

## SISTEMAS DISTRIBUIDOS PROCESAMIENTO PARALELO

### Introducción

- **Concepto de muchos campos:**
  - Sistemas Operativos
  - Procesamiento Paralelo (PP)
  - Sistemas en Tiempo Real
  - Sistemas Multimedia Distribuidos
  - Tipos de comunicaciones:
    - SD, PP, STR
  - Bases de datos distribuidas
  - Cliente / Servidor, Internet, 2 vs 3 niveles
  - Objetos/Componentes Distribuidos
  - Computación móvil / agentes
  - Peer to peer
  - Aplicaciones tradicionales

### Introducción (cont.)

- **S.O:**
  - S.O genéricos + red vs S.O. distribuidos
- **Procesamiento Paralelo:**
  - Conjunto de procesadores los cuales están en una misma máquina
  - Que tipo de problemas resuelven? Rpta. Reducir el tiempo de ejecución
- **STR:**
  - Interactúan con el entorno, controla un proceso y posiblemente lo modifica.
  - El correcto funcionamiento del sistema depende del tiempo en el que se produzcan los datos mas que la precisión de éstos.
  - Principalmente aplicados a "Sistemas de Control"
  - Hay 2 categorías: "duros" (hard real-time) y "blandos" (soft real-time)
  - Los sistemas multimedia son SRT

### Introducción (cont.)

- **Sistemas de comunicaciones**
  - Se agregan automáticamente cuando se refieren a "Distribuidos"
  - Cuales arquitecturas? ATM, TCP/IP, LANs, ???
  - Dependen del tipo de aplicación o sistema distribuido.
- **Sistemas Multimedia Distribuidos**
  - Sistemas en Tiempo Real "blandos"
  - tipos de medios: Audio, Video, Imágenes, datos, etc.
  - Impone altas restricciones en las redes y S.O.
  - Muchos conceptos: dispositivos, codificadores, redes, TR, QoS, sincronización, BD, lenguajes, frameworks, middlewares, apps...
- **BD Distribuidas**
  - Se han tenidos que desarrollar como un campo aparte
  - De existir SOD todas sus apps por defecto son Distribuidas
  - Alta vulnerabilidad a la red

### Introducción (cont.)

- **La aplicación más tradicional de los SD es el modelo C/S para aplicaciones tradicionales.**
- **Que otros campos o aplicaciones?**
  - Multimedia Networking
  - Lenguajes de programación concurrentes
  - SVA
  - realidad virtual

### Sistema Distribuidos vs Procesamiento Paralelo

- **SD:**
  - Conjunto de máquinas/CPUs que conforma un sistema multicomputadora, los cuales poseen una memoria privada o propia por nodo interconectados a través de una red y que hacen ver la usuario como si tuviera un computador único virtual - Monoprocesador virtual
  - Sistemas Débilmente Acoplados (loosely-coupled systems)
  - Memoria Distribuida y retardo en la red (LCS)
  - Objetivo: tolerancia a fallos, disponibilidad y compartir recursos
  - No es un problema de HW sino de SW
  - Procesamiento Distribuido? PVM, MPI, etc.
  - Ventajas:
    - Económicas (mejor relación precio/desempeño que mainframes), Velocidad, Distribución inherente, Confiabilidad, crecimiento horizontal, Compartir recursos (datos - BD, dispositivos, comms)
  - Desventajas:
    - Software complejo, redes, seguridad

universidad EAFIT

### SD vs PP (cont.)

- **PP:**
  - Conjunto de CPUs en un mismo computador que comparten una misma memoria
  - Sistemas Fuertemente Acoplados (tightly-coupled systems)
  - Memoria compartida y hay poco retardo entre CPUs
  - Objetivo: reducir el tiempo de ejecución de una aplicación

Sistemas Distribuidos - Introducción Edwin Montoya

universidad EAFIT

### SD vs PP (cont.)

- **SD problema de SW**
- **PP problema de HW**
- **HW para PP**
  - ¿Cómo se interconectan las CPUs?
  - No de instrucciones
  - No de datos (memoria)
  - Clasificación Flynn para PP:
    - SISD: Single Instruction Single Data (sist. Uniprocador)
    - SIMD: Single Instruction Multiple Data (proc. Vectoriales, muchos registros de datos)
    - MISD: Multiple Instruction Single Data (NO existen)
    - MIMD: Multiple Instruction Multiple Data (PP y SD)
      - Existen dos clases de MIMD
      - Multiprocador (PP, TCS)
      - Multicomputador (SD, LCS)

Sistemas Distribuidos - Introducción Edwin Montoya

universidad EAFIT

### Taxonomía de sistemas distribuidos y paralelos (Tanenbaum)

```

    graph TD
      MIMD[MIMD] --> CPD[Computadores Distribuidos y en Paralelo]
      CPD --> FA[Fuertemente acoplados]
      CPD --> DA[Debilmente acoplados]
      FA --> MP[Multiprocador Memoria compartida]
      DA --> MC[Multicomputador Memoria privada]
      MP --> MB[En BUS]
      MP --> C[Conmutados]
      MC --> MB
      MC --> C
  
```

Sistemas Distribuidos - Introducción Edwin Montoya

universidad EAFIT

### Multiprocador en BUS

- **Buses de datos, control y direcciones**
- **Que se debe asegurar:**
  - Solo una CPU acceda la memoria al tiempo
  - Para no generar contención hacia la única memoria, se utiliza un cache por procesador de muy alta velocidad
  - Otro problema: Como garantizar coherencia en una memoria global?
  - Cada cambio en la memoria cache se debe reflejar en la memppal
  - Procesador coherente: Cuando se garantiza que un solo procesador maneja un dato en un determinado momento. Se controla el acceso concurrente.
  - Caches de 64 Kb a 1 Mb presenta aciertos muy altos.
  - Políticas de manejo de cambios en el cache:
    - Read-Back
    - Write Through

Sistemas Distribuidos - Introducción Edwin Montoya

universidad EAFIT

### Multiprocador Conmutada

- Se tiene una malla entre CPUs y Memorias.
- Crosspoints (switching)
- Acceso exclusivo por Memoria.
- Multicomputador en BUS
- Procesamiento en redes convencionales
- Multicomputador/multiprocador Conmutada
- Grid / Rejilla
- Canales de alta velocidad entre computadores
- Problema: El número de conexiones crece exponencialmente con el número de nodos
- CPU/M -> malla o hypercubo

Sistemas Distribuidos - Introducción Edwin Montoya

universidad EAFIT

### Software para SD / PP

- Así como en HW existe LCS y TCS, de la misma manera existe SW de los dos tipos (LCSw y TCSw)
- Como se relaciona el HW y el SW
- LCSw tiene dos modalidades
  - C/S
  - verdaderos S.O.D.
- C/S
  - Funciona sobre LCH (HW debil/. Acopl.)
  - El S.O. Por cada nodo puede ser distinto
  - Reglas de comunicación en red (request / reply)
  - NFS es un ejemplo de LCSw con C/S
    - No hay dif. Entre una llamada local o remota

Sistemas Distribuidos - Introducción Edwin Montoya

### Software para SD / PP (cont.)

- **Verdaderos S.O.D:**
  - Todos los S.O. Comparten el mismo kernel lo que significa:
    - igual system calls
    - igual política de scheduler
    - No hay dif. Entre una llamada local o remota
    - Ej: Locus, Math Spring, Amoeba.
- **TCSw:**
  - requiere LCHW
  - se implementa en sistemas embebidos
  - Lenguajes concurrentes
  - Multihilos
  - SMP vs AMP

### Metas o aspectos de diseño

- **Existen diferentes parámetros para evaluar S.D:**
  - Transparencia
  - Escalabilidad
  - Rendimiento
  - Flexibilidad
  - Interoperabilidad
- **Uno de los más importantes es “transparencia”, el cual tiene varios ítemes:**
  - acceso
  - localización
  - replicación
  - migración
  - fallas
  - conurrencia

### Sistemas Operativos Distribuidos

- **Conceptos interesantes pero fuera de la parte comercial**
- **Porque no existen o se han desarrollado SOD comerciales?**
  - Que se va a distribuir?
    - Procesos
    - Almacenamiento
    - gran competencia en los middlewares y máquinas virtuales
    - Seguridad
    - Homogeneidad vs Interoperabilidad
- **Conclusión:**
  - S.O. Convencionales con aplicaciones distribuidas
  - C/S consolidado
  - Interoperabilidad

### Modelos de Sistemas Distribuidos

- **WS / Server: Existe una LAN la cual comparte almacenamiento, dispositivos únicos en el servidor y posiblemente procesamiento.**
- **Pull de procesadores:**
  - Conjunto de CPUs en una máquina
  - Posiblemente cada CPU corre un S.O. Diferente, dependiendo de los requerimientos del usuario (loader)
- **Modelo Integrado**
  - Reúne toda la capacidad gráfica de la WS con la potencia y flexibilidad de un Pull de procesadores.
  - Modelo muy vigente actualmente
    - E1000 de Sun
    - SP de IBM