Arquitectura e Ingeniería de Computadores

Modulo III. Multiprocesadores

Tema 7. Introducción a los computadores paralelos

Lección 2-3

UCM

Departamento de Arquitectura de Computadores y Automática

Arquitecturas Paralelas (1)

- **Computador Paralelo**
  - Colección de elementos de procesamiento que cooperan y se comunican para resolver problemas grandes (cálculo, almacenamiento) "rápidamente". (Almansi and Gottlieb 1989)
- **Objetivos**
  - Rendimiento / Eficiencia
  - Tolerancia a fallos
- **Esta definición da lugar a muchas preguntas:**
  - ¿Cuántos procesadores?
  - ¿Cómo de potente cada procesador?
  - ¿El número puede crecer de una manera sencilla: es escalable el sistema?
  - ¿Cuánta memoria?
  - ¿Cómo se comunican y cooperan dichos elementos?
    - ¿Cómo se transmiten los datos entre los procesadores?
    - ¿Qué tipo de interconexión conecta los distintos procesadores?
  - ¿Qué abstracción (primitivas) proporciona el hardware /software al programador?

Arquitecturas Paralelas (2)

- **Arquitecturas Paralelas Ayer**
  - Históricamente, la computación paralela ha consistido en una serie de modelos rivales y arquitecturas divergentes, sin una línea de desarrollo predecible.
  - La incertidumbre arquitectural ha paralizado el desarrollo del software paralelo

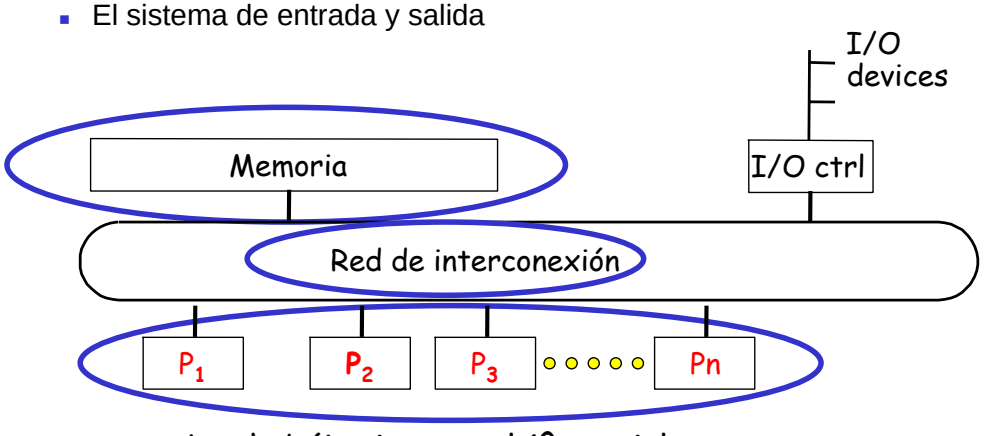
- **Arquitecturas Paralelas Hoy**
  - Extensión de la noción clásica de la arquitectura de computadores, para soportar comunicación y cooperación.

Arquitectura Convencional + Arquitectura de Comunicaciones

- Repertorio de Comunicaciones (Abstracción):
  - Operaciones de Comunicación y Sincronización Básicas disponibles a nivel usuario.
- Microarquitectura/Organización: Implementa la Abstracción

Arquitecturas Paralelas (3)

- **Estudio de la arquitectura de un sistema con multiprocesadores**
  - Partes a estudiar
    - El nodo de computo (procesador)
    - El sistema de memoria
    - El sistema del sistema de red de comunicación
    - El sistema de entrada y salida



Los habéis visto en el 1º parcial:

- Estas facetas están relacionadas y solapadas entre sí
  - El grado de solapamiento depende de la implementación concreta.
- Presentan varias alternativas de diseño que
  - proporcionan distintas prestaciones.
    - procesadores escalares con pipeline din + espec,
    - superescalares,
    - multithread,
    - simultáneos multithread

Arquitecturas Paralelas (4)

- Forma clásica de organizar las computadoras:

Taxonomía de Flynn (1966)

- Se basa en el número de instrucciones y de la secuencia de datos que la computadora utiliza para procesar información

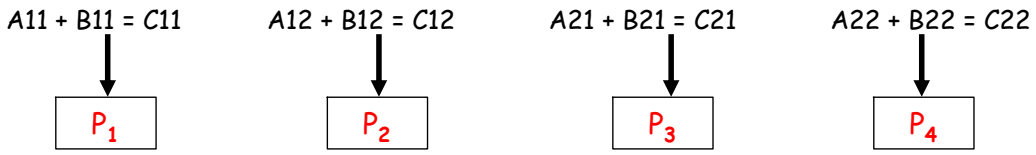
- Clasificación
  - **SISD** (Single Instruction Single Data)
    - El modelo tradicional de computación secuencial. Uniprocesadores
  - **SIMD** (Single Instruction Multiple Data)
  - **MIMD** (Multiple Instruction Multiple Data)

Arquitecturas paralelas

Arquitecturas Paralelas (5)

- **SIMD** (Single Instruction Multiple Data)
  - Todos lo procesadores ejecutan la misma instrucción, pero cada uno con diferente dato.

Ejemplo: Sumando dos matrices A + B = C.
Siendo A y B matrices de 2x2 y teniendo 4 procesadores:



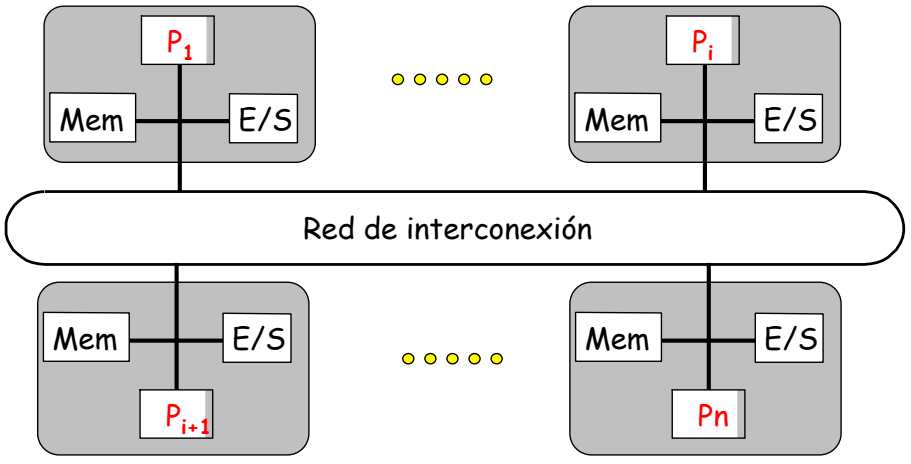
- Computadoras vectoriales
- Rendimiento adecuado en programas numéricos
- Paralelización automática viable

Arquitecturas Paralelas (6)

- **MIMD** (Multiple Instruction Multiple Data)
  - Cada procesador puede ejecutar su propia secuencia de instrucciones y tener sus propios datos.
  - Los sistemas MIMD se clasifican en:
    - Sistemas de **Memoria Compartida.**
    - Sistemas de **Memoria Distribuida**.
  - Más flexible que SIMD y de mayor interés para nosotros
  - Procesamiento de propósito general
  - Paralelización automática complicada

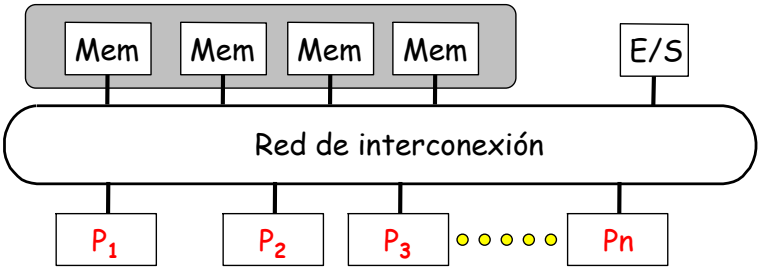
Arquitecturas Paralelas (7)

- **Clasificación de las arquitecturas paralelas MIMD**
  - Sistemas con **memoria distribuida** (DM) o **multicomputadores**
    - Cada procesador tiene su propio espacio de direcciones
      - Si un procesador requiere los datos contenidos en la memoria de otro procesador, deberá enviar un mensaje solicitándolos
      - El acceso a memoria remota puede ser lento
    - El programador necesita conocer donde están almacenados los datos



Arquitecturas Paralelas (8)

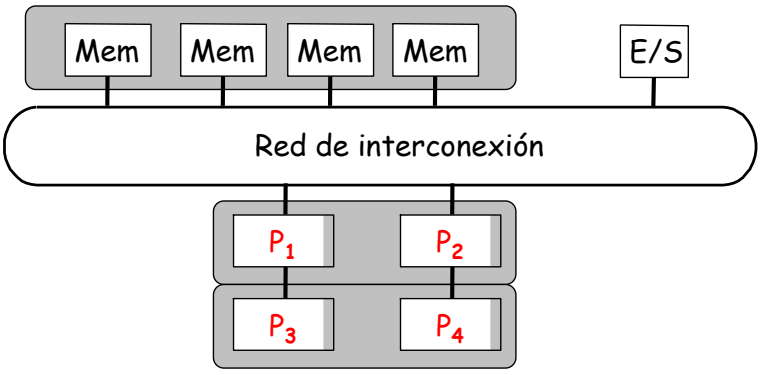
- **Clasificación de las arquitecturas paralelas MIMD**
  - Sistemas con **memoria compartida** (SM) o **multiprocesadores**
    - Todos los procesadores comparten el mismo espacio de direcciones
      - Cada procesador tiene acceso a toda la memoria
      - Tiempos de acceso a memoria uniformes
      - Lecturas y escrituras de todos los procesadores tienen exactamente las mismas latencias;
      - El acceso a memoria es por medio de un canal común.
    - El programador no necesita conocer donde están almacenados los datos



UCM

Arquitecturas Paralelas (9)

- **Clasificación de las arquitecturas paralelas MIMD**
  - **Multicores**
    - Sistemas con **memoria compartida**
    - Los distintos procesadores están en el mismo chip
      - Comunicación entre los procesadores muchos más rápida, no es necesario usar la red de interconexión



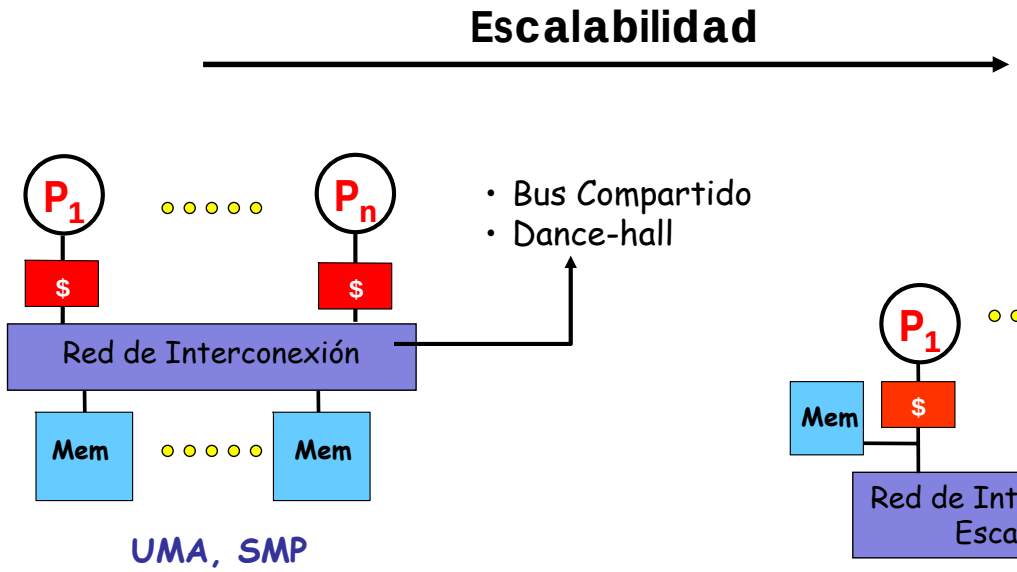
UCM

Arquitecturas Paralelas (7)

- Diferencias entre ambas arquitecturas
  - los multiprocesadores (**memoria compartida**)
    - Mayor **latencia**
    - Menor **escalabilidad**
    - El mecanismo de **comunicación** entre procesadores sencillo
    - **Multiprocesadores (memoria compartida)**
      - Proporcionan soporte en Hw para incrementar las prestaciones del SW de sincronización
    - **Programación** (herramientas, programador) sencillas
      - No es necesario distribuir la carga de trabajo
  - los multicomputadores (**memoria distribuida**)
    - Menor **latencia**
    - Mayor **escalabilidad**
    - El mecanismo de **comunicación** entre procesadores complejo
      - Implementar Mecanismos explícitos en SW para poder comunicarse
    - **Sincronización**
      - Se aprovechan los mecanismos de comunicación
    - **Programación** (herramientas, programador) complejas
      - Ubicar en la memoria local de cada procesador el código que se va a ejecutar y los datos que este código utiliza.

UCM

Arquitecturas Memoria Compartida (1)



Dance-hall

- Red Escalable (multietapa...)
- Es una arquitectura simétrica y escalable, pero la memoria está igual de lejos de todos los procesadores

•

•El controlador de memoria determina:

- Acceso local
- Transacción Red (mensaje con controlador de memoria remoto)

UCM

Programación paralela (1)

- **Modelo de Programación**: Nuevo concepto
  - ¿Cómo se **comparte** la información?
  - ¿Cómo transferir la información entre distintas partes de un programa que se ejecuta en paralelo?
  - ¿Cómo se lleva a cabo la **coordinación** de actividades?
  - ¿Qué primitivas de sincronización hay para coordinar actividades?
- Paralelizar una aplicación:
  - Partir de un código secuencial
    - Uso del **compilador,** en el recae la responsabilidad
    - La paralelización depende del código
  - Partir de la definición del problema de una manera paralela,
    - Uso de **librerías** paralelas
    - Más esfuerzo para el programador

Programación paralela (2)

- **Modos de programación paralela**
  - Paralelismo de datos SPMD (Single-Program Multiple-Data)
    - Todos los códigos que se ejecutan en paralelo se obtiene compilando el mismo programa
    - Cada copia trabaja con datos distintos y se ejecuta en un procesador diferente
  - Paralelismo de tareas SPMD (Multiple-Program Multiple-Data)
    - La aplicación a compilar **se divide en unidades independientes**
    - Todos los códigos que se ejecutan en paralelo se obtiene compilando las distintas unidades independientemente
    - Cada unidad trabaja con un conjunto de datos y se asigna a un procesador diferente.

Programación paralela (3)

- **Arquitecturas Paralelas (Ayer) :**
  - **Las arquitecturas paralelas se diferencian en el estilo de programación que más favorece su implementación HW.**
  - Se integra en el concepto de arquitectura
    - Modelo de Programación
    - Arquitectura de Comunicaciones
  - Clasificación

    Estilo de programación / Arquitectura paralela
    - Variables Compartidas / Memoria compartida
    - Paso de Mensajes / Memoria distribuida
    - Paralelismo de Datos / Procesadores matriciales
    - …

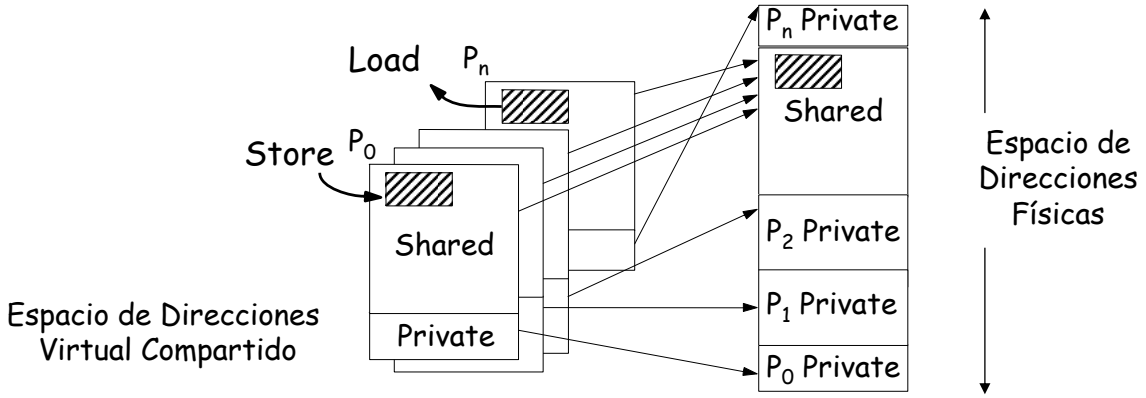
UCM

Arquitecturas Memoria Compartida (1)

Estilo de programación / Arquitectura paralela

Variables Compartidas / Memoria compartida

- Características
  - Se aplica en procesadores con memoria compartida ya que los diferentes procesos comparten variables.
  - **Comunicaciones entre procesos implícitas**: operaciones LOAD/STORE
  - **Sincronizaciones entre procesos**: primitivas que ofrece el SW:
    - (cerrojos, semáforos…)



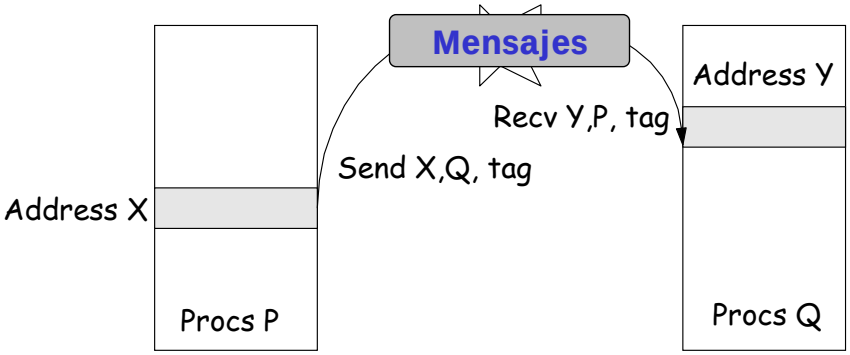
UCM

Arquitecturas Paso de Mensajes (1)

Estilo de programación / Arquitectura paralela

Paso de Mensaje / Memoria distribuida

- Características
  - Cada procesador tiene su espacio de direcciones propio (memoria local)
  - **Comunicación explícita** mediante operaciones **send/recv**
    - Sobrecarga: Construir la cabecera del mensaje; copiar los datos en el buffer de la red; enviar el mensaje; copiar los datos en el buffer receptor.
  - Las comunicaciones están integradas en el **sistema de E/S** en lugar de en el sistema de memoria
  - Los mensajes se aprovechan para sincronizar procesos



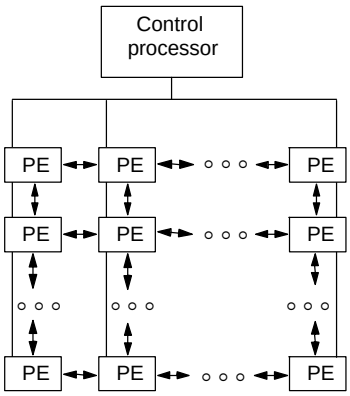
UCM

Arquitecturas con Paralelismo de Datos (1)

Estilo de programación / Arquitectura paralela

Paralelismo de datos / Procesadores vectoriales

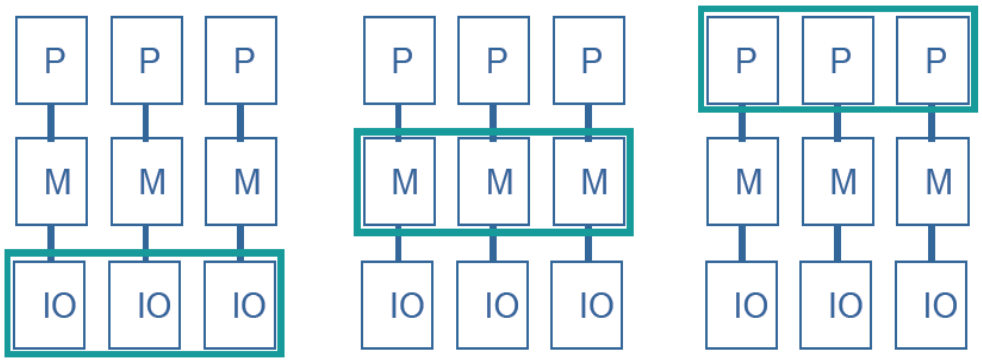
- Características
  - El programador escribe un programa con construcciones que permiten aprovechar el paralelismo de datos
    - Blucles
    - Operandos no escalares: vectores, matrices,…..
  - **Sincronización implícita** la genera el **compilador**



UCM

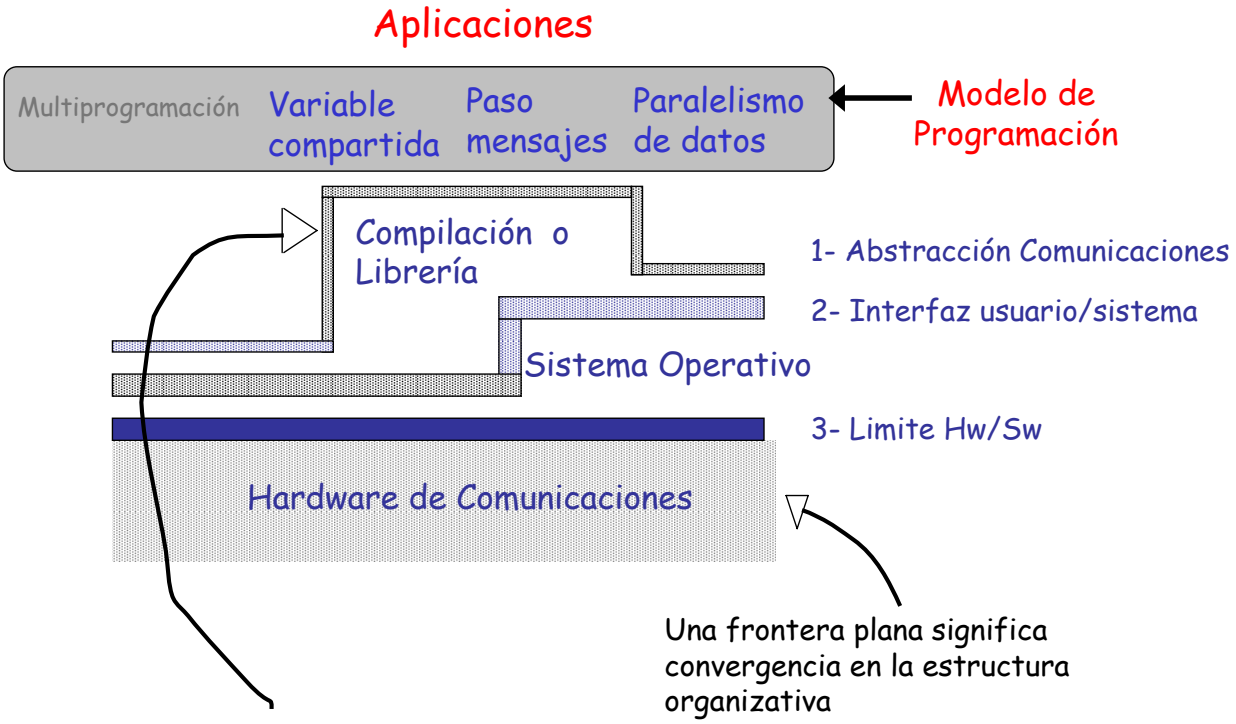
Resumen

- Clasificación histórica



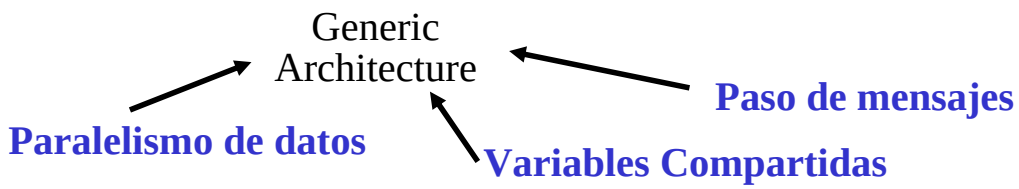
Union:	I/O (Red)	Memoria	Procesador
Modelo:	Paso mensajes	Memoria Compartida	Paralelismo de Datos

Resumen (2)



Convergencia: Arquitectura Paralela Genérica

- Clasificación tiene menos sentido en la actualidad :
  - SW: Los distintos modelos de programación se soportan en las diferentes Arq
    - Paso de mensajes se soporta fácilmente en las máquinas con memoria compartida (buffer compartido, locks y Flags)
    - En las máquinas de paso de mensajes, se puede establecer un espacio de direcciones virtual compartido
  - HW: **convergencia** entre las estructuras hardware utilizadas en el diseño de:
    - Las arquitecturas de paso de mensajes
    - Las arquitecturas de memoria compartida escalables



Aspectos de diseño en arquitecturas paralelas

- **Identificación** (*Naming*):
  - ¿Cómo se comunican los datos y/o el destino?
- **Operaciones**:
  - ¿Qué operaciones se puede realizar sobre dichos datos?
- **Sincronización** (*Ordering*):
  - ¿Cómo se coordinan los productores y consumidores de los datos?
- **Rendimiento**:
  - Latencia: ¿Cuánto se tarda en establecer la comunicación?
  - Ancho de Banda: ¿Cuántos datos pueden comunicarse por segundo? ¿Cuántas operaciones por segundo?

Rendimiento: Latencia

- **La latencia es problemática cuando los procesadores son rápidos**
  - Afecta al rendimiento.
    - El procesador tiene que esperar
  - Afecta a la facilidad de programación.
    - Requiere pensar en las comunicaciones para limitar la penalización
- **Soluciones:**
  - Reducir la frecuencia de los eventos de latencia elevada
    - Cambios algorítmicos, distribución de cómputos y datos, conmutar a otras tareas
  - Reducir la latencia de las comunicaciones
    - Almacenar en cache los datos compartidos
    - Diseño del interfaz de la red

Rendimiento: Ancho de banda

- Necesidades locales y globales
- **Ancho de banda local/privado**:
  - Distribuir la memoria entre los elementos de proceso
  - Cambios en las aplicaciones, uso de caches locales, diseño del sistema de memoria
- **Ancho de banda global**
  - Red de interconexión escalable
  - Memoria y caches distribuidas
  - Interfaces de red eficientes
  - Evitar contención

Coste de comunicaciones

Cost = Freq x (Overhead + Lat + Length/BW - Overlap)

- Dónde
  - Frequency = nº de comunicaciones por unidad de trabajo
    - Algoritmo, emplazamiento, replicación
  - Overhead = ciclos empleados en el comienzo/gestión
    - Copias, eventos
  - Latency = tiempo empleado en mover los bits de fuente a destino
    - Asistente comunicaciones, topología, encaminamiento, congestión
  - Transfer time = Lat + Length/BW
    - Asistente comunicaciones, enlaces, congestión
  - Overlap = porción que se solapa con trabajo útil
    - Asistente comunicaciones, operaciones comunicación, diseño procesador

Prestaciones en arquitecturas paralelas (1)

- Medidas que sirven para evaluar las prestaciones
  - **Tiempo de ejecución**:
    - Tiempo de ejecución de una aplicación en el sistema
  - **Productividad:** (throughput)
    - Número de aplicaciones que el sistema es capaz de procesar por unidad de tiempo
  - **Funcionalidad:**
    - Cargas de trabajo para las que está orientado el diseño de la arquitectura
  - **Tolerancia a fallos:**
    - Capacidad de un sistema de mantenerse en funcionamiento ante un fallo
  - **Expansibilidad:**
    - Posibilidad de expandir el sistema modularmente
  - **Escalabilidad:**
    - Evolución del incremento (en ganancia) en prestaciones (tiempo de ejecución o productividad) que se consigue en el sistema conforme se añaden recursos

Prestaciones en arquitecturas paralelas (2)

- Medidas que sirven para evaluar las prestaciones
  - **RAS** (Reliabitliy, Availabitliy, Serviceabitliy)
    - **Fiabilidad** (reliabitliy)
      - Representa la probabilidad de que un sistema funcione conforme a sus especificaciones durante un periodo de tiempo.
      - Está relacionada la frecuencia de fallos
    - **Disponibilidad** (availabitliy)
      - Está relacionada con la penalización que suponen los fallos de un sistema
    - **Serviciabilidad** (serviceabitliy)
      - Mide la facilidad con la que un técnico (HW) puede realizar el mantenimiento (para prevenir o corregir) de un sistema
  - **Eficiencia:**
    - Evalúa en qué medida las prestaciones que ofrece un sistema para sus aplicaciones, se acerca a las prestaciones máximas que idealmente debería ofrecer
    - En qué medida se aprovechan los recursos del sistema