales, Puertas lógicas, Transistores.
  - Nivel de arquitectura: Microprocesadores, Microchips.
  - Nivel software básico: Software de gestión de recursos, Sistema operativo.
  - Nivel software intermedio: Intérpretes, Compiladores, APIs (*Application Programming Interface*).
  - Nivel de aplicaciones: Aplicaciones específicas, Programas de usuario, Librerias.

- Los computadores secuenciales tradicionales están basados en el modelo introducido en la década de los cuarenta por John von Neumann, el cual consiste en una unidad central de proceso (CPU) y una memoria.
- Este modelo computacional emplea una secuencia de instrucciones y opera en una secuencia de datos, por lo que se denomina a este tipo de procesamiento computación secuencial. Los computadores de este tipo se denominan computadores SISD (Single Instruction Single Data).
- El modelo anterior puede considerarse como una buena forma de organizar el procesamiento (instrucciones/datos) y su generalización ha dado lugar a muchos de los lenguajes de alto nivel que actualmente utilizamos.

- La velocidad de un computador SISD está limitada por dos factores: la rapidez en la ejecución de instrucciones y la velocidad en la que la información es intercambiada entre memoria y CPU.
- Una forma de mejorar esa velocidad es mediante el uso de memorias rápidas de enlace (memorias cache), que actúan de intermediario entre la memoria principal, mucho más lenta, y la CPU, incrementando sustancialmente el rendimiento del sistema.
- Por otro lado, el ritmo de ejecución de instrucciones puede ser incrementado mediante la ejecución solapada de distintas etapas de varias instrucciones (pipelining).

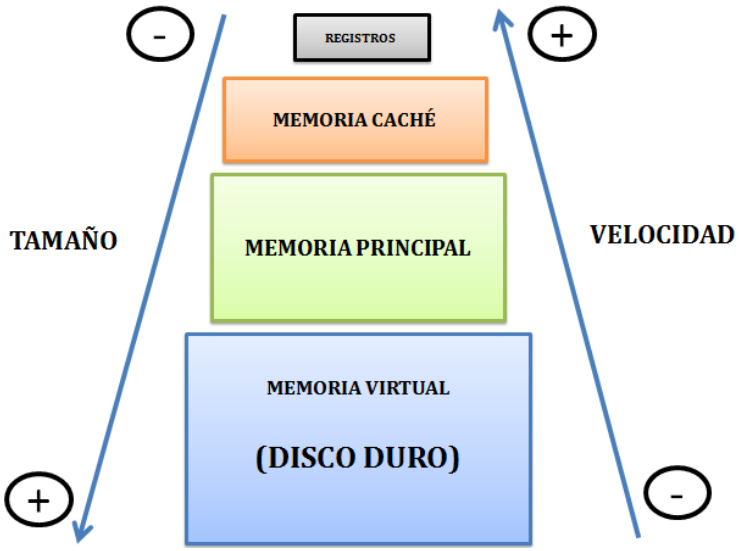

Figura 4: Jerarquía de memoria.

- Estas técnicas de mejora de la velocidad de un computador SISD son usadas comúnmente en los computadores SISD de alto rendimiento; no obstante tienen sus limitaciones.
- Existen otros modelos que nos permiten mantener un cierto grado de simultaneidad.
- Una forma alternativa para incrementar la tasa de ejecución de instrucciones es usar múltiples unidades de proceso y múltiples unidades de memoria interconectadas de algún modo (paralelismo).

- Denominaremos computación paralela a la computación que se lleva a cabo sobre computadores paralelos.
- Al poder realizar la ejecución de distintas instrucciones de forma simúltanea se deben aprovechar estas circunstancias para mejorar las prestaciones de los ordenadores.
- Esto requiere: rediseñar algoritmos, replantear las estructuras de datos, nuevos paradigmas de programación, etc.
- En definitiva, nos planteamos otro tipo de programación al que denominaremos computación paralela.

¿Necesito un supercomputador?

- Ranking de los 500 computadores más rápidos (**http://www.top500.org/lists/2013/11/**)
- Rendimiento evaluado en una sola aplicación.
  - High Performance Linpack (HPL).
  - Rendimiento en TFLPOS (10^{12} ops coma flotante/segundo).
- Sólo aparecen en la lista aquellos ordenadores que envian sus resultados. Los computadores privados, o confidenciales, no aparecen en la lista

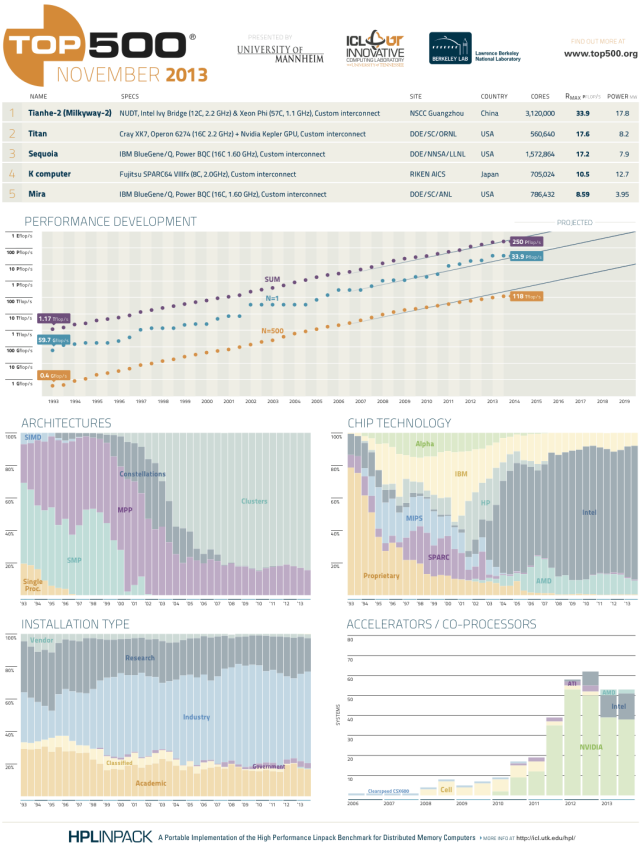

Figura 5: TOP 500.

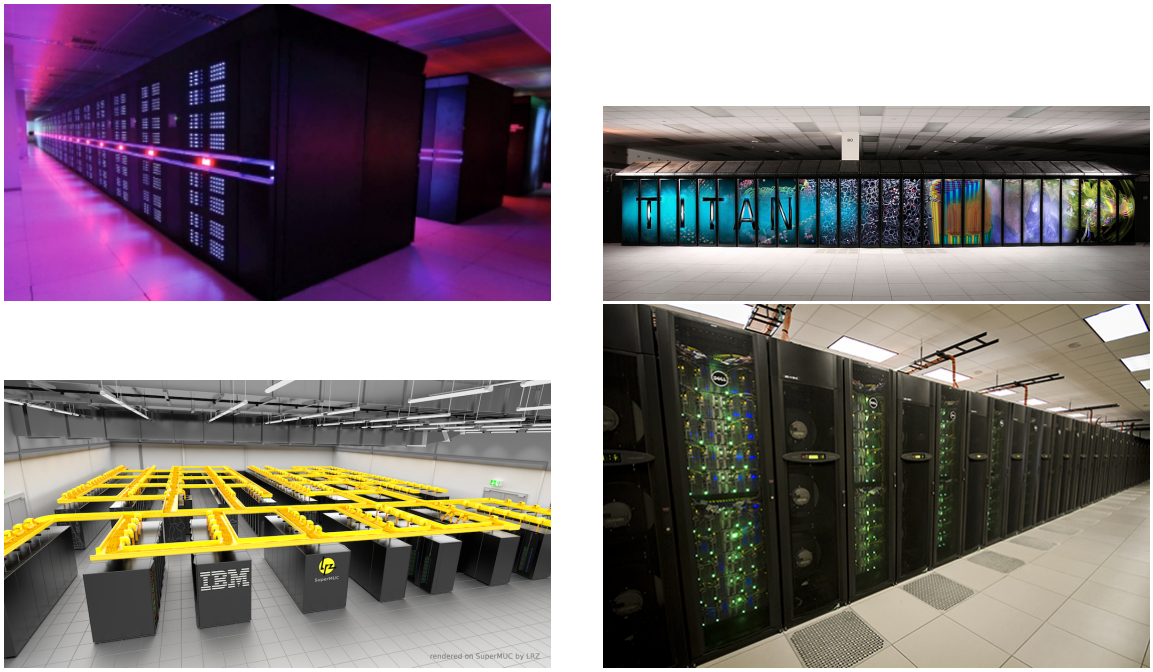

Figura 6: TOP 10.

¿Alternativas?

Figura 7: Alternativas.

Contenido

1.3 Multiplicidad del flujo de instrucciones y de datos

- En 1966 Michael J. Flynn propuso una forma de clasificación de las computadoras (taxonomía de Flynn). Esta organización se basa en la pluralidad de instrucciones y de datos que la computadora utiliza para procesar información.
- Es precisamente en esta división cuando aparecen por primera vez las máquinas paralelas divididas en tres clases:
  - SIMD (Single Instruction Multiple Data),
  - MISD (Multiple Instruction Single Data),
  - MIMD (Multiple Instruction Multiple Data).

- La clasificación de Flynn pone de manifiesto dos tipos de paralelismo, de *datos* y *funcional*.
- Paralelismo de datos: la misma función o instrucción se ejecuta repetidas veces con datos distintos y se da en los modelos SIMD y puede darse en los MIMD si la aplicación se implementa como un programa paralelo SPMD (Single Program Multiple Data).
- Paralelismo funcional: diferentes funciones o instrucciones se ejecutan en paralelo con distintos datos (modelos MIMD). Este paralelismo funcional puede darse a nivel de instrucciones, a nivel de bucle, a nivel de funciones o a nivel de programas.

1.3.1 Organización del Espacio de Direcciones

- La arquitectura del espacio de direcciones compartido proporciona un soporte hardware para los accesos de lectura y escritura de todos los procesadores a un espacio de direcciones compartido. Los procesadores se relacionan mediante la modificación de los datos almacenados en el espacio de direcciones compartido.

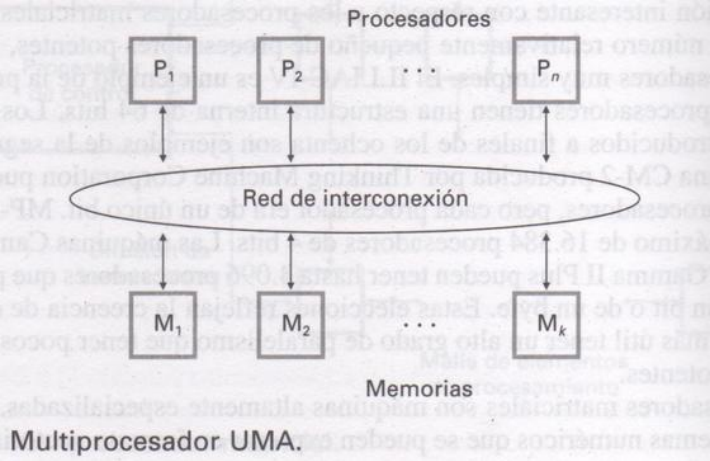

Figura 8: Ejemplo de espacio de direcciones compartido (UMA).

- Estos computadores se clasifican en dos categorías atendiendo a la uniformidad, o no, en el tiempo de acceso a todas las posiciones del espacio de memoria: UMA (Uniform Memory Access) y NUMA (NonUniform Memory Access).
- La mayoría de los computadores de memoria compartida tienen una memoria cache local en cada procesador para incrementar así el ancho de banda efectivo entre memoria y procesador. Los computadores de memoria compartida se denominan habitualmente multiprocesadores.
- Ejemplos de este tipo de computadores paralelos son NYU Ultracomputer y Finis Terrae del Centro de Supercomputación de Galicia (CESGA).

- En una arquitectura de memoria distribuida los procesadores están conectados usando una red de interconexión y no comparten recursos. Cada procesador tiene su propia memoria local, la cual es accesible sólo desde ese procesador, y ejecuta su propia copia del sistema paralelo.

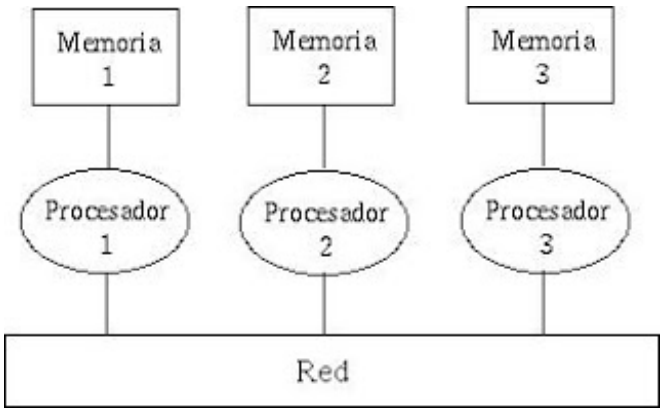

Figura 9: Típica arquitectura de paso de mensajes.

- La relación entre los distintos procesadores se realiza mediante el intercambio de información, siendo el paradigma de paso de mensaje la técnica empleada. Los computadores MIMD de memoria distribuida se denominan comúnmente multicomputadores.
- La programación es más compleja que en el caso de memoria compartida, ya que cuando los datos a usar por un procesador están en el espacio de direcciones de otro hace falta solicitarlas y transferirlas a través de mensajes.
- Ejemplos de computadores paralelos de memoria distribuida son Cosmic Cube, Paragon XP/S, iPSC, CM-E, nCUBE2, Mare Nostrum del Centro Nacional de Supercomputación (CNS), grupos de estaciones de trabajo (PCs).

Contenido

1.4 Redes de interconexión

- Los computadores de espacio de direcciones compartido y distribuido pueden ser construidos mediante la conexión de procesadores y de unidades de memoria de acuerdo a una variedad de redes de interconexión.
- Una red de interconexión es el conjunto de interruptores, líneas eléctricas de enlace, software y hardware de gestión que permite conectar varios elementos de proceso y/o módulos de memoria entre sí.
- Atendiendo a la posibilidad de cambiar las conexiones entre los nodos, podemos encontrar redes estáticas o dinámicas.

- Las redes estáticas están construidas mediante enlaces físicos existentes entre dos nodos y unicamente conectan los nodos prefijados. En este tipo de red los enlaces entre los procesadores son pasivos y no se puede reconfigurar la red con el fin de conectar otros procesadores. Se utiliza sobre todo en el caso de los multicomputadores.

- Las redes dinámicas están construidas fundamentalmente mediante elementos de comunicación activos (*switches*), siendo posible reconfigurar la topología de los enlaces entre los procesadores de forma dinámica. Se utilizan sobre todo en los multiprocesadores.

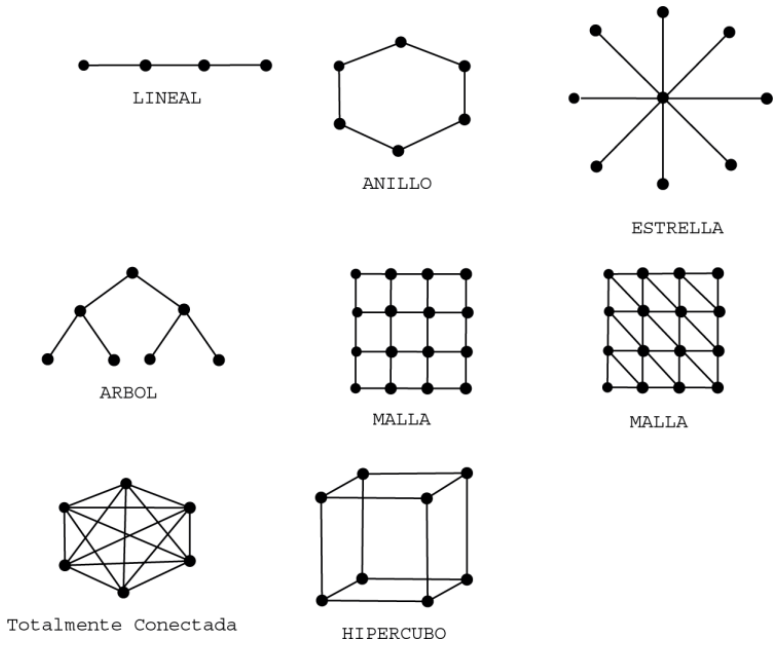

Figura 10: Redes de interconexión estáticas.

- Una de las mayores sobrecargas en la ejecución de programas paralelos proviene del intercambio de información entre los elementos de proceso. El coste de la comunicación es dependiente de una variedad de características como el modelo de programación, la topología de la red, el manejo de los datos, etc.
- El tiempo empleado en comunicar un mensaje entre dos procesadores en sistemas distribuidos es la suma del tiempo que se dedica a:
  - preparar un mensaje para su transmisión,
  - atravesar la red hasta su destino.

- Los principales parámetros que determinan el coste de la comunicación son los siguientes:
  - Tiempo de arranque (t_s), es el tiempo requerido para manejar un mensaje en los nodos emisor y receptor.
  - Tiempo de transferencia por palabra (t_w), si el ancho de banda de un canal es r palabras por segundo, entonces cada palabra emplea un tiempo $t_w = 1/r$ para atravesar esa unión.
  - Tiempo por nodo (t_h), una vez que un mensaje abandona un procesador, emplea una cantidad finita de tiempo en alcanzar el siguiente procesador en su camino.

- Uno de los factores que influye en los costes de comunicación en una red es la técnica de encaminamiento utilizada. Las dos técnicas de comunicación típicas para hacer llegar la información a su destino en los computadores paralelos se denominan almacenar-reenviar y segmentación.

- En el caso de almacenar-reenviar, cuando un mensaje atraviesa un camino con múltiples enlaces, cada procesador intermedio envía el mensaje al siguiente procesador después de recibir y almacenar el mensaje completo.
- Coste: e número de enlaces, m tamaño del mensaje

$$T_{ALMACENAR-REENVIAR} = t_s + e\ m\ t_w.$$

- La técnica de segmentación consiste en dividir el mensaje en unidades de tamaño fijo denominadas unidades de control de flujo o flits.
- La primera unidad es enviada desde el nodo origen al nodo destino para establecer de ese modo la comunicación.
- Una vez que ésta se haya establecido, el resto de unidades serán enviadas una tras otra.
- Un procesador intermedio no esperará por el mensaje completo; en cuanto le llegue un trozo del mensaje original, lo pasará al siguiente nodo.

- Por lo tanto, esta técnica emplea menos memoria y menos ancho de banda en los procesadores intermedios y es más rápida.

- Coste:

$$T_{SEGMENTACION} = t_s + e\ t_h + m\ t_w.$$

Las operaciones básicas de comunicación que pueden tener lugar y sus descripciones son las siguientes:

- Transferencia de un mensaje entre dos procesadores, es la operación más básica de comunicación y consiste en enviar un mensaje de un procesador a otro.

- Envío de uno a todos, un procesador envía idénticos datos a todos los procesadores o a un subconjunto de ellos.

- Envío de todos a todos, es una generalización del envío de uno a todos, en el que todos los procesadores inician simultáneamente la transmisión.

- Envío de uno a todos personalizado, en esta operación el procesador origen del envío transmite p mensajes, uno a cada uno de los p procesadores.

- Desplazamiento circular, consiste en que cada procesador envía un mensaje a un único procesador

Contenido

1.5 Métricas de rendimiento para sistemas paralelos

- La evaluación del rendimiento de un algoritmo paralelo no puede estar desligada de la arquitectura paralela sobre la cual se está ejecutando e incluso del sistema real que implementa dicha arquitectura.
- El tiempo de ejecución de un algoritmo paralelo depende no sólo del tamaño del problema sino también del número de elementos de proceso utilizados, de sus relativas velocidades de cálculo y de comunicación, de la topología de la red de interconexión, de la técnica de encaminamiento, etc.
- Un algoritmo que se ejecute de manera eficaz en un sistema, puede que no lo haga si una o varias de las características del sistema se ven modificadas.

- En esta sección, introduciremos algunas métricas que son comúnmente usadas para analizar el rendimiento.
- El tamaño del problema se define como el valor o conjunto de valores asociados a la entrada del problema que representa una medida de su tamaño respecto de otras entradas posibles.
- W denota la cantidad de operaciones básicas necesarias que hay que realizar sobre los datos de entrada para resolver un problema, siendo tal cantidad función del tamaño del problema.
- El tiempo de ejecución de un algoritmo secuencial, $T_{SECUENCIAL}$, es el tiempo transcurrido entre el comienzo y el fin de su ejecución en un único procesador.

- El tiempo de ejecución de un algoritmo paralelo en k procesadores, $T_{PARALELO}(W, k)$, es el tiempo empleado desde que el primer procesador inicia la ejecución de algoritmo hasta que el último procesador completa su ejecución.
- El tiempo paralelo se compone de una suma de, al menos, dos tiempos, que son:
  - Tiempo de cálculo, $T_{CALCULO}(W, k)$, lo definiremos como el tiempo de cómputo del procesador que más tarda en realizar sus cálculos,
  - Tiempo de comunicación, $T_{COMUNICACION}(W, k)$, que es el tiempo destinado a las transferencias de datos entre los procesadores.

Incremento de velocidad y Eficiencia

- Cuando se evalúa un sistema paralelo, es interesante determinar el aumento de rendimiento obtenido al realizar una implementación paralela de un algoritmo dado.
- Esta medida, denominada incremento de velocidad o Speedup, S, recoge el beneficio relativo de solucionar un problema en una plataforma paralela.
- Formalmente, S está definida como la relación entre el tiempo empleado para solucionar un problema en un único procesador y el tiempo requerido para solucionar el mismo problema en un multicomputador con k procesadores idénticos:

$$S(W,k) = \frac{T_{SECUENCIAL}(W)}{T_{PARALELO}(W,k)}.$$

- Debido a las distintas sobrecargas del procesamiento paralelo, los procesadores no pueden dedicar todo su tiempo a computación útil, computación que es también realizada por su homólogo secuencial.
- La fracción del tiempo empleada por un procesador en tareas útiles determina la eficiencia, E, de un sistema paralelo.
- La eficiencia se define como la proporción entre el incremento de la velocidad y el número de procesadores, así

$$E(W,k) = \frac{S(W,k)}{k} = \frac{T_{SECUENCIAL}(W)}{k\ T_{PARALELO}(W,k)}.$$

- En un sistema paralelo “ideal”, el incremento de velocidad es igual al número de procesadores y la eficiencia es igual a 1.
- En la práctica, S es menor que el número de procesadores y la eficiencia está entre 0 y 1, dependiendo del grado de efectividad con el que los procesadores han sido utilizados.
- En ocasiones, se puede observar un valor de S superior a k (eficiencia mayor que 1), efecto conocido como condición de superlinealidad.
- Esto es normalmente debido a un algoritmo secuencial no óptimo o a características hardware que sitúan al algoritmo secuencial en desventaja.

- La *Ley de Amdahl* (1967) establece que “una pequeña parte serie que no se puede realizar en paralelo en un proceso puede provocar una fuerte disminución del rendimiento del sistema”.
- Esta Ley pone un límite superior al incremento de velocidad, y por tanto también a la eficiencia, de un sistema paralelo atendiendo al hecho de que los procesos suelen tener partes que no pueden ser ejecutadas en paralelo, sino de forma secuencial pura.
- Supongamos que P es la proporción de un programa que puede ser paralelizada utilizando k procesadores, entonces, el máximo incremento de velocidad que puede alcanzarse es

$$S(W,k) = \frac{1}{(1-P) + P/k}.$$

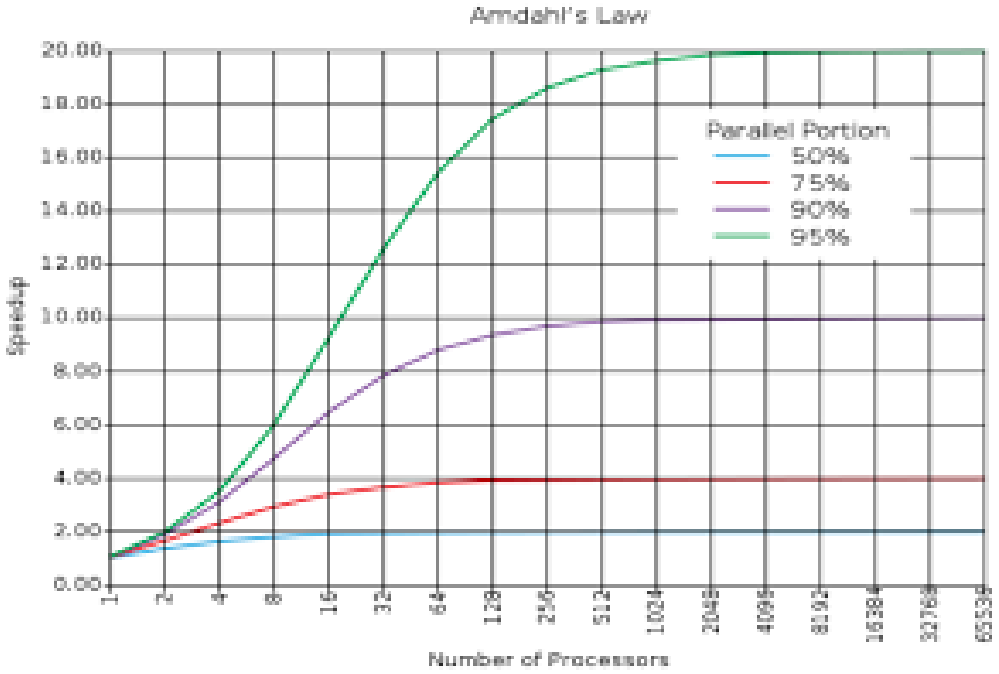

Figura 11: Ejemplo Ley de Amdahl.

Escalabilidad

- Se define escalabilidad de un algoritmo paralelo como su capacidad para hacer un uso efectivo de los recursos del sistema con un número mayor de procesadores dentro de una arquitectura dada.
- Suelen plantearse dos modelos genéricos de escalado del sistema:
  - Modelo de *escalado por tiempo crítico*, donde el objetivo es disminuir el tiempo de ejecución del algoritmo al aumentar el número de procesadores, manteniendo el tamaño del problema.
  - Modelo de *escalado por precisión crítica*, el objetivo en este caso es poder aumentar el tamaño del problema, manteniendo constante la eficiencia, el tiempo de ejecución, la velocidad o la memoria.

Contenido

1.6 Caso de estudio

- Utilizaremos como ejemplo inicial la suma de n números almacenados en un vector que puede hacerse, en secuencial, tal como se describe en el siguiente algoritmo:

Algorithm 1 Suma secuencial

Entrada: $a[n]$
Salida: s suma de los valores de a

1: $s = a[0]$
2: **for** $i = 1 : n - 1$ **do**
3: $\quad s = s + a[i]$
4: **end for**

- Si analizamos el número de instrucciones que se ejecutan, $n-1$ iteraciones, cada iteración dos instrucciones, el coste es $2n-1$. Si analizamos sólo el número de sumas realizadas el coste es $n-1$.
- Consideraremos a continuación que podemos disponer de $n/2$ procesadores. Supongamos que los datos están disponibles para todos los procesadores, memoria compartida, el estudio se puede extender al caso de memoria distribuida con paso de mensajes.
- Inicialmente los $n/2$ procesadores suman cada uno de ellos un par de números, en el siguiente paso la mitad de los procesadores ($n/4$) suman un par de números, en el siguiente intervienen la cuarta parte ($n/8$) y así sucesivamente, hasta que al final uno de los procesadores obtiene la suma total.

- El problema se simplifica si consideramos n potencia de 2, en cuyo caso se realizan $\log(n)$ operaciones en coma flotante.
- Estamos lejos del valor mínimo que podríamos intentar conseguir debido, fundamentalmente, a una distribución no adecuada del trabajo entre todos los procesadores en todos los pasos de computación (*desequilibrio en la carga*).
- Existe, además, una limitación intrínseca (ley de Amdahl) que hace que el algoritmo no sea totalmente paralelizable.

Contenido

1.7 Bibliografía

- F. Almeida, D. Giménez, J.M. Mantas, A.M. Vidal, *Introducción a la programación paralela*, Paraninfo Cengage Learning, 2008.
- R. Cortina, *El Método de Neville: un enfoque basado en computación de altas prestaciones*, Tesis Doctoral, 2008.
- G.H. Golub, C.F. Van Loan, *Matrix computations* (3rd ed.), Johns Hopkins University Press, 1996.
- A. Grama, G. Karypis, V. Kumar, A. Gupta, *Introduction to Parallel Computing* (2nd ed.), Addison-Wesley, 2003.