

# Interconnection networks & Ley de Amdahl y Gustafson

Luis Felipe Sante Taipe

Computación Paralela y Distribuida  
Grupo A

Profesor: Alvaro Henry Mamani Aliaga

9 de febrero de 2025

## 1. Introducción

La interconexión juega un papel decisivo en el rendimiento de los sistemas de memoria compartida y distribuida: incluso si los procesadores y la memoria tienen un rendimiento prácticamente ilimitado, una interconexión lenta degradará seriamente el rendimiento general de todos los programas paralelos, excepto los más simples [1].

## 2. Interconnection networks

### 2.1. Interconexiones de memoria compartida

Las dos interconexiones más usadas son:

#### ✓ Bus

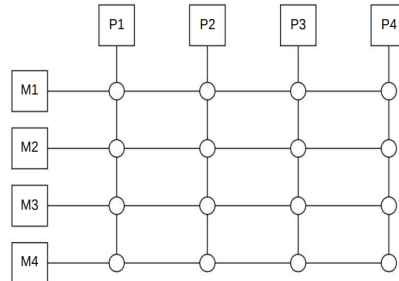
- Colección de cables de comunicación paralelos.
- Dichos cables son compartidos por dispositivos que están conectados a él.
- Mientras más dispositivos conectados es probable que una mayor disputa por el uso del bus y el rendimiento disminuye.
- Una consecuencia es que si tenemos varios procesadores estos tendrán que esperar para poder acceder a la memoria principal.
- Si el tamaño de los sistemas de memoria compartida aumenta, los buses se reemplazan rápidamente por **switched interconnects** que sirven para controlar el enrutamiento de datos entre dispositivos conectados.
- Bajo costo.

#### ✓ Crossbar

- Las líneas son enlaces de comunicación bidireccional.
- Los círculos son **switches**.

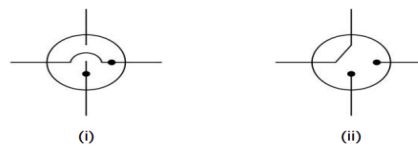
- Los cuadrados son núcleos

**Figura 1.** Crossbar



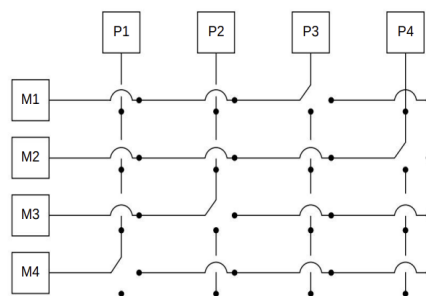
Configuración de **switches** internos en un crossbar.

**Figura 2.** Configuración



Por ejemplo en [Figura 3](#) notamos que P1 tiene un enlace con la memoria 4, y es evidente ya que P1 al final hay una desviación para poder conectarse con M4, así funciona para todos los procesadores

**Figura 3.** Accesos simultáneos a la memoria por parte de los procesadores



## 2.2. Interconexiones de memoria distribuida

- ✓ Interconexión directa

Ring	Toroidal Mesh
Permite varias múltiples comunicaciones simultáneas	Los switches son más complejos
Si hay $p$ procesadores el número de enlaces es de $2p$	Si hay $p$ procesadores el número de enlaces es de $3p$

Una medida de "número de comunicaciones simultáneas" es **bisection width**, por ejemplo.

Tendremos dos casos, supongamos que tenemos 8 nodos conectados en forma de anillo, en el primer caso dividimos esto en 2 mitades, por lo que solo existirá dos formas de comunicación entre mitades, en el segundo caso dividimos los nodos son un etiquetado intercalado para que estos puedan tener 4 comunicaciones.

### ¿Qué es el bisection width?

Se supone que el bisection width da una estimación del "peor de los casos". Una forma alternativa de calcular es eliminar el número mínimo de enlaces necesarios para dividir el conjunto de nodos en dos mitades iguales. Si tenemos un toroidal mesh cuadrada de 2D con  $p = q^2$  nodos, podemos dividir los nodos en dos mitades eliminando los enlaces horizontales centralesz los enlaces horizontales "envolventes". Esto sugiere que bisection width es como máximo  $2q = 2\sqrt{2}$ . De hecho, este es el menor número posible de enlaces y la anchura de bisección de una malla toroidal bidimensional cuadrada es de  $2\sqrt{2}$ .

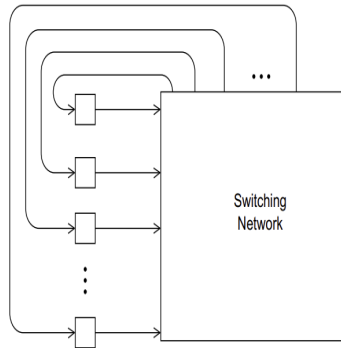
El hipercubo es una interconexión directa altamente conectada que se ha utilizado en sistemas reales. Los hipercubos se construyen de forma inductiva: un hipercubo unidimensional es un sistema totalmente conectado con dos procesadores. Un hipercubo bidimensional se construye a partir de dos hipercubos unidimensionales uniendo los switches correspondientes". De manera similar, un hipercubo tridimensional se construye a partir de dos hipercubos bidimensionales.

Cabe resaltar que un hipercubo tiene  $p = 2^d$  nodos y su bisection es su mitad, además tiene más conectividad que un anillo o toroidal mesh, pero los switches deben ser más poderosos ya que soportará  $1 + d = 1 + \log(p)$  hilos.

### ✓ Interconexión indirecta

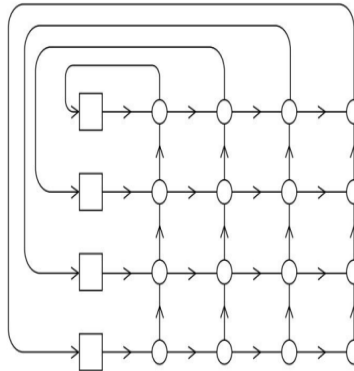
Se muestran con enlaces unidireccionales y una colección de procesadores, cada uno de los cuales ya tiene asignado un enlace de salida, entrada y una red de switches

**Figura 4.** Interconexión indirecta



El **crossbar** y la **red omega** son ejemplos relativamente simples de redes indirectas. Un ejemplo de interconexión indirecta es un diagrama de barra transversal de memoria distribuida, además que mientras dos procesadores no intenten comunicarse con el mismo procesador, todos los procesadores pueden comunicarse simultáneamente con otro procesador.

**Figura 5.** Una interconexión de barra cruzada para memoria distribuida



Bisection width es más complicado para las redes indirectas pero el principio de división sigue siendo el mismo, El bisection width de un crossbar  $p * p$  es  $p$  y el bisection width de una red omega es  $p/2$ .

### 2.3. Latencia y ancho de banda

Siempre que se transmiten datos, nos interesa saber cuánto tardarán en llegar a su destino. Hay dos cifras que se utilizan a menudo para describir el rendimiento de una interconexión: la latencia y el ancho de banda. La **latencia** es el tiempo que transcurre entre que el origen comienza a transmitir los datos y el destino comienza a recibir el primer byte y el **ancho de banda** es la velocidad a la que el destino recibe datos después de haber comenzado a recibir el primer byte.

### 3. Ley de Amdahl y Gustafson

#### 3.1. Ley de Amdahl

Utilizada para averiguar la mejora máxima de un sistema de información cuando solo una parte de éste es mejorado [2].

La Ley de Amdahl se basa en una carga de trabajo o tamaño de entrada prefijados. Esto implica que la parte secuencial de un programa no cambia con respecto al número de procesadores de la máquina, sin embargo, la parte paralelizable es uniformemente distribuida en el número de procesadores [3].

##### 3.1.1. Terminología

- ✓  $F_m$  = Fracción de tiempo que el sistema utiliza el subsistema mejorado [2].
- ✓  $A_m$  = Factor de mejora que se ha introducido en el subsistema mejorado [2].
- ✓  $T_a$  = Tiempo de ejecución antiguo [2].
- ✓  $T_m$  = Tiempo de ejecución mejorado [2].
- ✓  $A$  = Es la aceleración o ganancia en velocidad conseguida en el sistema completo debido a la mejora de uno de sus subsistemas [2].
- ✓  $A_m$  = Es el factor de mejora que se ha introducido en el subsistema mejorado.

##### 3.1.2. Fórmula original

$$T_m = T_a \cdot \left( (1 - F_m) + \frac{F_m}{A_m} \right) \quad (1)$$

Se puede reescribir usando la definición del incremento de la velocidad.

$$\frac{T_a}{T_m} = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}}$$
$$A = \frac{1}{(1 - F_m) + \frac{F_m}{A_m}} \quad (2)$$

##### 3.1.3. Ejemplo

Suponer una mejora 10 veces más rápida que la máquina original, pero sólo se utiliza el 40 % del tiempo. ¿Cuál es la aceleración que genera dicha mejora? [4]

$$\checkmark \quad \frac{f_m=40}{100=0,4}$$

$$\checkmark \quad A = 10$$

Usando la fórmula 2

$$A = \frac{1}{(1 - 0,4) + \frac{0,4}{10}}$$

$$A = 1,56$$

### 3.2. Ley de Gustafson

Es una ley en ciencia de la computación que establece que cualquier problema suficientemente grande puede ser eficientemente paralelizado. La ley de Gustafson está muy ligada a la ley de Amdahl, que pone límite a la mejora que se puede obtener gracias a la paralelización, dado un conjunto de datos de tamaño fijo, ofreciendo así una visión pesimista del procesamiento paralelo. Por el contrario la ley de Gustafson ofrece un nuevo punto de vista y así una visión positiva de las ventajas del procesamiento paralelo [3].

#### 3.2.1. Terminología

- ✓  $P$  = es el número de procesadores.
- ✓  $S$  = es el speedup.
- ✓  $\alpha$  = la parte no paralelizable del proceso.

#### 3.2.2. Fórmula

$$S(P) = P - \alpha \cdot (P - 1) \tag{3}$$

### 3.3. Aplicaciones

La ley de Amdahl y la ley de Gustafson tienen aplicaciones en la investigación tanto como en los sistemas computacionales [3] .

Ley de Amdahl	Ley de Gustafson
Presupone que los requerimientos de cómputo se mantendrán invariables, dado el incremento del poder de procesamiento	Argumenta que más poder de cómputo causará que los datos sean analizados más profunda y cuidadosamente: pixel por pixel o unidad por unidad
Si una computadora demora en estar lista un minuto, aumentado el poder cómputo se supone que mejora el tiempo de arranque, pero mayor paralelización fallaría eventualmente en hacer el inicio más rápido, si alguna parte del proceso de inicio fuera esencialmente secuencial	Argumenta que un aumentando el poder de cómputo conllevaría a un incremento similar en las capacidades del sistema.

### 3.4. Limitaciones

#### ✓ Ley de Amdahl

- La principal limitación en la aplicación de esta ley es que cuando se introducen varias mejoras a la vez no todas ellas pueden ser calculadas a la vez, en este caso los cálculos se harían por separado [5].
- No siempre la mejora aplicada al sistema mejorará el rendimiento global del mismo en igual proporción [5].

#### ✓ Ley de Gustafson

- Los algoritmos no lineales dificultan la utilización de las ventajas del paralelismo expuestas por la ley de Gustafson [3].

## 4. Conclusiones

Es muy importante saber en qué situación se encuentra nuestro problema (dependiente o independiente) para poder elegir el mejor método de paralelización ya que nos permitirá usar la mejor comunicación entre procesos, la mejor sincronización además de repartir trabajo equitativo para todos los procesos, evitando fusiones vacías.

## 5. Referencias

1. Pacheco, P., *An Introduction to Parallel Programming*, 1.<sup>a</sup> ed.; Morgan Kaufmann: 2011.

2. Wikipedia Ley de Amdahl - Wikipedia, [Internet; descargado 26-septiembre-2022], 2020.
3. Wikipedia Ley de Gustafson - Wikipedia, [Internet; descargado 26-septiembre-2022], 2022.
4. Sánchez, L. M. Ejercicios de Ley de Amdahl, [Internet; descargado 26-septiembre-2022], 2022.
5. Cristian Brochero, e. a. Ley de Amdahl, [Internet; descargado 26-septiembre-2022], 2022.