UNIVERSIDAD DE LA REPÚBLICA URUGUAY

TEMA 2
ARQUITECTURAS PARALELAS

Contenido

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

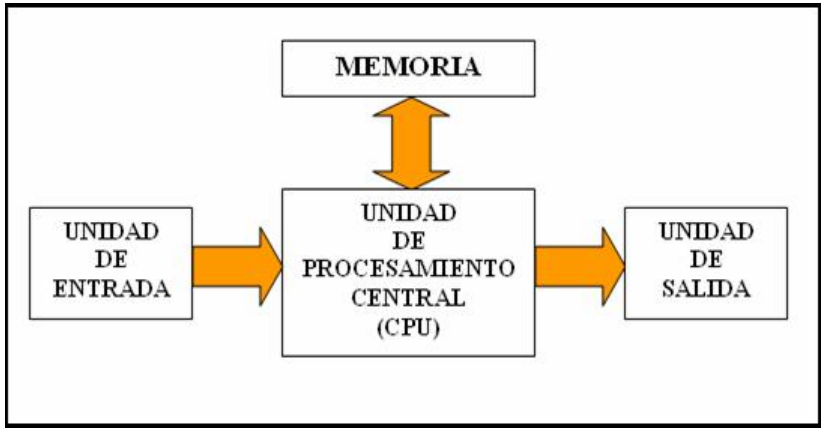
- Arquitecturas secuenciales y paralelas
  - Clasificación de Flynn
  - Modelo SIMD
  - Modelo SISD
  - Modelo SIMD
  - Arquitectura MIMD
    - MIMD con memoria compartida
    - MIMD con memoria distribuida
- Factores que determinan la eficiencia
- Máquina paralela virtual
- Clusters
- Arquitecturas multinúcleo

2.1: ARQUITECTURAS SECUENCIALES Y PARALELAS

Arquitecturas paralelas

- Modelo estándar de computación:

  Arquitectura de Von Neumann
  - CPU única
    - Ejecuta un programa (único)
    - Accede a memoria
  - Memoria única
    - Operaciones read/write
  - Dispositivos



Arquitectura de Von Neumann

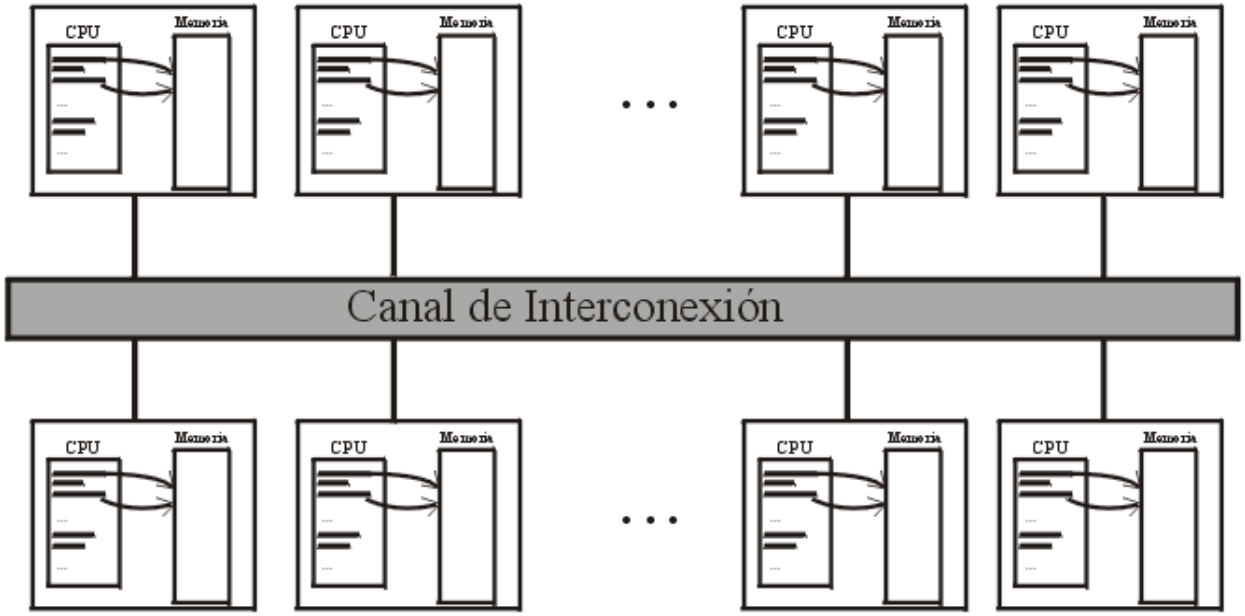
- Modelo robusto, independiza al programador de la arquitectura subyacente.

Neumann János

- Permitió el desarrollo de las técnicas de programación (estándar)

Arquitecturas paralelas

- Extendiendo el modelo a la computación paralela, para lograr abstraer el hardware subyacente
- Existen varias alternativas, genéricamente contempladas en el modelo del multicomputador:
  - Varios nodos (CPUs de Von Neumann)
  - Un mecanismo de interconexión entre los nodos



Multicomputador (de memoria distribuida)

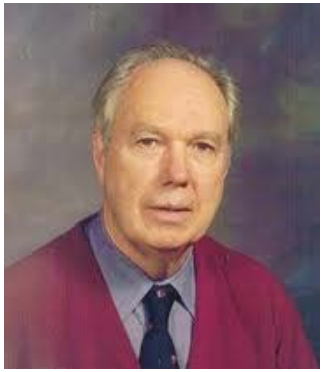
Arquitecturas paralelas

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Extendiendo el modelo a la computación paralela ...
- Otras alternativas
  - Multiprocesador de memoria compartida
    - Nodos de Von Neumann
    - Memoria única
  - Computador masivamente paralelo
    - Muchísimos nodos (sencillas CPUs estilo Von Neumann)
    - Topología específica para interconexión entre los nodos
  - Cluster
    - Multiprocesador que utiliza una red LAN como mecanismo de interconexión entre sus nodos

Categorización de Flynn

- Clasificación de arquitecturas paralelas que considera la manera de aplicación de las instrucciones y el manejo de los datos



Michael Flynn

		Instrucciones	
		SI	**MI**
Datos	**SD**	**SISD**	**(MISD)**
	MD	**SIMD**	**MIMD**

Taxonomía de Flynn (1966)

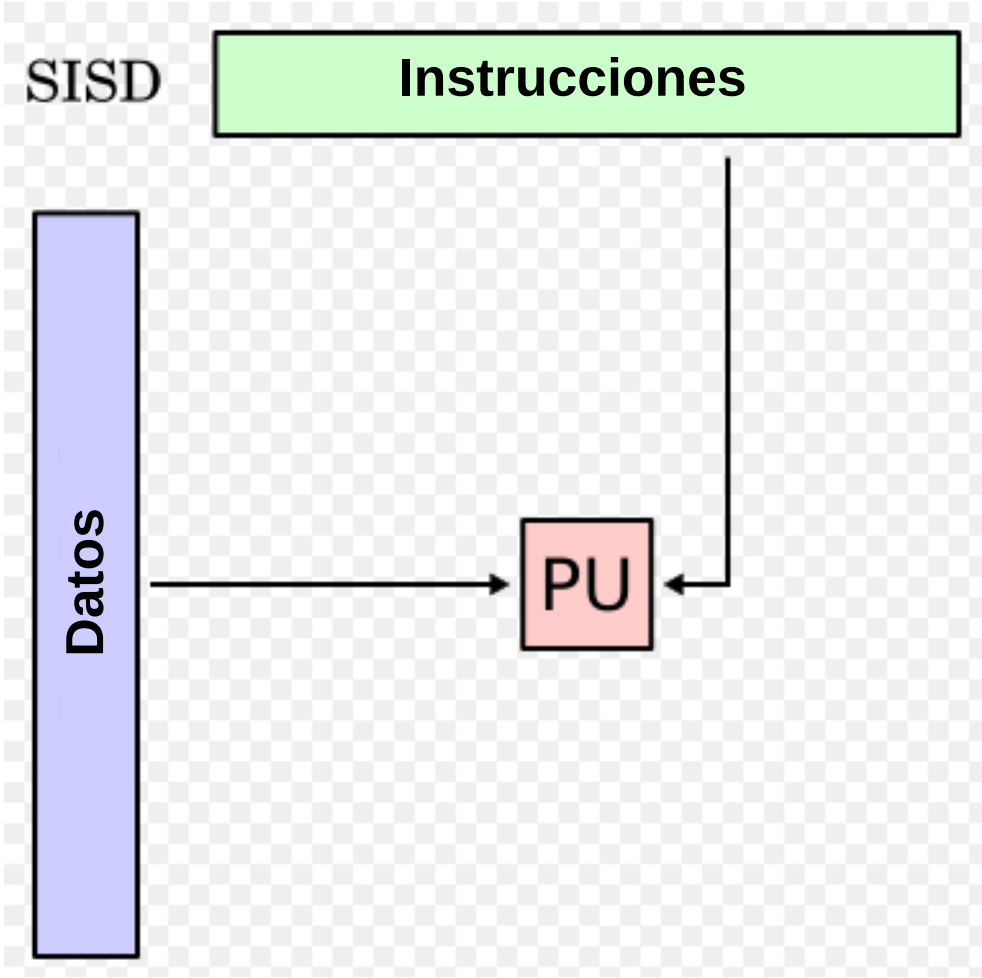
S=single, M=multi, I=Instrucción, D=Datos

Categorización de Flynn

- SISD – Modelo convencional de Von Neumann
- SIMD – Paralelismo de datos, computación vectorial
- MISD – Pipelines, arrays sistólicos
- MIMD – Modelo general, varias implementaciones

- El curso se enfocará en el modelo MIMD, utilizando procesadores de propósito general o clusters de computadores
- El modelo SIMD se estudiará enfocado en el procesamiento de propósito general en aceleradores y procesadores gráficos

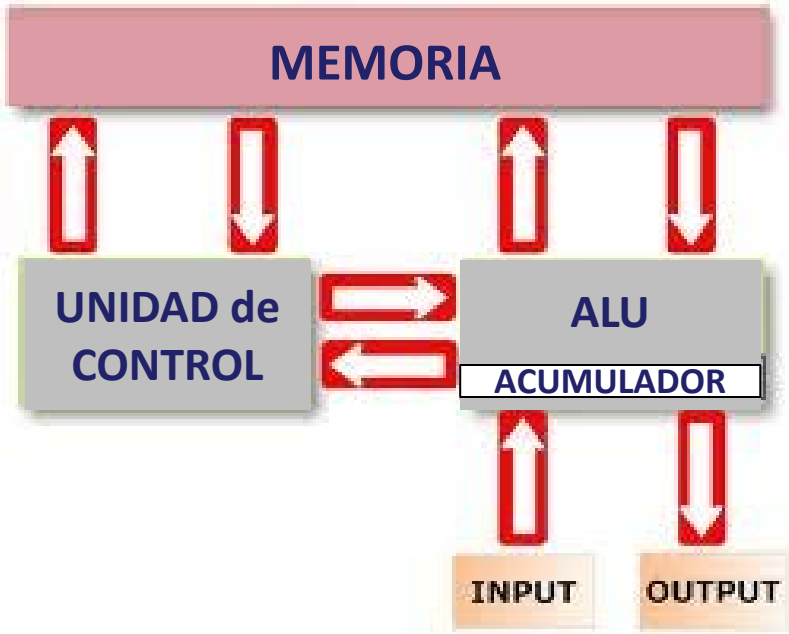
Categorización de Flynn: SISD



Single Instruction Single Data

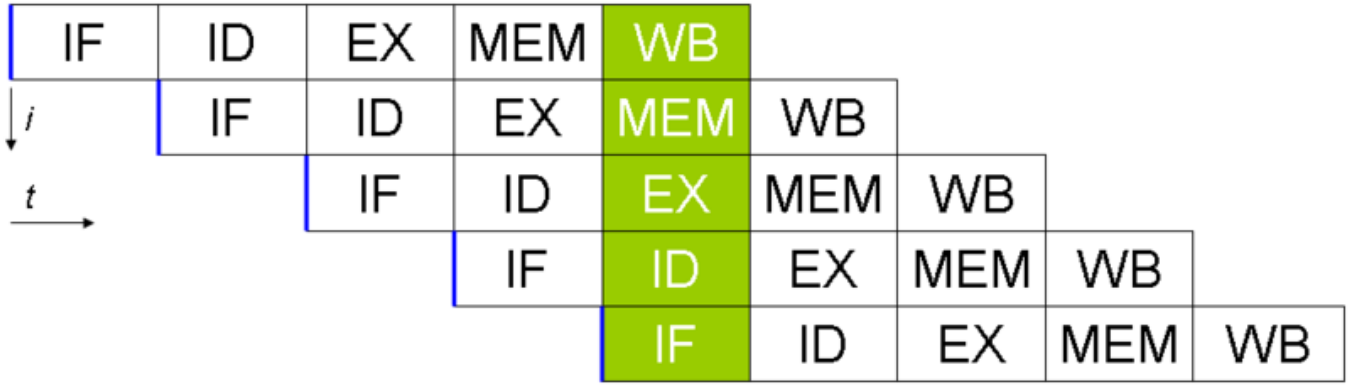
Arquitectura SISD

- Máquina de Von Neumann
  - Un procesador capaz de realizar operaciones aritmético-lógicas secuencialmente, controlado por un programa que se encuentra almacenado en una memoria conectada al procesador
  - Este hardware está diseñado para dar soporte al procesamiento secuencial clásico, basado en el intercambio de datos entre memoria y registros del procesador, y la realización de operaciones aritmético-lógicas en ellos



Arquitectura SISD

- En la década de 1980 se diseñaron computadores secuenciales que no correspondían estrictamente al modelo SISD
- A partir de la introducción de los procesadores RISC se comenzaron a utilizar varios conceptos de las arquitecturas paralelas, como el pipelining, la ejecución paralela de instrucciones no dependientes, el prefetching de los datos, etc., para lograr un incremento en la cantidad de operaciones realizadas por ciclo de reloj

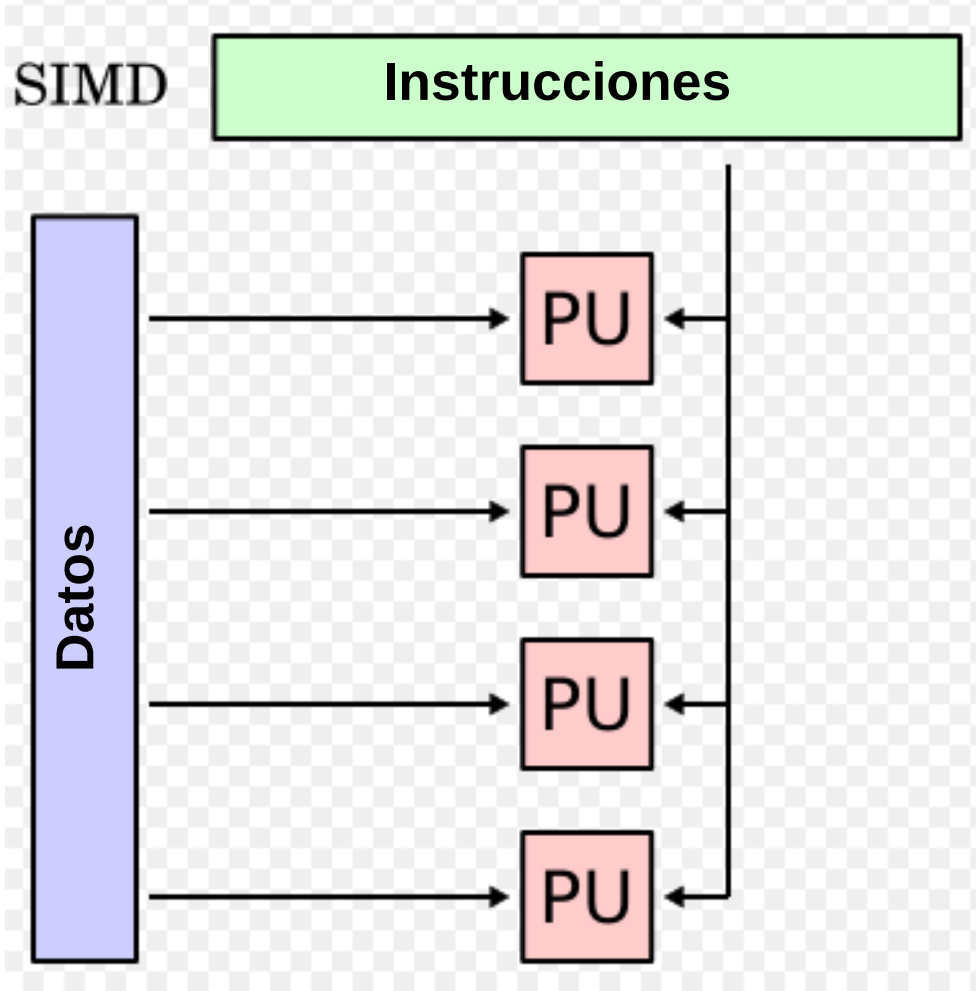


Pipeline

Arquitectura SISD

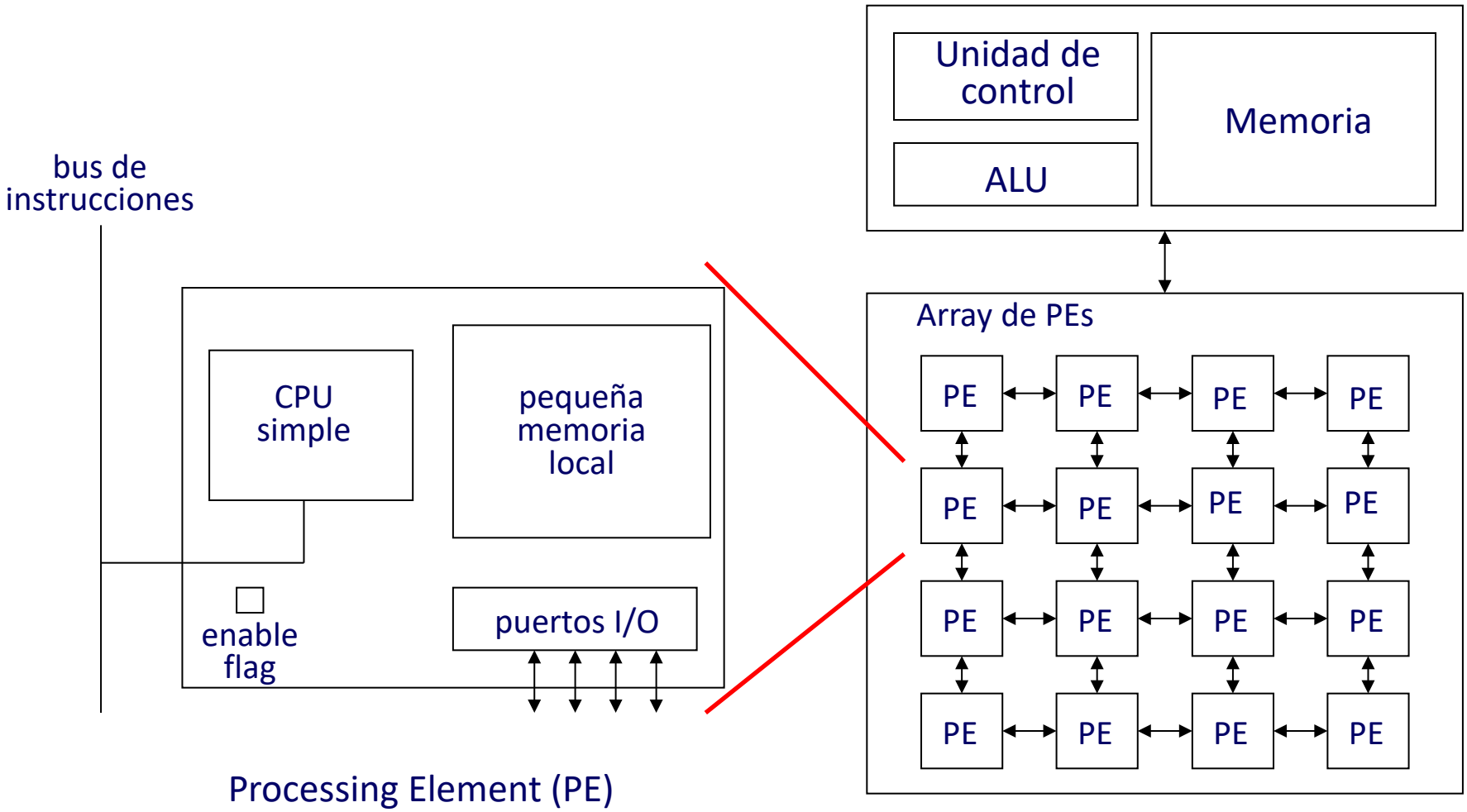
- Ley de Moore: la capacidad de procesamiento se duplica cada 24 (18) meses aproximadamente.
- Aún mejorado, el modelo SISD no fue suficiente.
- Los problemas crecieron, o surgió la necesidad de resolver nuevos problemas de grandes dimensiones (manejando enormes volúmenes de datos, mejorando la precisión de las grillas, etc.).
- Si bien las maquinas SISD mejoraron su performance
  - Arquitecturas CISC y RISC.
  - Compiladores optimizadores de código.
  - Procesadores acelerando ciclos de relojes, etc.
- Aún no fue suficiente, y el ritmo de mejoramiento se desaceleró (principalmente debido a limitaciones físicas).
- En este contexto se desarrollaron los computadores paralelos.

Categorización de Flynn: SIMD



Single Instruction Multiple Data

Arquitectura SIMD



Arquitectura SIMD

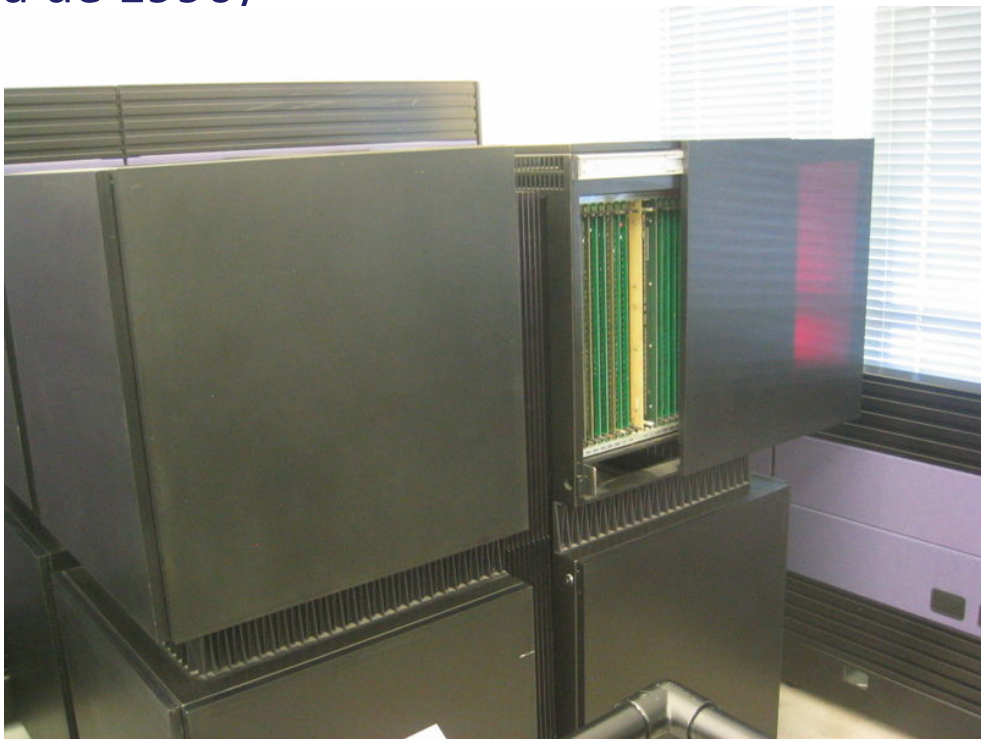
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Ún unico programa controla los procesadores.
- Útil en aplicaciones uniformes
  - Procesamiento de imágenes
  - Aplicaciones multimedia
  - Aplicaciones numéricas sobre grillas
- Su aplicabilidad está limitada por las comunicaciones entre procesadores.
  - La topología de interconexión es fija
- Los elementos de procesamiento tienen capacidad de cómputo limitada (1 bit a 8 bits), por lo que se pueden colocar una gran cantidad por chip (e.g. Connection Machine 2 con 64k PEs)
- Fueron los primeros multiprocesadores difundidos comercialmente (en la década de 1980)

Arquitectura SIMD

- Ejemplos comerciales (históricos)
  - Cray X-MP (computador más potente entre 1983–1985)
  - Connection Machine (CM-2, CM-200, década de 1980)
  - MasPar MP2 (inicios de la década de 1990)

CRAY X-MP/24



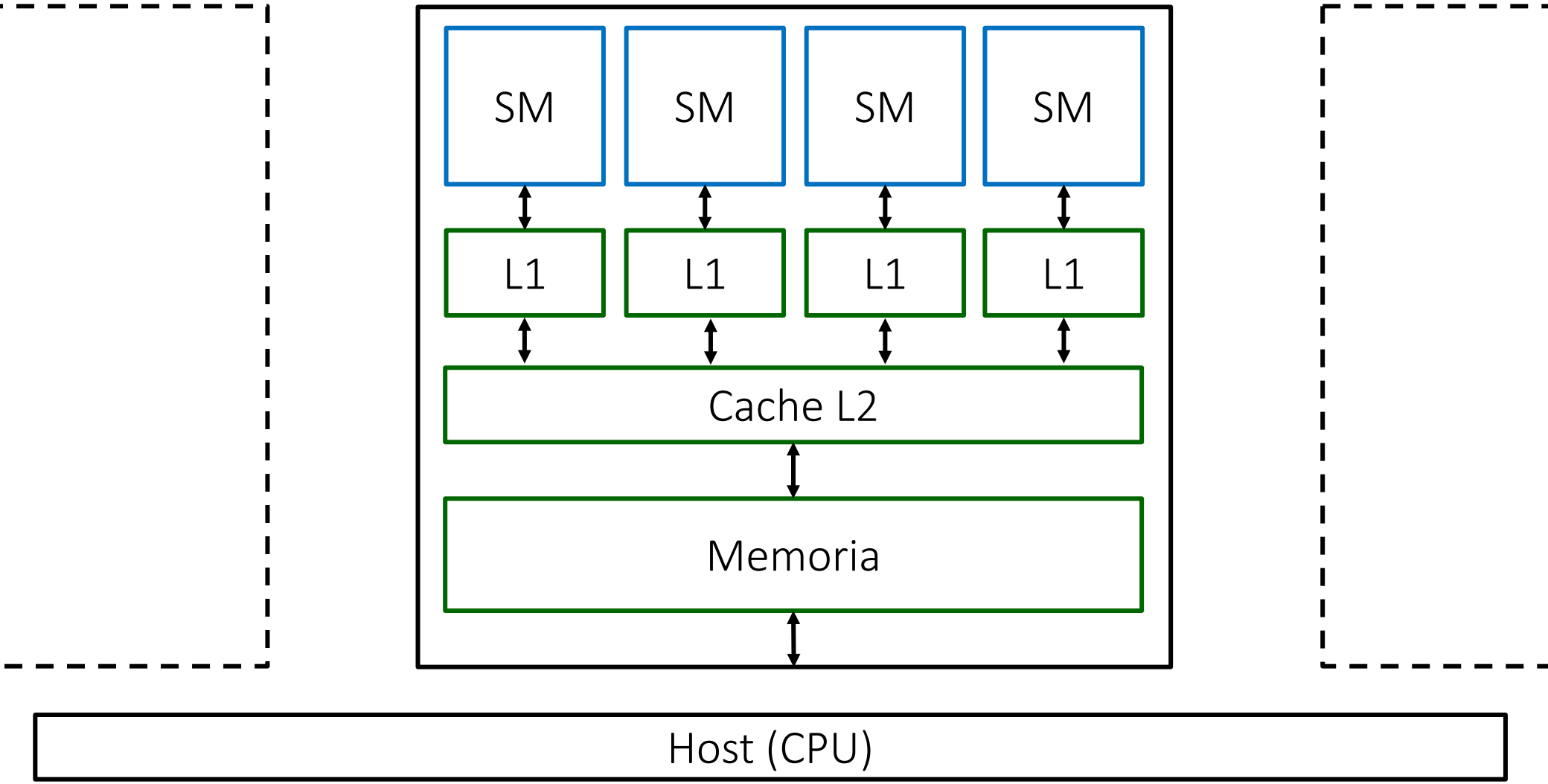
Connection Machine CM-2

Arquitectura SIMD

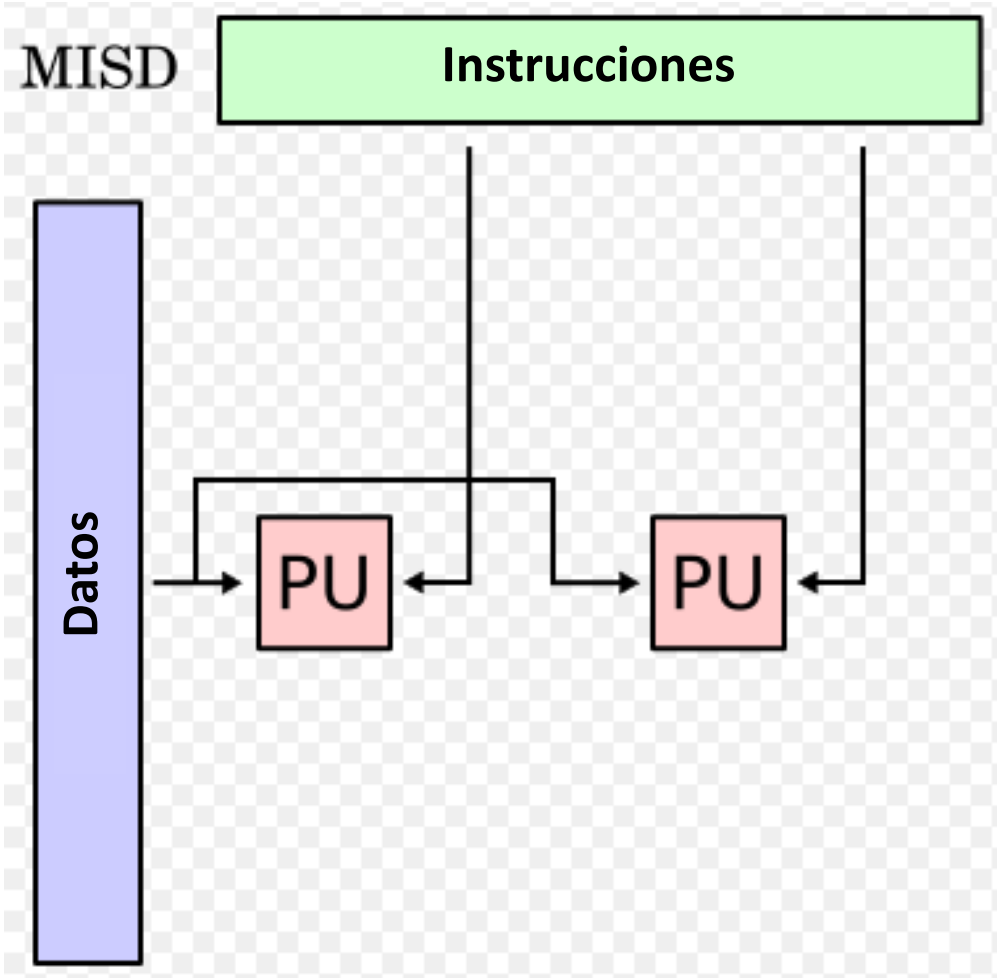
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- GPUs (Graphics Processing Units): multithreading SIMD
- Arquitectura optimizada para “computación raster” (basada en grillas).
- Arreglos multidimensionales (matrices) 2D, etc.
  - A diferencia de CPUs multicore, que tienen cores optimizados para la ejecución de threads.
- GPUs: memoria de alto ancho de banda (y alta latencia), caches compartidas por (un gran número de) procesadores multistreaming que permiten ejecutar multiples threads.
- Las diferencias en la arquitectura implican que los programas deban diseñarse y optimizarse de manera diferente para CPU y GPU.
- GPUs especializadas para paralelismo de datos.

Arquitectura SIMD: GPUs



Categorización de Flynn: MISD



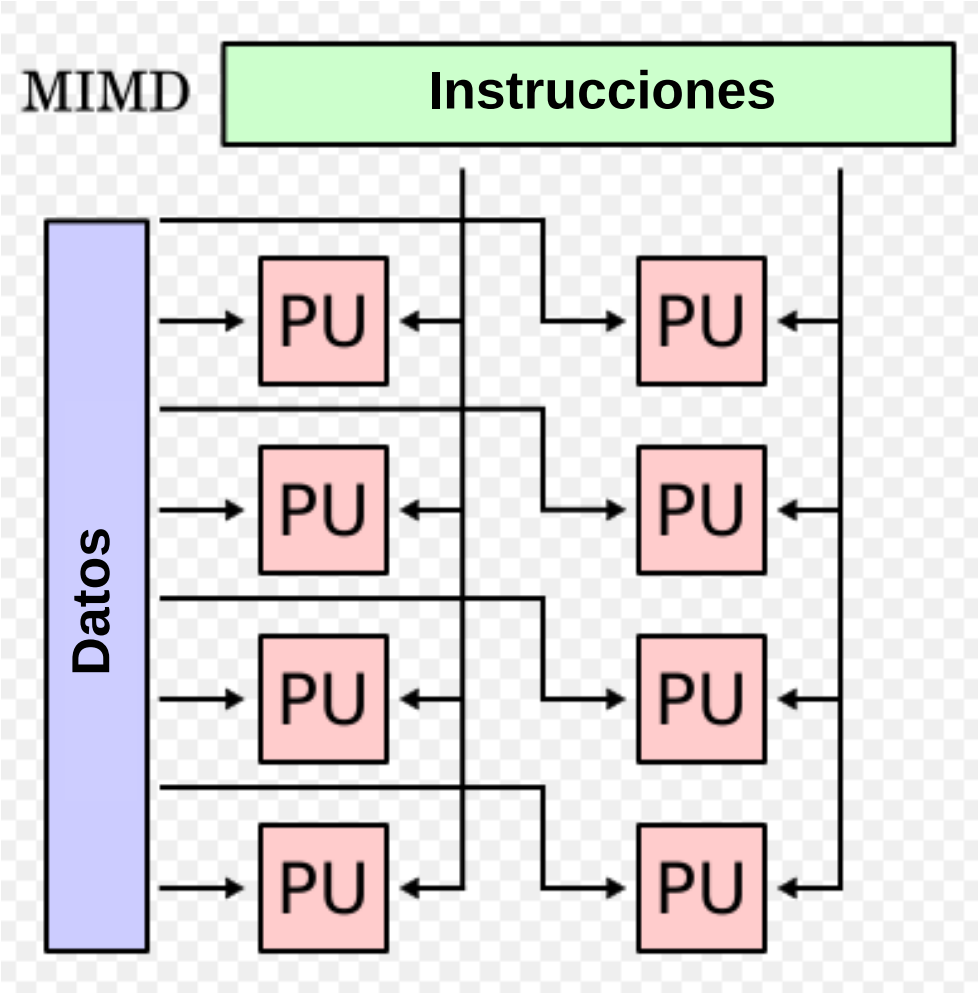
Multiple Instruction Single Data

Arquitectura MISD

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Varias unidades funcionales ejecutan diferentes operaciones sobre el mismo conjunto de datos.
- Las arquitecturas de tipo pipeline pertenecen a esta clasificación
  - Aunque no puramente, ya que pueden modificar los datos sobre los que operan.
- Otros modelos: arrays sistólicos, FPGA celulares.
- También pertenecen al modelo MISD los computadores tolerantes a fallos que utilizan ejecución redundante para detectar y enmascarar errores.
- No existen otras implementaciones específicas.
- Los modelos MIMD y SIMD son más apropiados para la aplicación del paralelismo tanto a nivel de datos como de control.

Categorización de Flynn: MIMD

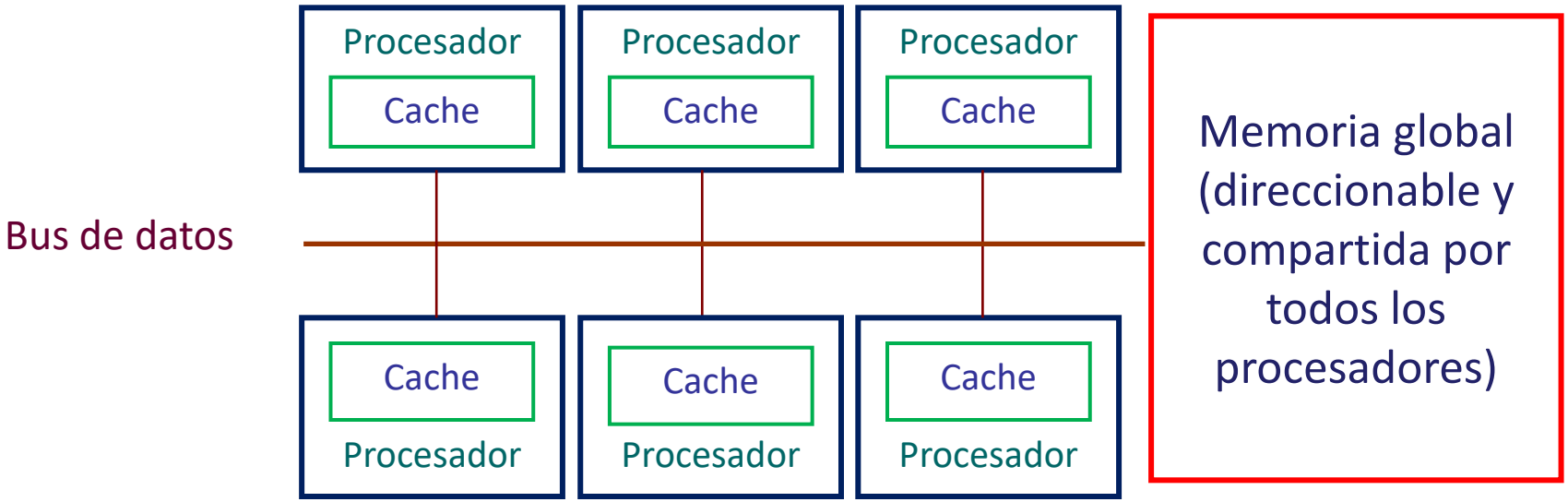


Multiple Instruction Multiple Data

Arquitectura MIMD

- Consistieron en el “siguiente paso” en la evolución de las arquitecturas paralelas.
  - Fueron desplazando lentamente al modelo SIMD.
- A diferencia de los modelos SISD y MISD, las computadoras MIMD pueden trabajar asincrónicamente
  - Los procesadores tienen la capacidad de funcionamiento semi-autónomo.
- Existen dos tipos de computadores MIMD, de acuerdo al mecanismo utilizado para comunicación y sincronización:
  - MIMD de memoria compartida (fuertemente acopladas).
  - MIMD de memoria distribuída (poco acopladas).

MIMD con memoria compartida



Arquitectura MIMD con memoria compartida.

- Procesadores autónomos, trabajan asincrónicamente
- Comunicación entre procesadores a través del recurso compartido
  - Comunicación y sincronización se realiza en forma explícita
  - Emisor escribe y receptor lee de la memoria global

MIMD con memoria compartida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Fáciles de construir
  - SO convencionales de los SISD son portables.
- Buena solución para procesamiento transaccional (sistemas multiusuario, bases de datos, etc.)
- Limitaciones: confiabilidad y escalabilidad
  - Un fallo de memoria de algún componente puede causar un fallo total del sistema.
- Incrementar el número de procesadores puede llevar a problemas en el acceso a memoria
  - Caso de supercomputadores Silicon Graphics
- El bus (cuello de botella) limita la escalabilidad a un máximo de pocas decenas de procesadores.
  - Se puede mejorar usando caches locales, pero se introduce el problema de “coherencia de cache”

MIMD con memoria compartida

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Mecanismos para compartir los datos
- UMA = Uniform Memory Access
  - Acceso uniforme (todos los procesadores tienen el mismo tiempo de acceso a la memoria)
  - Multiprocesadores simétricos (SMP)
  - Pocos procesadores (32, 64, 128, por problemas de ancho de banda del canal de acceso)
- NUMA = Non-Uniform Memory Access.
  - Colección de memorias separadas que forman un espacio de memoria direccionable
  - Algunos accesos a memoria son más rápidos que otros, como consecuencia de la disposición física de las memorias (distribuidas físicamente)
  - Multiprocesadores masivamente paralelos (MPP)

MIMD con memoria compartida

- Ejemplos:
  - Encore MULTIMAX
    - Inicios de la década de 1990, hasta 20 procesadores
  - Sequent Symmetry
    - Década de 1990, de 10 a 30 procesadores
  - Sun SPARCcenter 2000
    - Escalable hasta 20 procesadores
    - Sun-4d (d por *dragon*, nombre código del SPARCcenter 2000)
    - La Facultad de Ingeniería tuvo un supercomputador de este tipo desde 2000 a 2008

Sun SPARC Center 2000

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Arquitectura SPARC (Scalable Processor ARChitecture), versión 8
  - Arquitectura RISC
  - Microprocesadres SuperSPARC II 85 Mhz
  - Direccionamiento 32 bits
  - Arquitectura de bus multinivel
- Clasificación:
  - MIMD de memoria compartida
- Sistema altamente acoplado (tighthly coupled system)
- Diseño Modular
  - Unidad Procesador
  - Unidad Memoria
  - Unidad Entrada/Salida

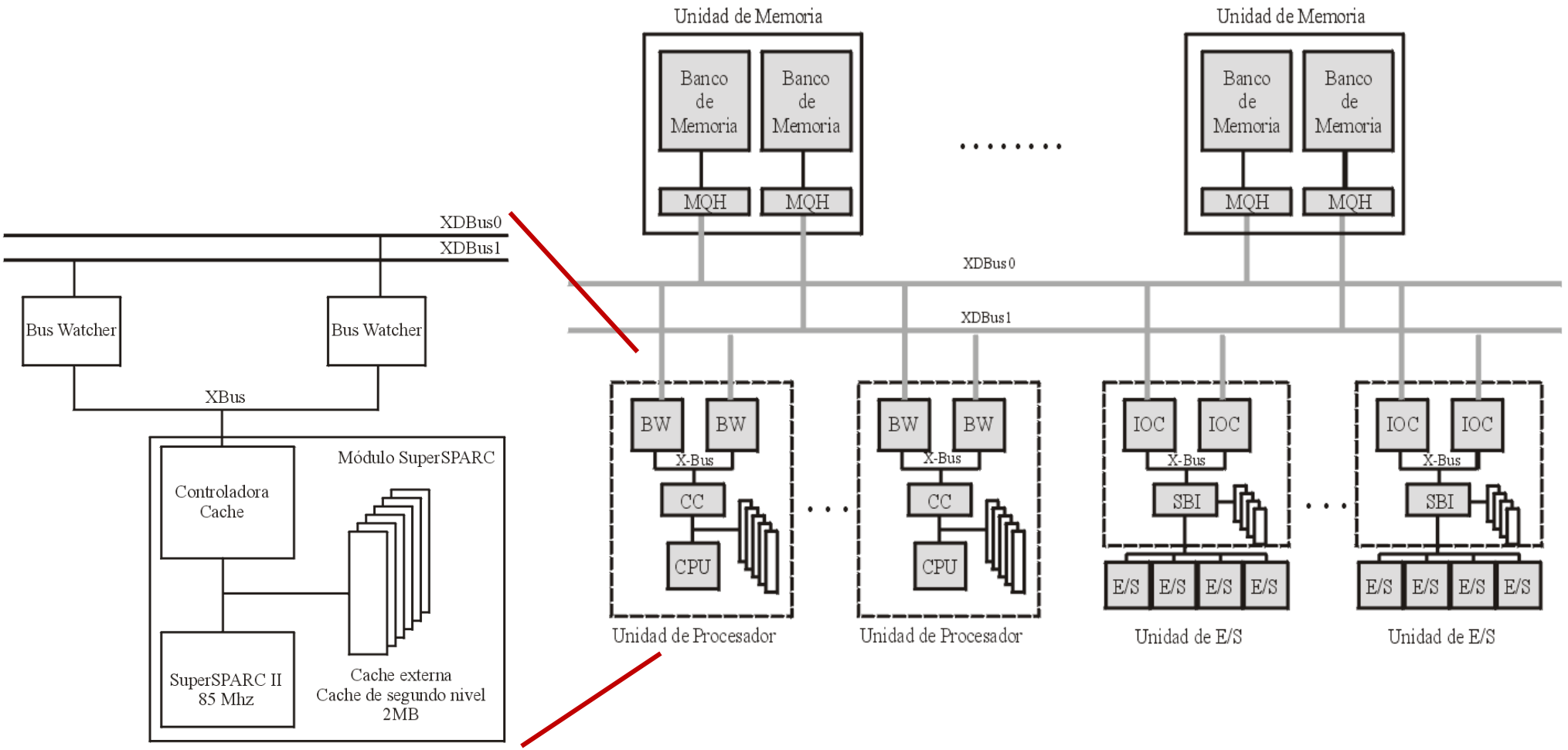
Sun SPARC Center 2000

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Componentes:
  - Placa de control
  - 10 placas de sistema
  - 20 procesadores SuperSPARC II (85 Mhz), 2 por placa de sistema
  - 5 GB de memoria RAM, 512 MB por placa de sistema
  - 40 dispositivos de entrada/salida, 4 por placa de sistema
  - 2 MB Level 2 cache por CPU
  - Bus Multinivel:
    - XDBUS, conecta unidades lógicas, canal de 72 bits (64 datos + 8 paridad), 400 Mb/s, packet-switched, sincrónico, coherencia de caché
    - XBUS, conecta BW y CC, similar al XDBUS
    - SBUS, conecta placa y dispositivos de entrada/salida

Sun SPARC Center 2000

- Esquema de arquitectura



MIMD con memoria distribuida

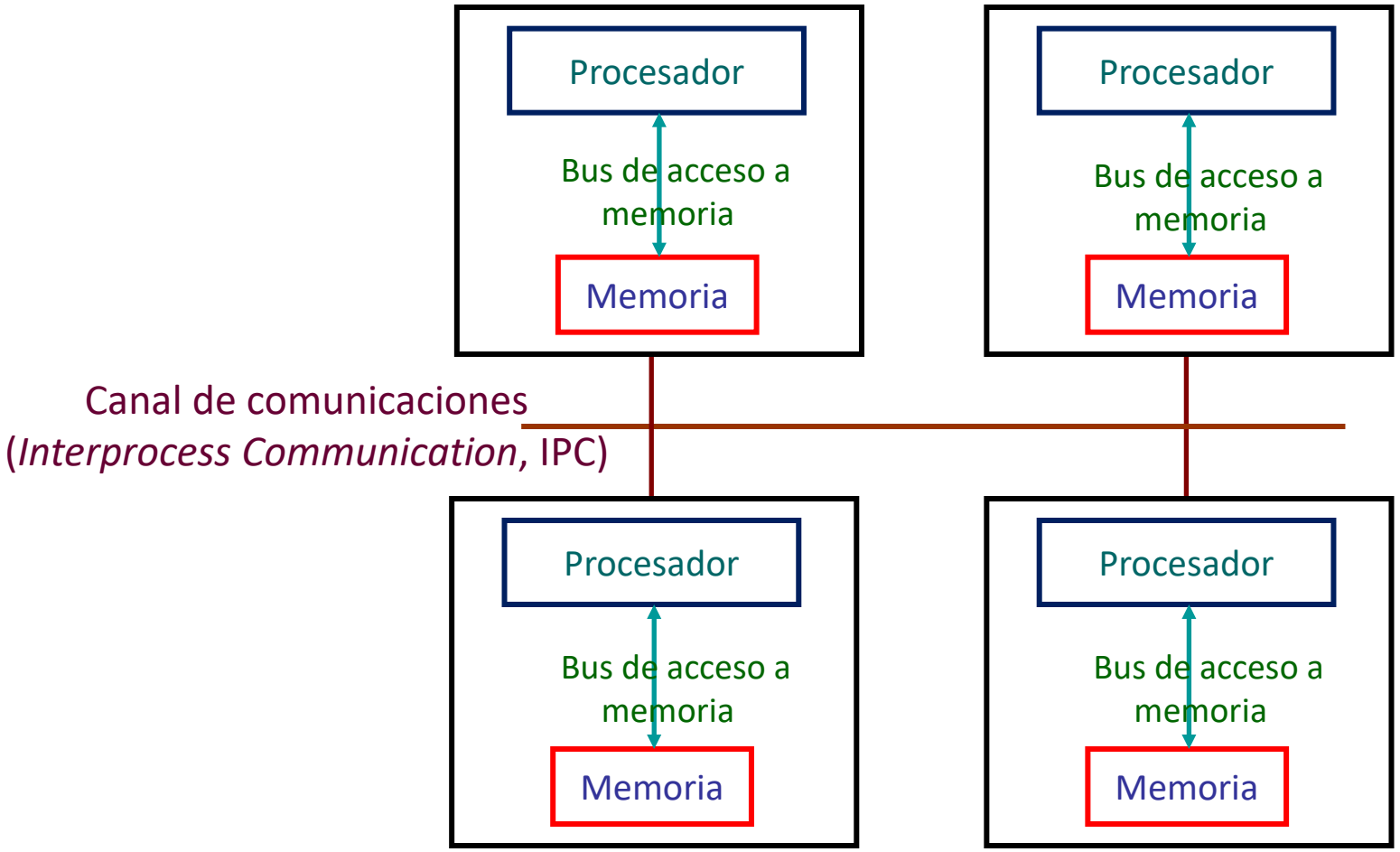
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- No existe el concepto de memoria global
- Comunicación y sincronización:
  - Mecanismos explícitos de IPC (pasaje de mensajes) sobre una red (en escenario óptimo, red de alta velocidad)
  - Tienen mayor costo que en MIMD de memoria compartida
- Las comunicaciones pueden ser cuellos de botella.

- Arquitectura escalable para aplicaciones apropiadas:
  - Decenas de miles de procesadores
  - Popularizadas en clusters

- Ventajas respecto a MIMD de memoria compartida
  - Fácilmente escalable
  - Alta disponibilidad: el fallo de una CPU individual no afecta a todo el sistema

MIMD con memoria distribuida



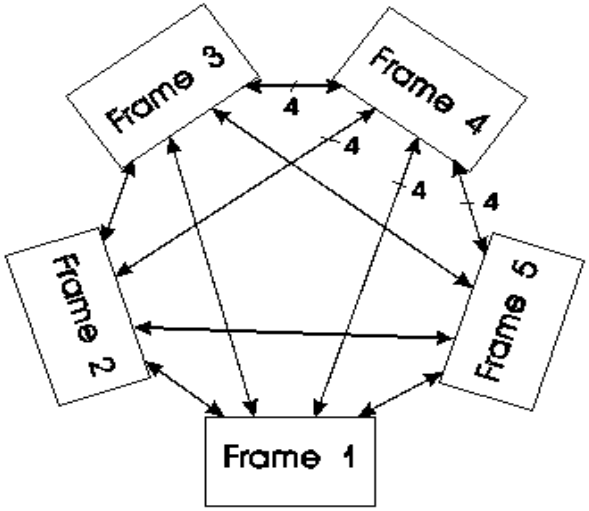
Arquitectura MIMD con memoria distribuida

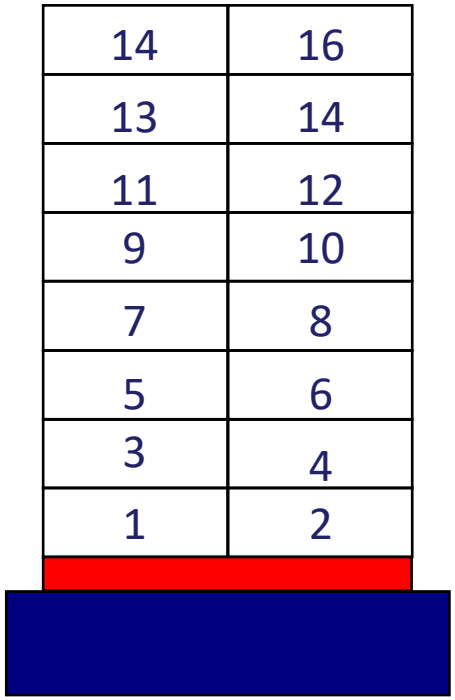
MIMD con memoria distribuida

- Ejemplos comerciales
  - Connection Machine CM-5 (1991, 16k procesadores).
  - Intel Paragon (1992: 2048 procesadores, luego 4000).
  - Cray
    - Luego de fusión con SGI: Scalable Node SN1, SN2.
    - T3E, 1997, hasta 2048 procesadores.
  - IBM SP (IBM Scalable POWER parallel)
    - Incluía tecnología High Performance Switch (HPS) para comunicación entre nodos.
    - Década de 1990 e inicios de los 2000
    - En 1999 incorporaron procesadores POWER3, en 2001 POWER4 y en 2004 POWER5.

IBM SP-2

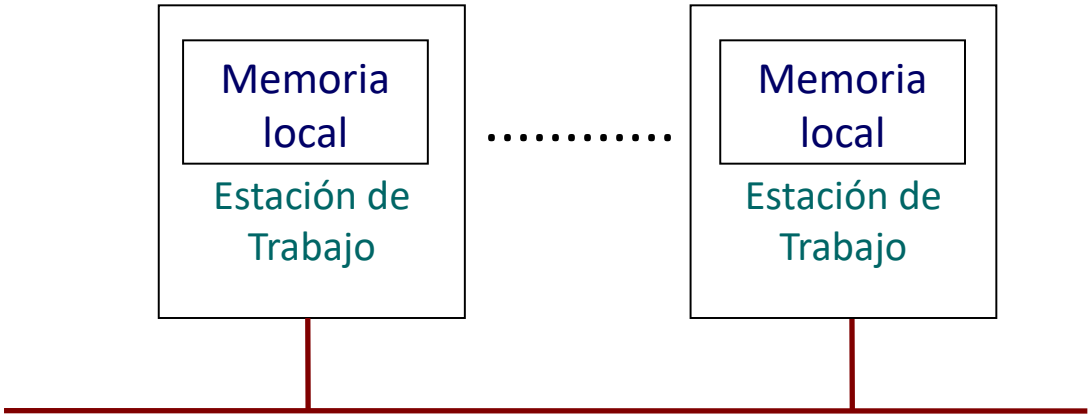
- Uno o más SP (Scalable POWER) frames
  - SP: Basado en RS/6000
  - 2 a 16 máquinas por frame
- Switch de alto desempeño:
  - Conexión bidireccional todos con todos
  - Packet-switched network (versus circuit-switched)
- Instalaciones en Uruguay: BPS, DGI, UTE, IBM





MIMD con memoria distribuida

CASO PARTICULAR (1)



MÁQUINA PARALELA VIRTUAL

Red de datos
(LAN, WAN, Ethernet, FastEthernet, Gigabit Ethernet, FDDI, etc).

VENTAJAS

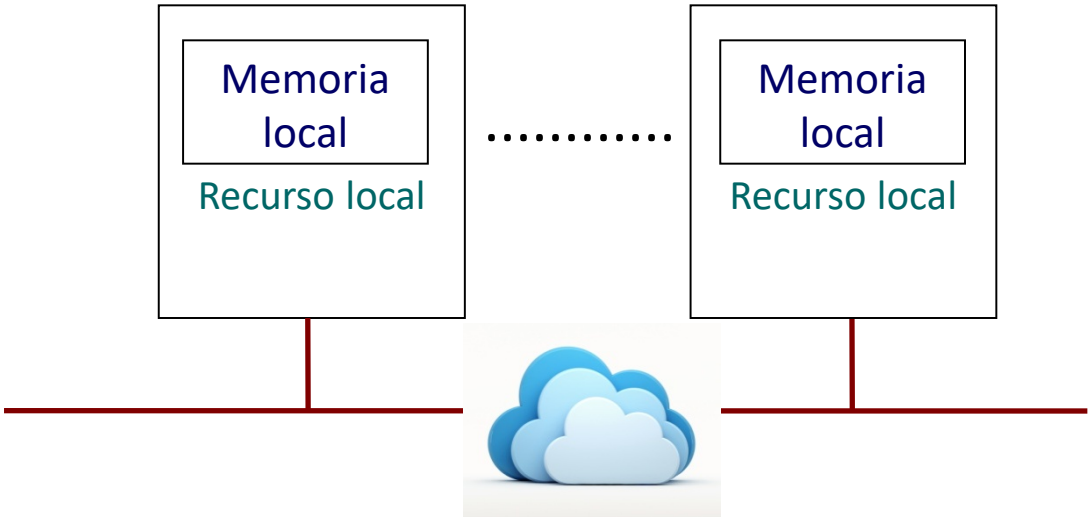
- Usa infraestructura existente
- Sistema escalable
- Fácilmente programable

DESVENTAJAS

- Grandes latencias en las comunicaciones
- Disponibilidad, seguridad

MIMD con memoria distribuida

CASO PARTICULAR (2)



GRIDS y CLOUDS computacionales

Red global (Internet)

VENTAJAS

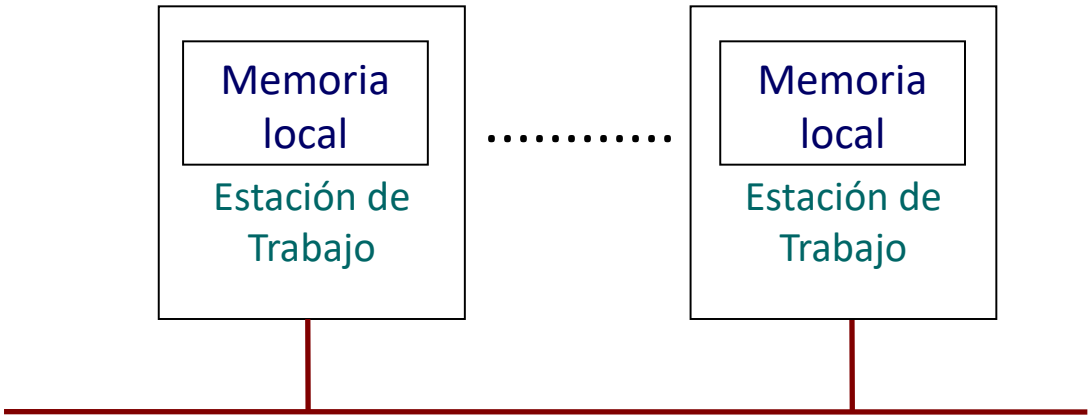
- Permite agregación e integración
- Sistema totalmente escalable
- Existen mecanismos de programación

DESVENTAJAS

- Enormes latencias en las comunicaciones

MIMD con memoria distribuida

CASO PARTICULAR (3)



Agregación de recursos de cómputo de alto desempeño: clusters

Red de datos
(LAN, WAN, Ethernet, FastEthernet, Gigabit Ethernet, FDDI, etc).

VENTAJAS

- Sistema escalable
- Fácilmente programable
- Uso compartido y colaborativo

DESVENTAJAS

- Latencias moderadas en las comunicaciones
- Requiere gestión eficiente para asegurar disponibilidad, seguridad

SMP y MPP

- SMP: Multiprocesamiento simétrico
  - Paralelismo sobre memoria compartida
  - Modelo SIMD y modelo MIMD UMA
  - Primera implementación en 1961
  - Dominante hasta mediados de la década de 1990
  - En 2006 se introdujeron los PCs dual-core

- MPP: procesamiento paralelo masivo
  - Sistema con muchas (decenas, cientos) unidades de procesamiento (ALUs, procesadores)
  - Unidades integradas en una única arquitectura
  - Dominaron el espectro de HPC hasta los inicios del 2000

Evolución tecnológica: clusters

- En la lista Top500 de 2025, 88% de los sistemas más potentes son clusters.
- El 12% son sistemas MPP.
- No hay sistemas de procesador único desde 1996.
- No hay sistemas SIMD desde 1997.
- No hay sistemas SMP desde 2002.
- No hay constelaciones (sistemas de memoria compartida) desde 2008.

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

2.3: CLUSTERS

Basado en el artículo “Cluster computing at a glance”. M. Baker, R. Buyya

¿Cómo mejorar el desempeño?

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

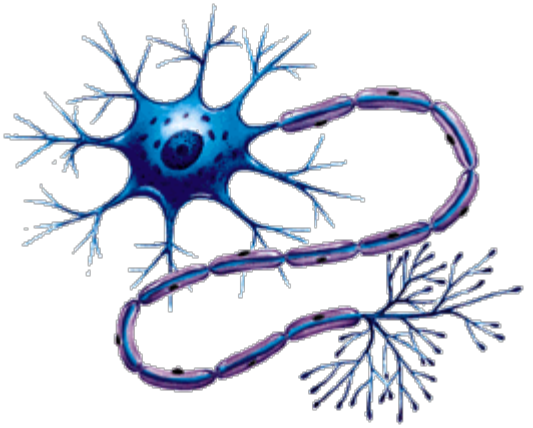
- Básicamente, hay 3 maneras:
  1. Trabajar más intensamente
  2. Trabajar más inteligentemente
  3. Solicitar ayuda
- Analogía para la computación
  1. Utilizar hardware especial, por ejemplo reducir el tiempo por instrucción con un procesador de mayor ciclo de reloj
  2. Optimizar algoritmos y técnicas de programación
  3. Utilizar múltiples recursos de cómputo para resolver el problema, ejecutando más instrucciones en el mismo tiempo

Secuencial vs paralelo

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Limitaciones de las arquitecturas secuenciales
  - Se alcanzaron las limitaciones físicas (velocidad de la luz, termodinámicas, cuánticas)
  - Mejoras de hardware como el pipelining, el procesador superscalar, etc., no son escalables y requieren una complicada tecnología de fabricación e instrumentación
  - El procesamiento vectorial sólo funciona adecuadamente para cierta clase de problemas
- Procesamiento paralelo en la década de 2000
  - La tecnología de paralelismo maduró y fue explotada comercialmente
  - Existía un amplio trabajo de investigación y desarrollo en herramientas y entornos de programación
  - El desarrollo significativo de la tecnología de redes permitió el avance de la computación heterogénea

Clusters: motivación



- Analogía biológica:
  - Procesamiento paralelo en estructuras cerebrales
  - La “velocidad global” con la cual las millones de neuronas del cerebro humano resuelven problemas muy complejos es asombrosa, aún cuando individualmente, el tiempo de respuesta de una neurona es lento (del orden de ms)
  - Este argumento sugiere la potencial utilidad de utilizar múltiples recursos de cómputo funcionando de modo coordinado para resolver un problema global
- Motivación para utilizar clusters:
  - Permiten reutilizar equipamiento disponible (no dedicado)
  - El ancho de banda para comunicaciones entre workstations ha crecido al desarrollarse nuevas tecnologías y protocolos e implementarse en LANs and WANs
  - Los clusters de workstations son más sencillos de integrar en los entornos de desarrollo y producción que las supercomputadoras

Resolución de problemas complejos

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- ¿Se necesitan más recursos de cómputo para resolver problemas complejos?
- Considerando:
  - La gran cantidad de equipos subutilizados.
  - La enorme cantidad de ciclos de procesador libres, a los cuales puede darse un uso práctico.
  - El costo desmedido de un supercomputador.
  - Los recursos de cómputo distribuídos encajan con el modelo de clusters.
- ¿Se necesitan más recursos de cómputo para resolver problemas complejos? ……

SI

- Pero no siempre es necesario HARDWARE ESPECIAL
- Es posible implementar soluciones con el equipamiento disponible (oficina, empresa, centro educativo, centro de datos).

MOTIVACIÓN PARA USAR CLUSTERS

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Estudios muestran que el uso de ciclos de CPU promedio de las estaciones de trabajo es típicamente menor al 10% de su capacidad.
  - Como la performance mejora, el uso (porcentual) de CPU decrece aún más.
- Se hace cada vez más difícil justificar la inversión importante para adquirir un supercomputador y sus herramientas de desarrollo.

- Las herramientas de desarrollo para PCs y workstations están exhaustivamente analizadas y probadas (inclusive algunas estandarizadas).
- Esta situación contrasta con las soluciones propietarias de las supercomputadoras, muchas de ellas no estandarizadas.
- Los clusters de workstations son escalables. Por relativamente poco costo adicional es posible agregar nuevos recursos de cómputo.

Usos de CPU

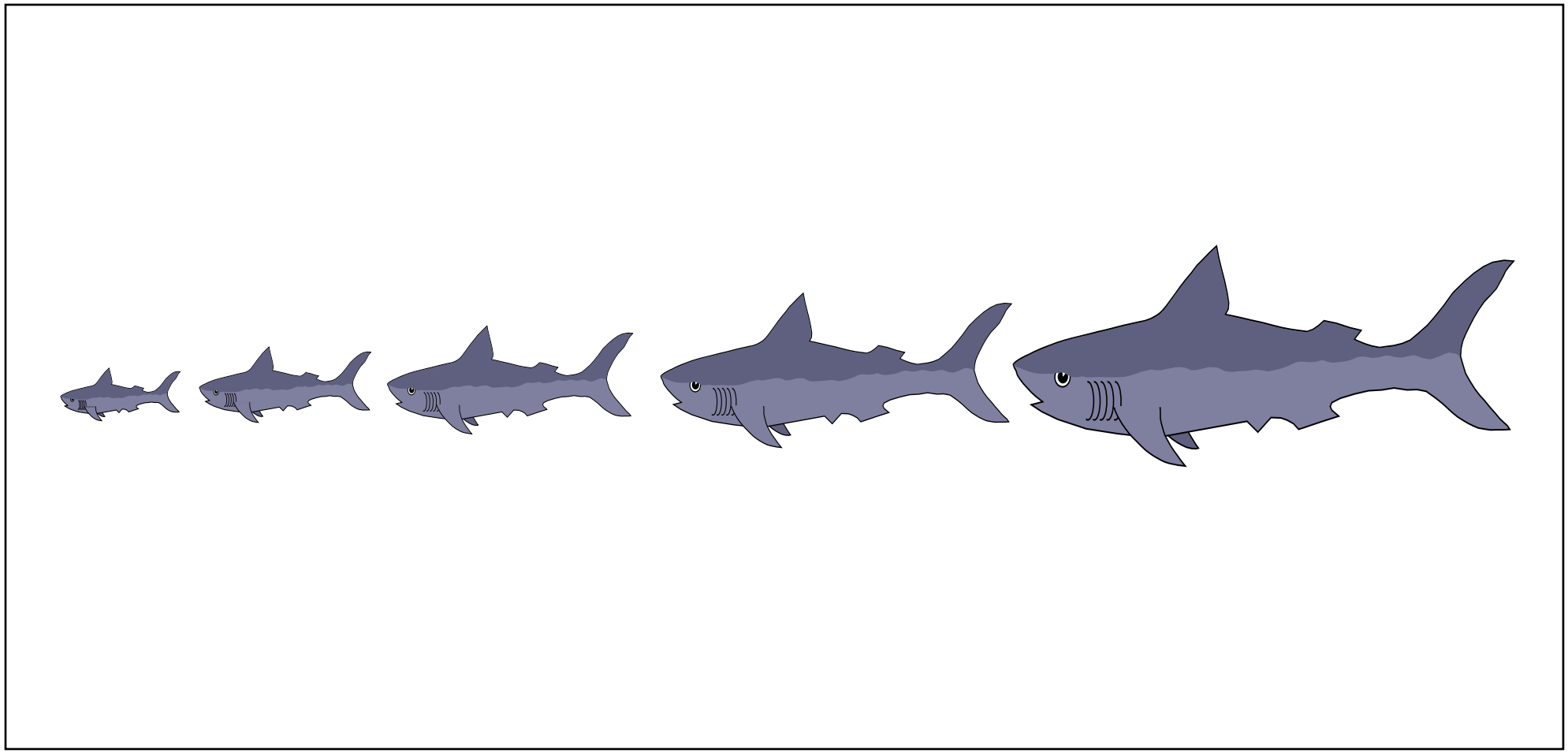
- Usualmente una workstation será propiedad de un indivuduo, grupo, departamento u organización, en el sentido de que estarán DEDICADAS al uso de su(s) propietario(s).
- ¿Cómo implementar un “cluster de workstations” para ejecución de aplicaciones paralelas y distribuídas?
- Típicamente, hay varios tipos de “propietarios”, de acuerdo al uso que dan a “su” CPU.
  1. Estudiantes o trabajadores ocasionales.
  2. Aplicaciones “de oficina” (email, documentos).
  3. Desarrollo de software (edición, compilación, test).
  4. Ejecución de aplicaciones que requieren cómputo intensivo.

El “robo de ciclos” de CPU

- Las técnicas de computación en clusters no dedicados tratan de “robar” ciclos de CPU a los propietarios que realizan tareas “livianas” [(1), (2) y (3)] de manera de proveer a los que ejecutan tareas “pesadas” [(4)].

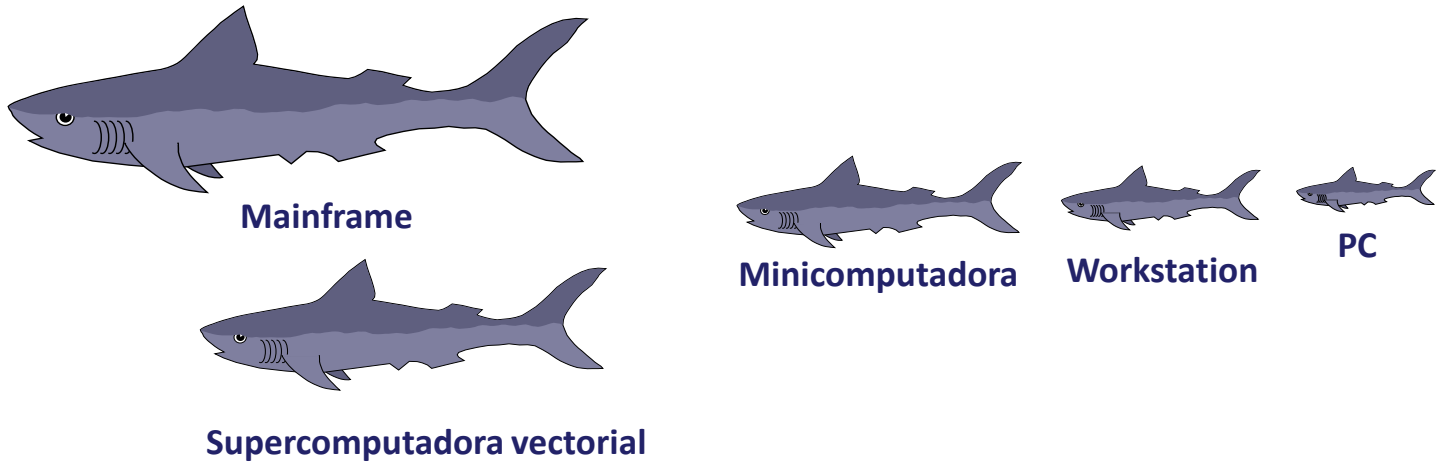
- Sin embargo, esta práctica requiere superar los egoísmos personales (los usuarios son muy protectores de “sus” equipos).
- Usualmente se requiere de autorización institucional para utilizar las computadoras de esta manera.
- El robo de ciclos de CPU realizado fuera de las horas estándar de trabajo es viable.
- El robo de ciclos de CPU realizado en horario laboral, sin impactar el uso interactivo de CPU y memoria es más difícil de implementar.

¿Dónde ubicar a los clusters?



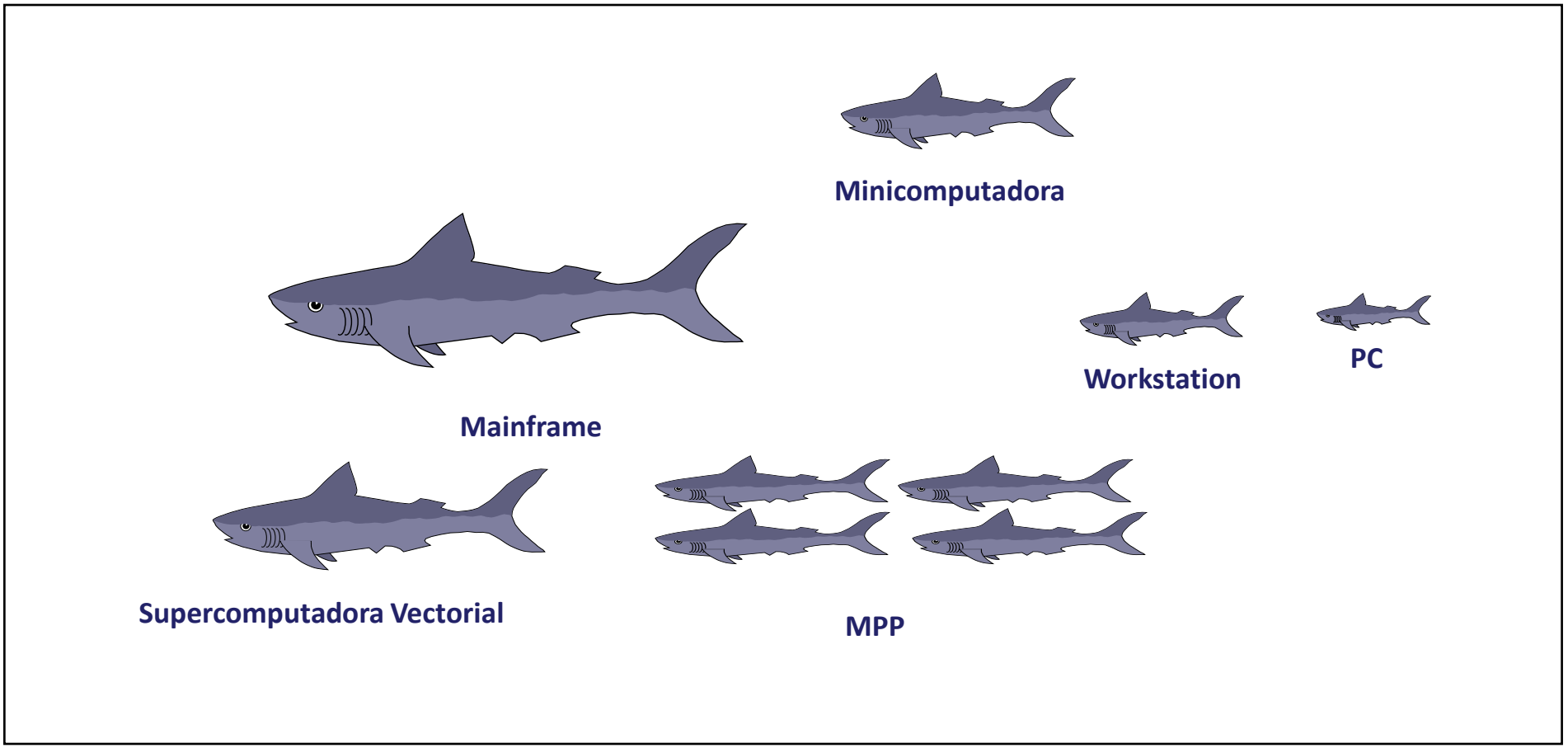
Cadena alimenticia real

¿Dónde ubicar a los clusters?



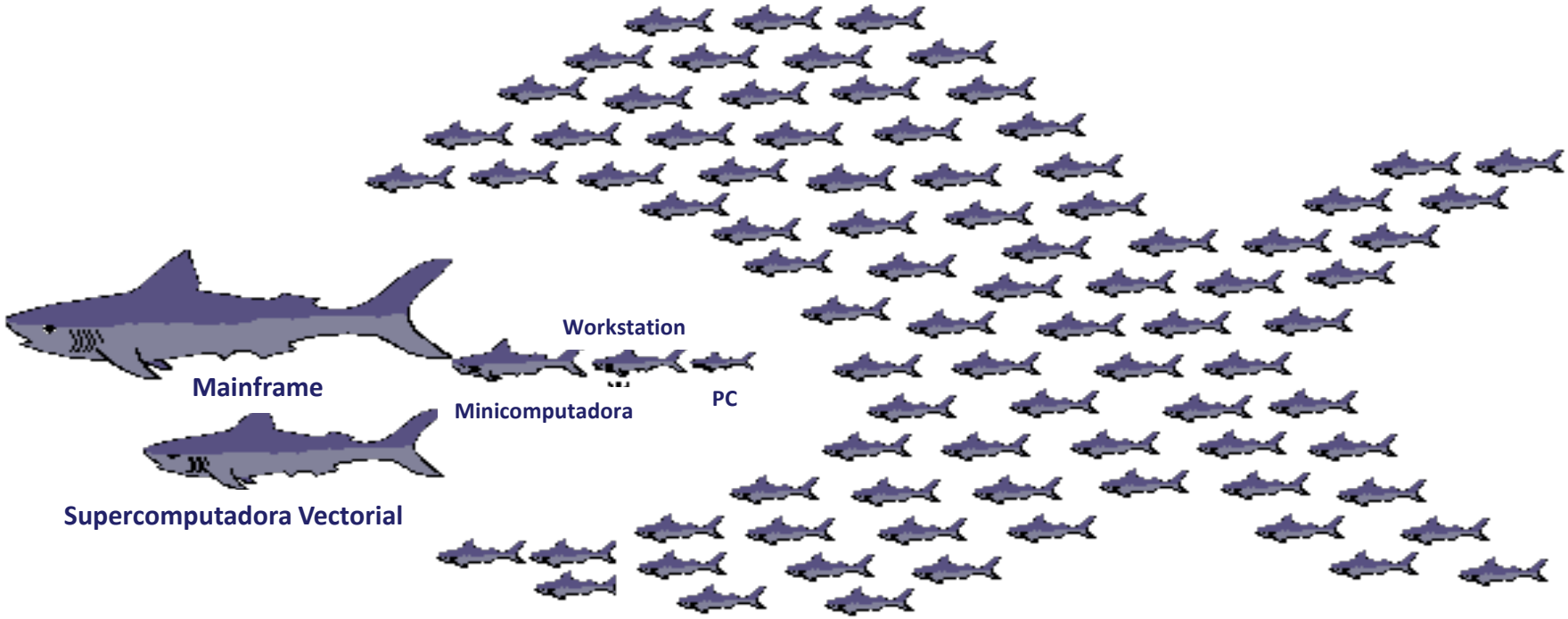
Cadena de las computadoras (década de 1990)

¿Dónde ubicar a los clusters?



Cadena de las computadoras (década de 2000)

¿Dónde ubicar a los clusters?



NOW (Network of Workstations) = CLUSTERS

Cadena de las computadoras (presente y futuro)

Formalmente: ¿qué es un cluster?

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Un cluster consiste en un tipo de sistema de procesamiento paralelo o distribuido, compuesto por un conjunto de computadoras que son capaces de trabajar cooperativamente como un único e integrado recurso de cómputo.

- Características de un típico cluster :
  - Red: rápida, mejor que una típica LAN
  - Protocolos de comunicación de latencia baja
  - Menor conexión que un SMP

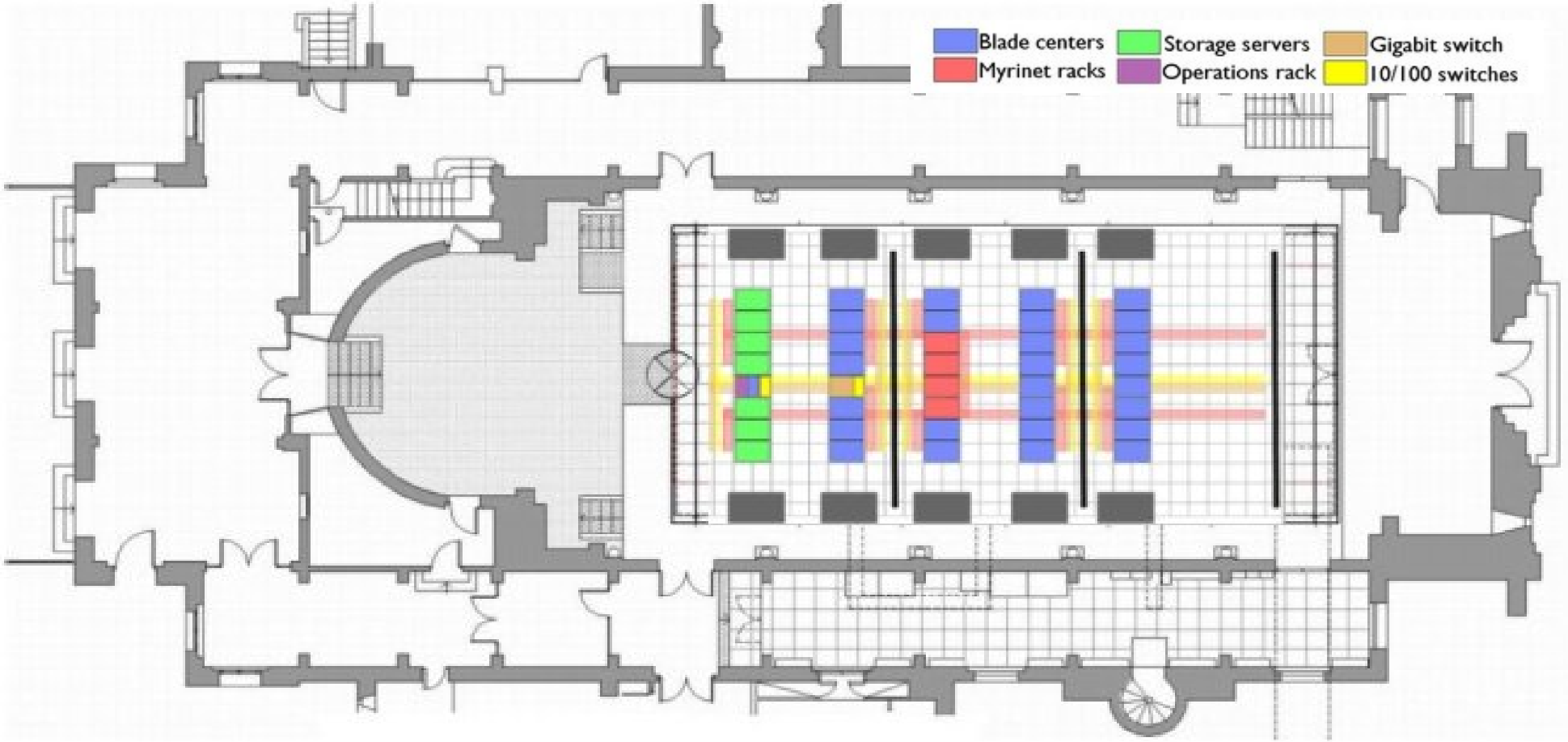
Clusters: motivos de su desarrollo

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- El desempeño de los recursos de cómputo se incrementó
  - Ley de Moore: capacidad de procesamiento se duplica cada 24 (18) meses aproximadamente.
- Las redes de comunicaciones se hicieron cada vez más veloces
  - Aumenta ancho de banda, interfaces simples.
- RAID (Almacenamiento redundante de bajo costo).
  - Alta disponibilidad y escalabilidad.
- Los clusters tienen escalabilidad incremental
  - Desempeño de nodos individuales puede mejorarse con recursos adicionales (memoria, disco).
  - Pueden agregarse nuevos nodos y reemplazar otros.
  - Clusters de clusters (metacomputadoras).
- Herramientas de software completas.
  - Threads, PVM, MPI, C, C++, Java, .NET, Compiladores, Debuggers, SOs, etc.
- Amplia gama de aplicaciones.

Ejemplos de clusters: Marenostrum

- MareNostrum es un cluster de procesadores PowerPC (arquitectura BladeCenter), SO Linux, y con red de interconexión Myrinet



#5 en Top500 (Junio de 2005): 27910 GFlops

Ejemplos de clusters: Marenostrum

- 42.144 Teraflops de rendimiento de pico teórico (42.144×10^{12} = 42 billones de operaciones por segundo).
- 4.800 procesadores PowerPC 970FX en 2400 Nodos duales de 2.2 GHz.
- 9.6 TB de memoria.
- 236 TB de almacenamiento en disco.
- 3 redes de interconexión:
  - Myrinet.
  - Gigabit Ethernet.
  - Ethernet 10/100.

Ejemplos de clusters: Thunder

#7 Top500 (Junio 2005): 19940 GFlops

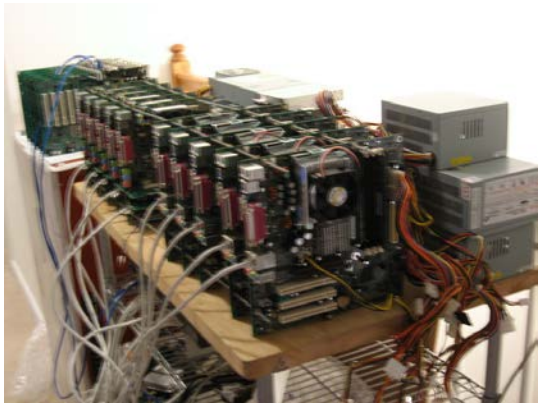
- 1024 Nodos
- 4 CPU/Nodo
- Itanium 2 1.4 Ghz CPU
- 8GB RAM/Nodo
- Discos: 75GB/Nodo
- SO: CHAOS

Millenium PC clumps

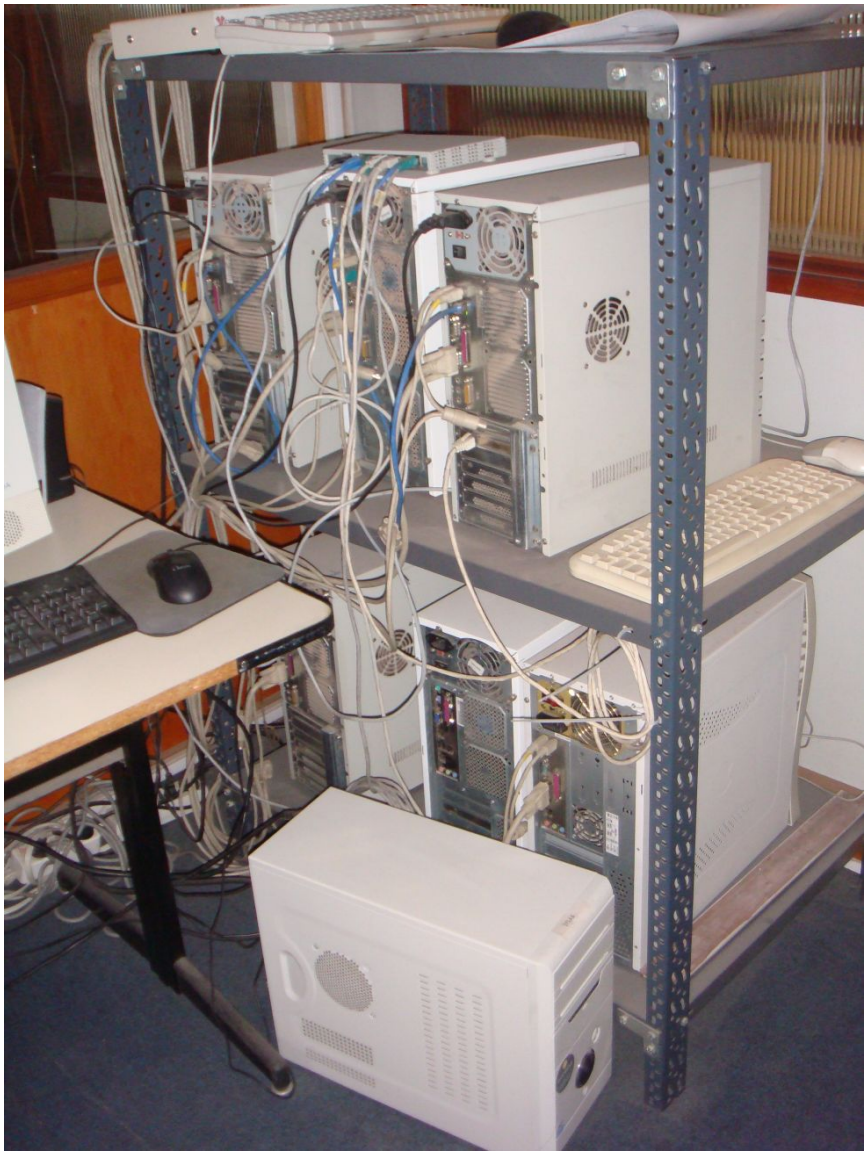
- Clusters baratos, fáciles de administrar
- Replicados en varios departamentos
- Prototipo para grandes clusters de PCs y servidores

Clusters Beowulf

- Sterling y Becker (NASA), 1994
- Computadores comerciales, idénticos, que ejecutan software libre, sistema operativo tipo Unix (BSD, Linux, Solaris)
- Conectados por TCP/IP sobre LAN
- Tienen instalado software que permiten el procesamiento cooperativo
  - PVM, MPI, OSCAR



El cluster rock



- Centro de Cálculo, Facultad de Ingeniería
- Primer cluster operativo de la Facultad (desde inicios de los 2000)
- Hasta 8 PCs conectados

El cluster FING

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Infraestructura para computación científica de alto desempeño, UdelaR.
- Operacional desde marzo de 2009
  - Autofinanciada
  - Autogestionada
- 1740 cores (540 de CPU, 960 de GPU, 240 Xeon Phi)
  - >1 TB de memoria RAM, >250 TB RAID storage, 34 kVA batería
  - Pico de performance: 6000 GFLOPS (6×10^{12} operaciones de punto flotante por segundo), *el mayor poder de cómputo disponible en el país*

Cluster FING: >11.000.000 hs de cómputo efectivo (2018)
http://www.fing.edu.uy/cluster

Centro Nacional de Supercomputación (Cluster UY)

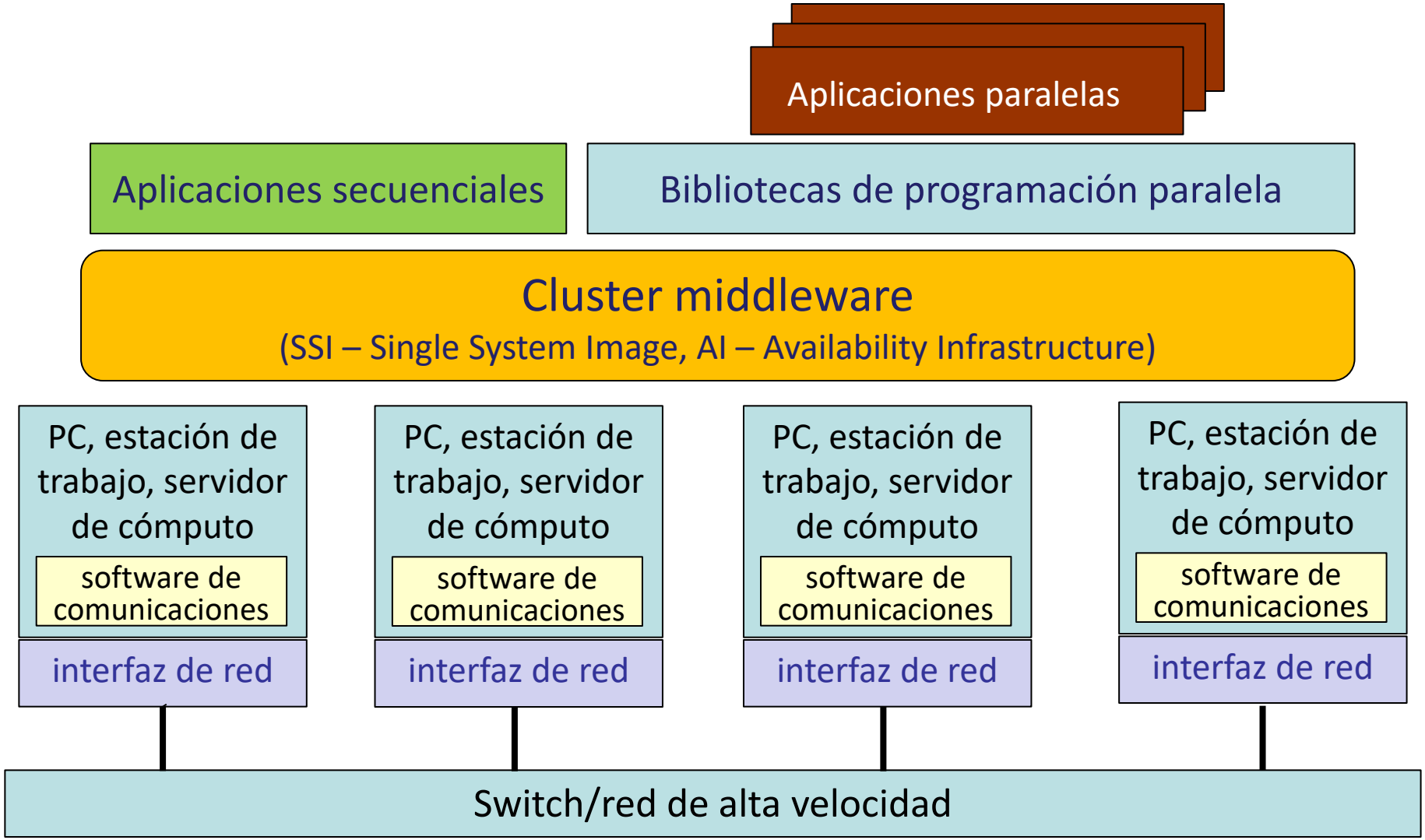
UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Proyecto de grandes equipos (ANII-UdelaR, 2017)
  - Autofinanciado y autogestionado
- +30 nodos interconectados: más de 1500 núcleos de cómputo de CPU y 125.000 núcleos de cómputo de GPU de última generación
- Capacidad de procesamiento estimada de 100 TeraFlops (100 billones de operaciones por segundo)
- 20× Cluster FING

El mayor poder de cómputo disponible en el país

http://cluster.uy/

Arquitectura de un cluster



Componentes de un cluster

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

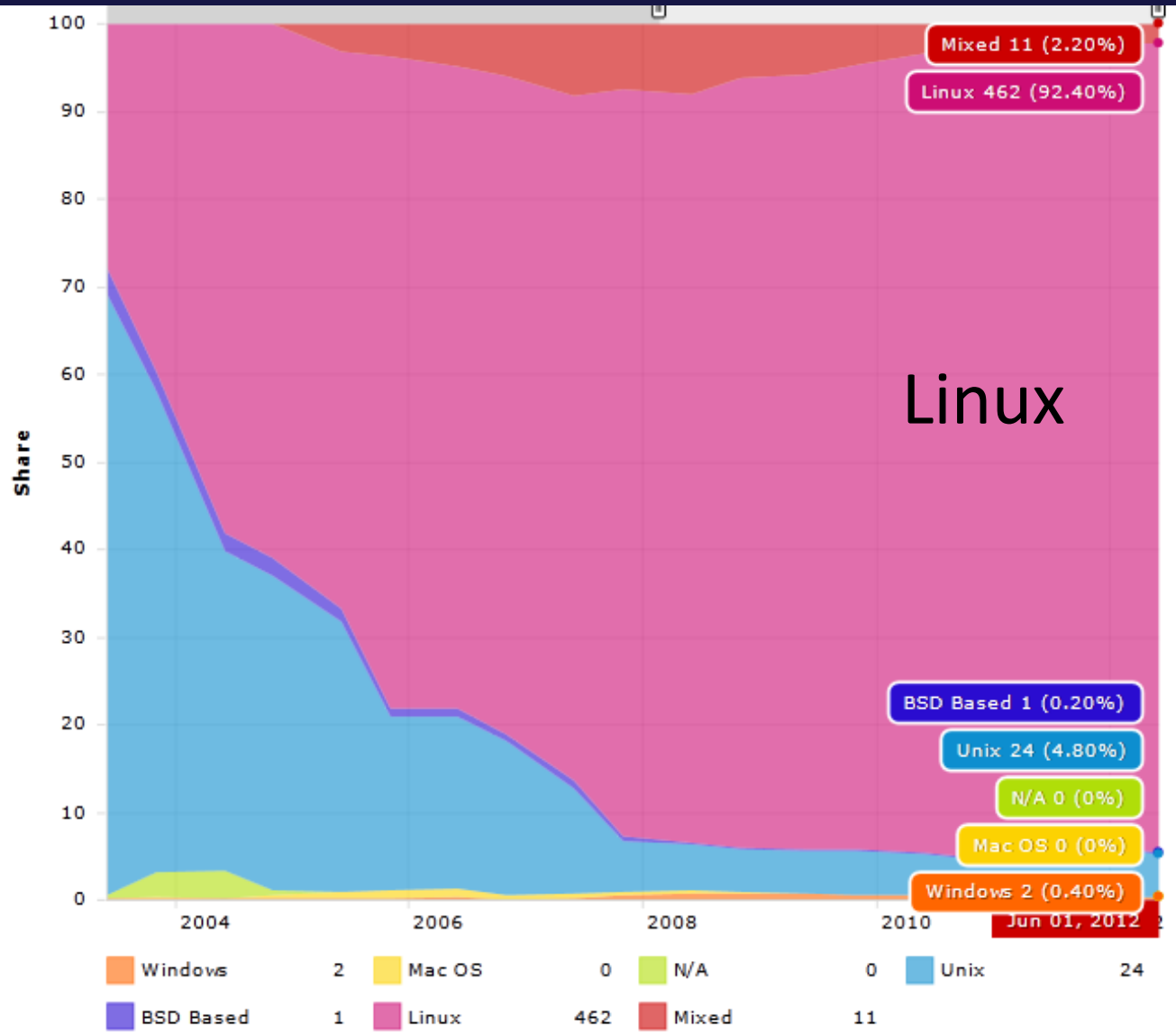
- Los componentes de un cluster incluyen:
  1. Nodos
     - Elementos de cómputo y almacenamiento de datos
  2. Software de base
     - Sistemas operativos
  3. Comunicaciones
     - Redes de alta velocidad
     - Interfaces y software para comunicaciones
  4. Middleware
  5. Entornos de programación
     - Bibliotecas y herramientas de desarrollo

Clusters: nodos y procesadores

- Nodos: múltiples componentes de alta performance
  - PCs y workstations, servers, GPUs, coprocesadores
  - Sistemas de HPC distribuidos, que conducen a la “metacomputación”
- Los componentes pueden ser de diferentes arquitecturas, y ejecutar con diferentes sistemas operativos (clusters heterogéneos)
- Procesadores de varios tipos: CISC/RISC/VLIW/Vectoriales
  - Intel: Xeon (Clovertown, Harpertown, Nehalem, Westmere)
  - AMD Opteron SixCore, QuadCore, DualCore
  - IBM Power6, PowerXCell
  - Otros: Sun SPARC y ULTRASPARC, SGI MPIS, Digital Alpha, etc.
- También se han utilizados procesadores que integran memoria, procesador y red en un único chip
  - IRAM (CPU y memoria, http://iram.cs.berkeley.edu)
  - Alpha 21366 (CPU, Controlador de memoria, NI)

Clusters: sistemas operativos

- Estado del arte según sistemas operativos:
  - Linux (desde los clusters Beowulf)
  - Desde noviembre de 2017, todos los sistemas del Top500 usan Linux



Clusters: comunicaciones

- Redes de alta velocidad
  - Ethernet (10Mb/s), Fast Ethernet (100Mb/s)
  - Gigabit Ethernet (1Gb/s), 10 Gigabit Ethernet
  - SCI (Dolphin-MPI latencia: 12 microsegundos)
  - ATM
  - Myrinet (1.2 Gb/s), Myrinet 2000 (2 Gb/s) y Myri-10G (10 Gb/s), altamente escalable
  - Infiniband (10 Gb/s, enlaces añadidos permiten alcanzar 100 Gb/s)
  - Digital Memory Channel, FDDI, etc.
- Tarjetas de red
  - Myrinet tiene NIC (Network Interface Card)
  - Soporte de acceso a nivel de usuario
  - Existen procesadores que integran controlador de memoria e interfaz de red en un único chip

Clusters: software de comunicaciones

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Facilidades estándar de IPC de los sistemas operativos
  - Sockets (TCP/IP), pipes, etc.
- Protocolos a nivel de usuario
  - Active Messages (Berkeley)
  - Fast Messages (Illinois)
  - U-net (Cornell)
  - XTP (Virginia)
- Sistemas que pueden ser construidos sobre los protocolos de base

Clusters: middleware

- Reside entre el SO y las aplicaciones, ofrece infraestructura para soportar:
  - Single System Image (SSI)
  - System Availability (SA)

- SSI permite que un cluster pueda ser visto como un único equipo ("globaliza" los recursos de un sistema)
  - Accceso a través de ssh cluster.myinstitute.edu, ssh cluster.uy
  - Monitoreo centralizado, sistema de archivos global

- SA provee disponibilidad
  - Check pointing y migración de procesos

Clusters: entornos de programación

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Threads (PCs, SMPs, …)
  - POSIX Threads
  - Java Threads
- Bibliotecas de desarrollo de programas paralelos y distribuidos
  - PVM (Parallel Virtual Machine)
  - MPI (Message Passing Interface)
- Software DSMs (Shmem)
- Colas de mensajes
- Otras tecnologías: Java RMI, .NET, etc.

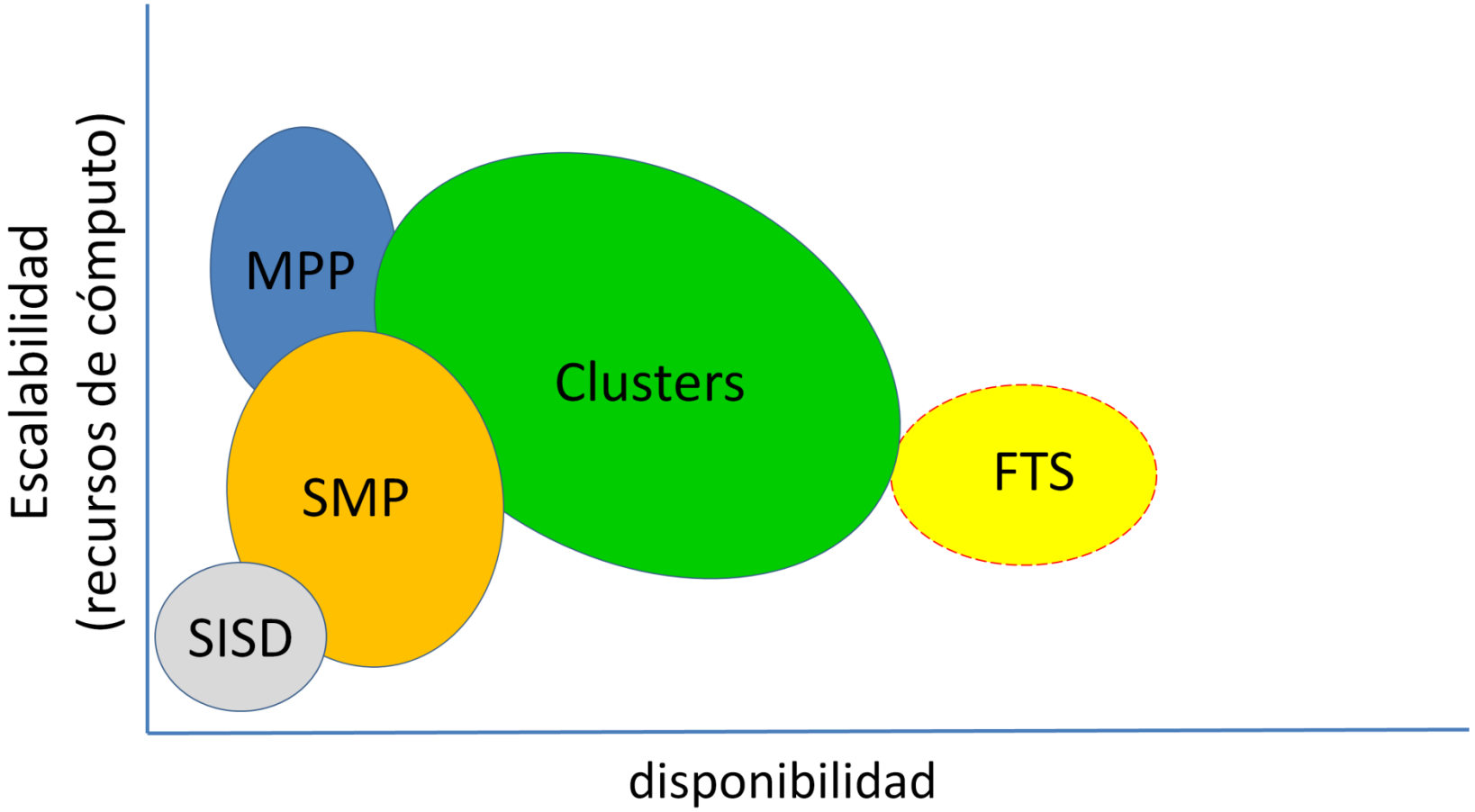
Clusters: otras herramientas

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- Compiladores
  - C/C++
  - Java
  - FORTRAN
- RAD (rapid application development tools)
  - Herramientas basadas en GUI para modelos de programación paralela.
- Debuggers
- Herramientas de análisis de performance
- Herramientas de visualización

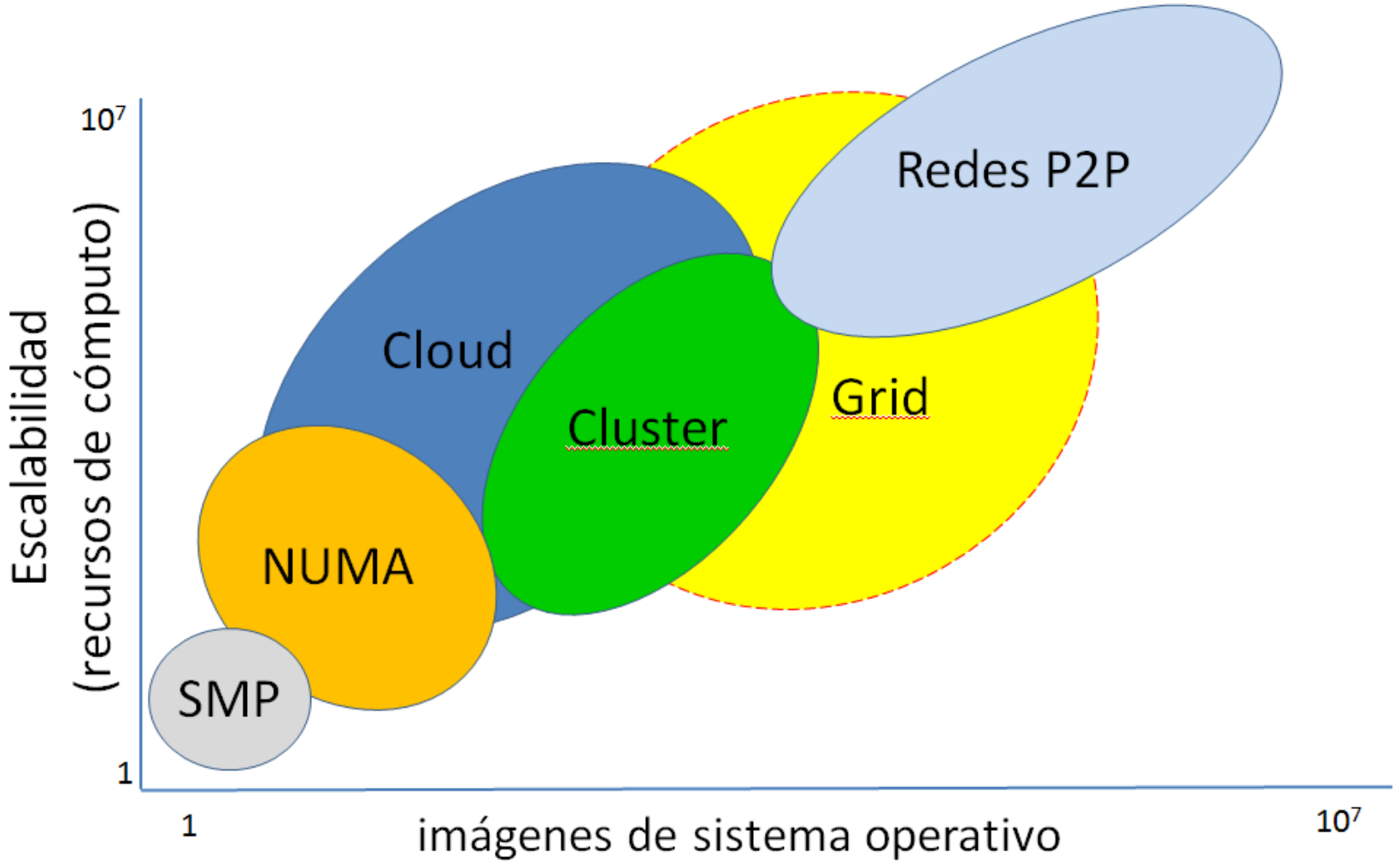
Clusters: eficiencia y robustez

- Escalabilidad versus disponibilidad



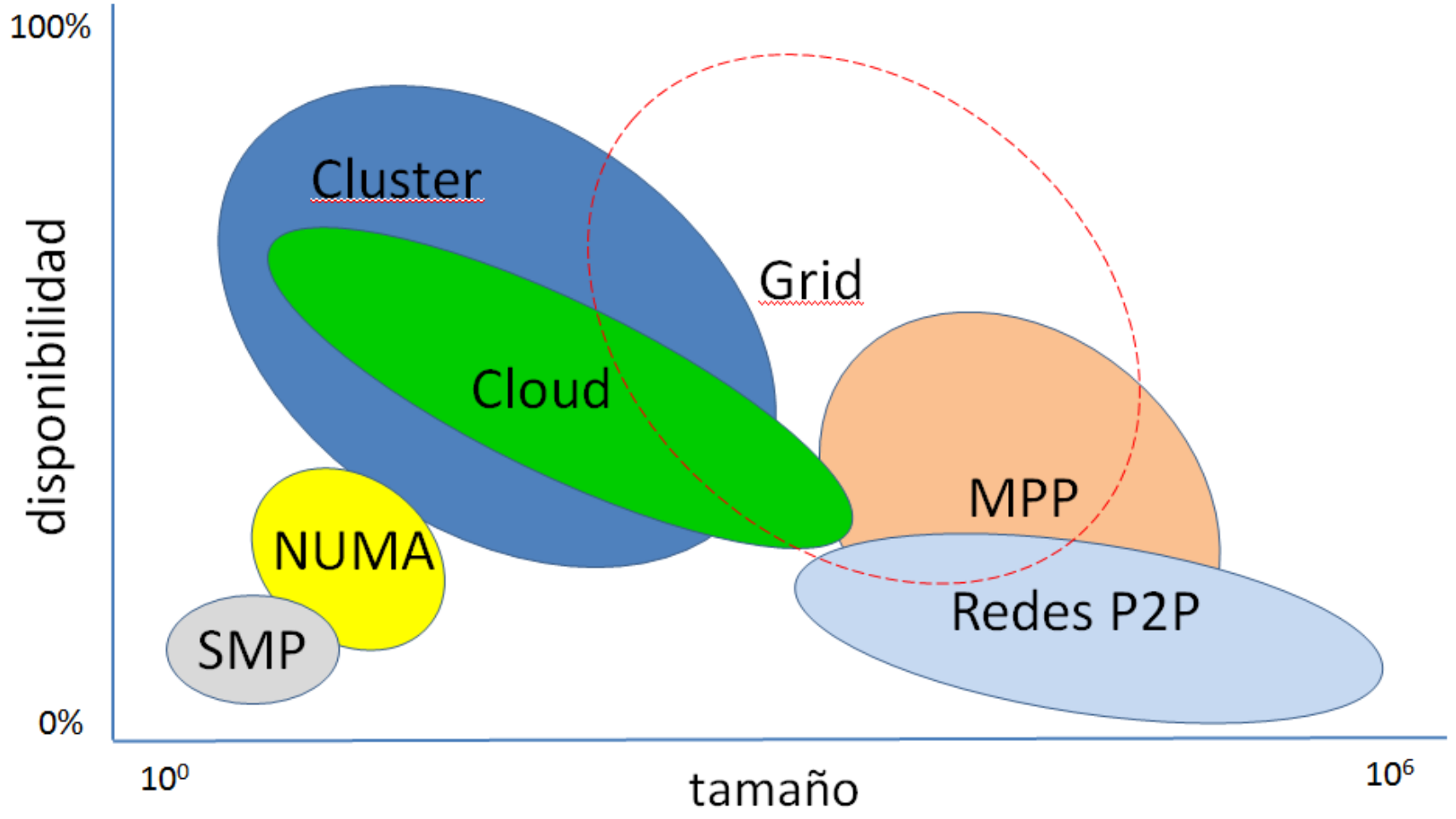
Clusters: escalabilidad

- Escalabilidad versus imágenes del sistema operativo



Clusters: disponibilidad

- Disponibilidad versus tamaño



Clusters: conclusiones

- Clusters
  - Insfraestructura promisoria para contemplar necesidades importantes de cómputo en un entorno de recursos limitados

- Principales ventajas:
  - Relación costo/performance
  - Escalabilidad incremental
  - Sistema “multipropósito” (no dedicado)

SERÁ LA PRINCIPAL PLATAFORMA DE TRABAJO A UTILIZAR EN EL CURSO

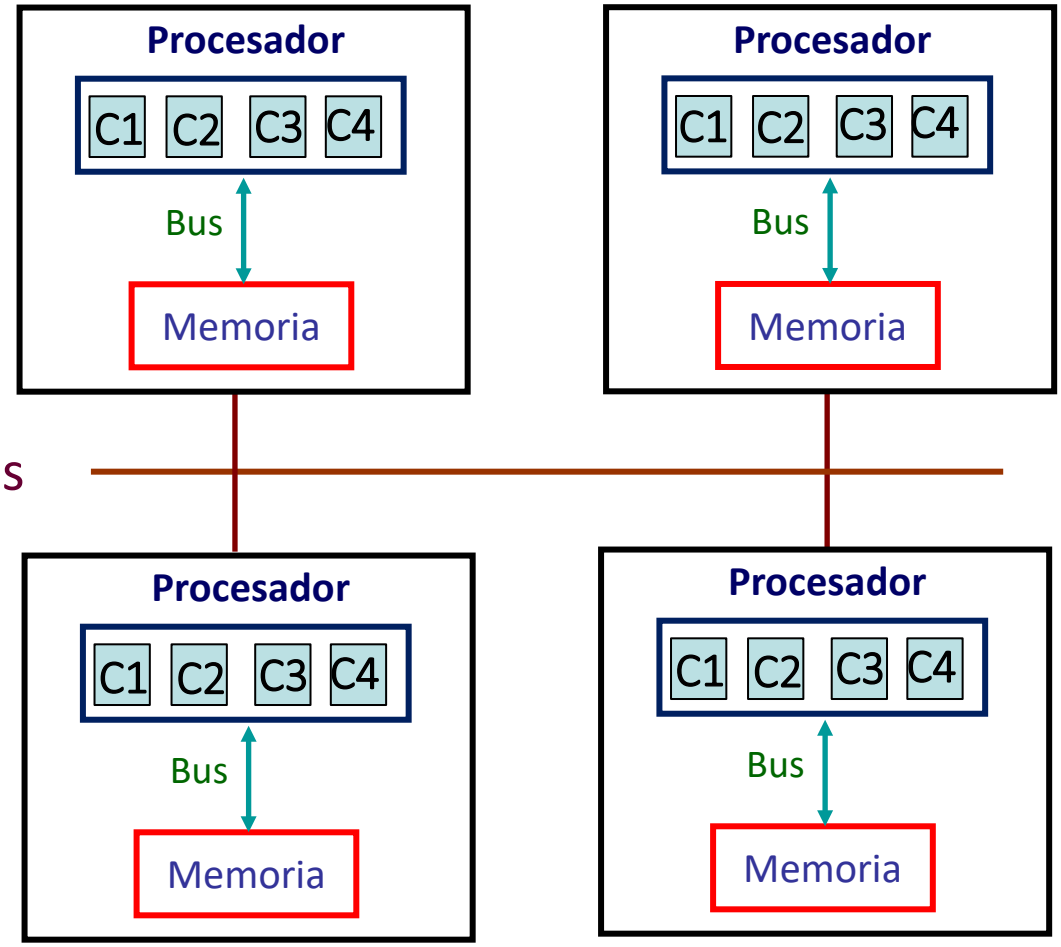
Clusters: MIMD híbridos

Combina memoria compartida y memoria distribuida

Los cores de cada procesador comparten memoria

Los procesadores tienen memoria distribuida

Canal de comunicaciones



UNIVERSIDAD DE LA REPÚBLICA URUGUAY

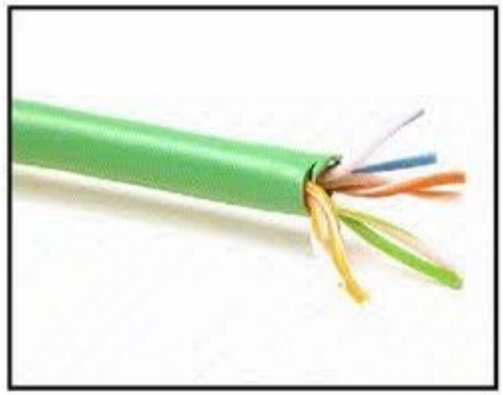
2.3: FACTORES QUE DETERMINAN LA EFICIENCIA

Conectividad entre procesadores

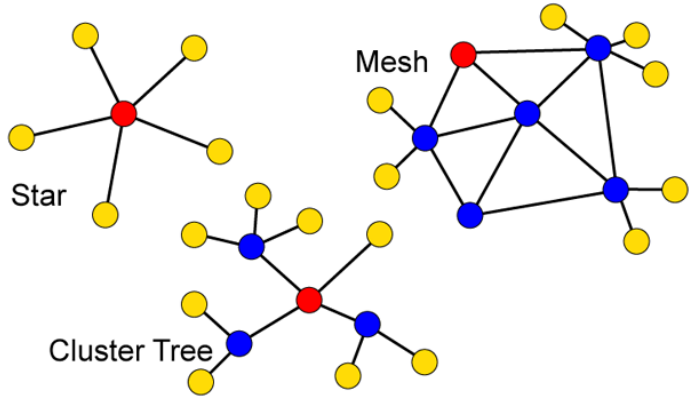
- ESTÁTICA
  - Caminos prefijados entre procesadores.
  - Usualmente canales de comunicación P2P entre procesadores.

- DINÁMICA
  - Los caminos se determinan dinámicamente.
  - Se implementa a través de switches.
    - Simplifica la programación al evitar problemas de comunicación.
    - Garantiza igualdad de latencia para comunicaciones entre distintos procesadores a una distancia fija.
    - Permite comunicaciones “all to all”.
    - Generan topologías fácilmente escalables.

Factores que determinan la eficiencia



- Ancho de banda
  - Número de bits capaces de transmitirse por unidad de tiempo
- Latencia de la red
  - Tiempo que toma a un mensaje transmitirse a través de la red
- Latencia de las comunicaciones
  - Incluye tiempos de trabajo del software y retardo de la interfaz
- Latencia del mensaje
  - Tiempo que toma enviar un mensaje de longitud cero
- Valencia de un nodo
  - Número de canales convergentes a un nodo
- Diámetro de la red
  - Número mínimo de saltos entre los nodos más alejados
  - Permite calcular el peor caso de retardo de un mensaje
- Largo máximo de un tramo de comunicación.



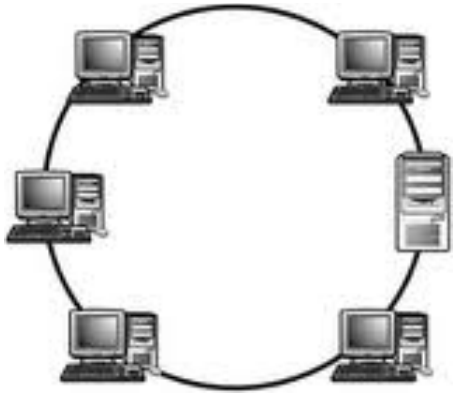
Factores que determinan la eficiencia

- Ancho de bisección
  - Número mínimo de enlaces que en caso de no existir la red se separaría en dos componentes conexas
- Costo
  - Cantidad de enlaces de comunicación

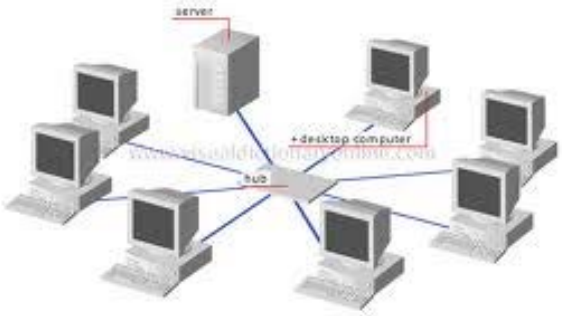
- CONFIGURACIÓN ÓPTIMA
  - Ancho de banda grande
  - Latencias (de red, comunicación y mensaje) bajas
  - Diámetro de la red reducido
  - Ancho de bisección grande
  - Valencia constante e independiente del tamaño de la red
  - Largo máximo de tramo reducido, constante e independiente del tamaño de la red
  - Costo mínimo

Conectividad entre procesadores

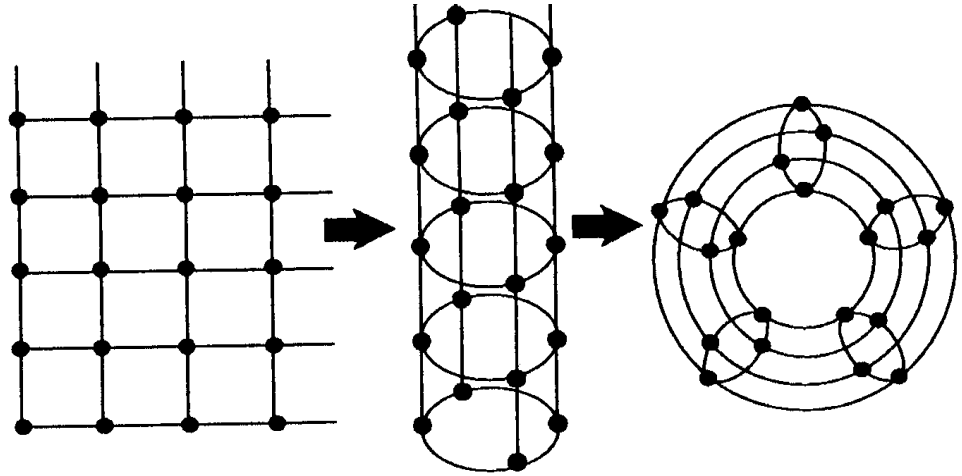
Modelos



Anillo



Estrella

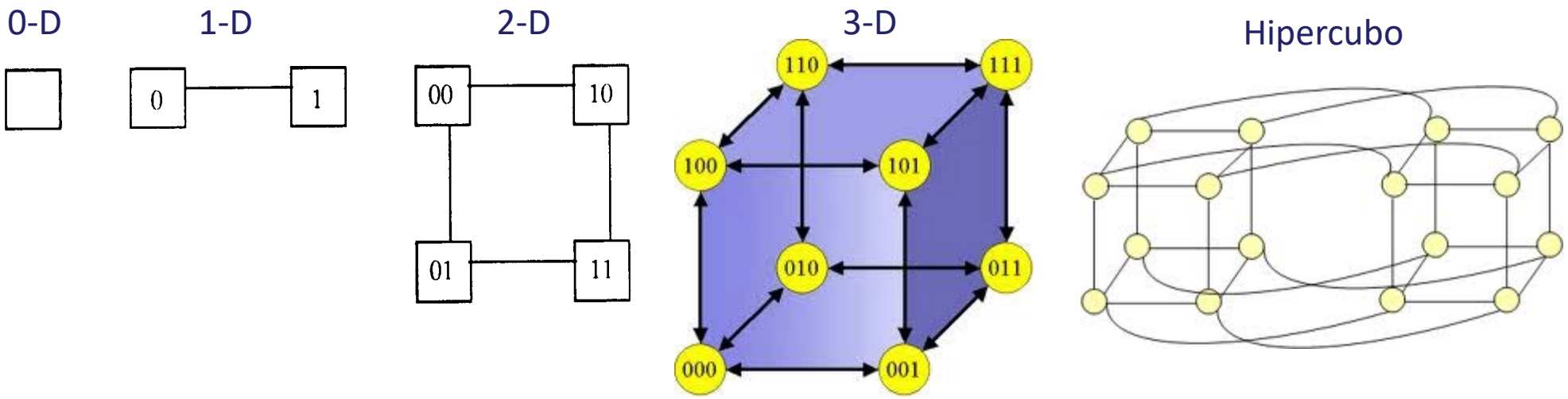


Grilla (2D) Cilindro Toro

Topologías geométricas (mallas)

Conectividad entre procesadores

- Modelos de conectividad



Topologías “dimensionales” (distancia constante)

2.4: ARQUITECTURAS MULTINÚCLEO

Tecnologías de procesadores

- Utilizando los avances tecnológicos, hasta la década de 2000 los fabricantes de procesadores prácticamente duplicaban la cantidad de transistores en un chip de 18 a 24 meses.
- Este ritmo se mantuvo hasta llegar a los límites físicos permitidos.
- El desarrollo de transistores se movió de un tamaño de 90nm a 65nm, lo que permite tener mas transistores en un chip.
- Sin embargo, existen reportes que pronostican que una vez que se alcance los 16nm en tamaño, el proceso no podrá controlar el flujo de los electrones a medida que el flujo se mueva a través de los transistores.
- De esa forma, llegará un momento que los chips no podrán ser más pequeños.
- Asimismo, al reducir su tamaño e incrementar su densidad, los chips generan mayor calor, causando errores en el procesamiento.

Multinúcleos (multicore)

- La tecnología de procesadores multinúcleo constituye una alternativa para mejorar la performance a pesar de las limitaciones físicas.
- Sin duda, los sistemas multinúcleo proponen mayores desafíos en cuanto al desarrollo de sistemas ya que se debe tener en cuenta que en el micro-tiempo se ejecuta más de una instrucción en el mismo equipamiento.
- Sin embargo, un buen uso de la tecnología puede implicar un beneficio importante en el poder de procesamiento.

Intel Core 2 Duo E6750

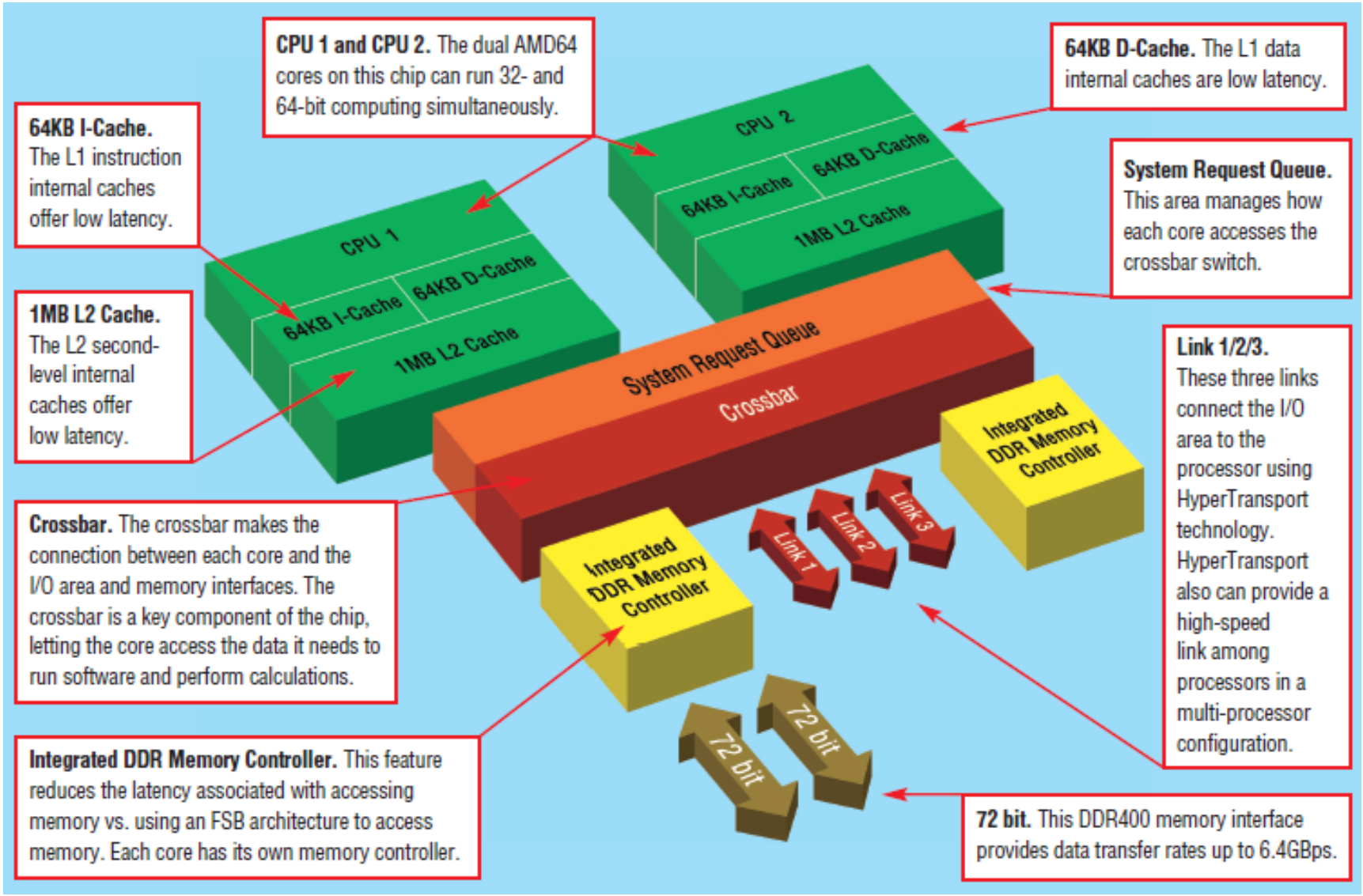
AMD Athlon X2 6400+

Primeras tecnologías AMD multinúcleo

UNIVERSIDAD DE LA REPÚBLICA URUGUAY

- En 2005, AMD lanzó la línea de procesadores Opteron con una tecnología que denominó Direct Connect Arquitecture, la cual integraba el controlador de memoria en el mismo chip del procesador.
- AMD también dispuso de una memoria cache de segundo nivel (L2 cache) independiente para cada procesador.
- Para interconectar los dos procesadores se presentó un crossbar switch de alto desempeño que permitía el acceso cruzado a la memoria.
- La interconexión con los dispositivos de entrada/salida se realizaba a través de la tecnología HyperTransport.

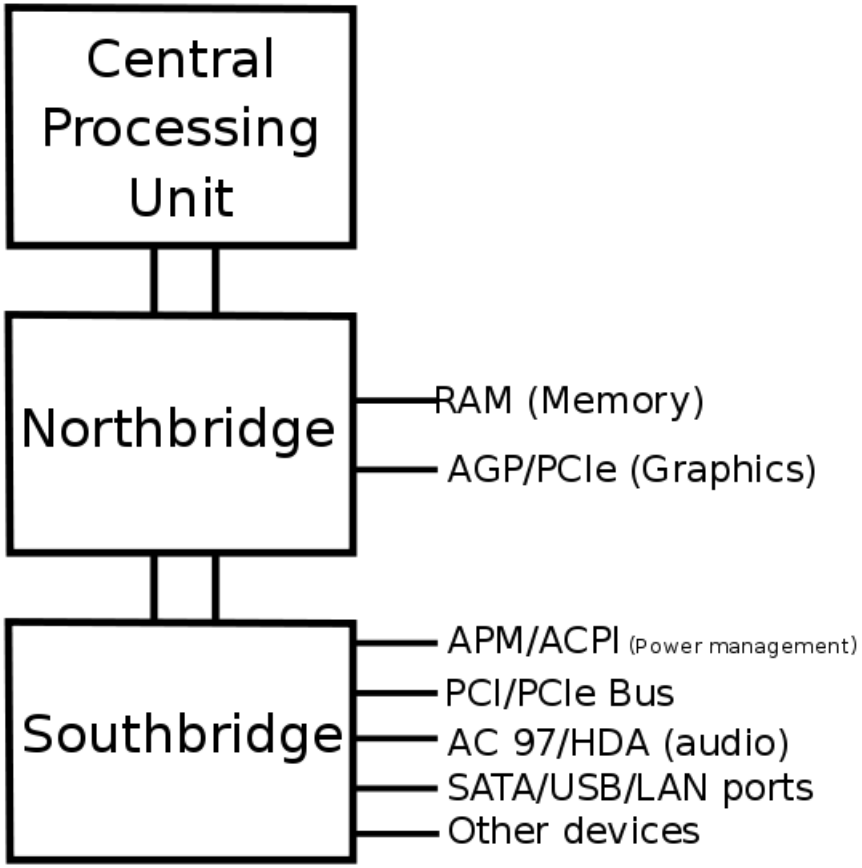
AMD dual-core

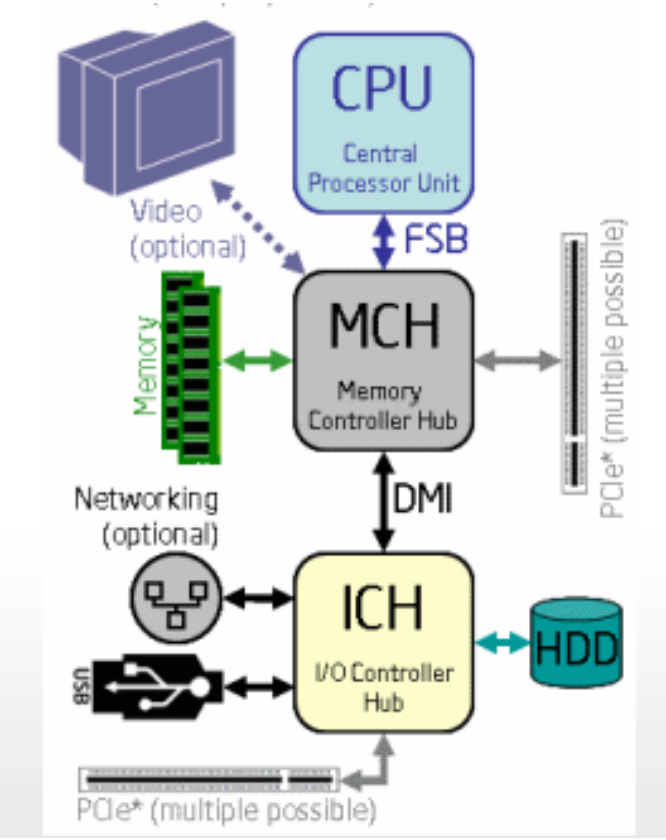


Primeras tecnologías intel multinúcleo

- Los procesadores Intel mantuvieron la interconexión a través de un chipset compuesto por el NorthBridge y el SouthBridge.
- El NorthBridge contiene el controlador de acceso a la memoria RAM.
- El SouthBridge permite la interconexión con dispositivos de entrada/salida.
- El bus de interconexión entre los procesadores entre el procesador y el controlador de memoria es denominado FSB (Front Side Bus).
- De esta forma, los procesadores Intel mantuvieron el sistema UMA.
- En este caso, a diferencia de lo propuesto por AMD, los núcleos compartían la memoria cache de segundo nivel (L2 cache).

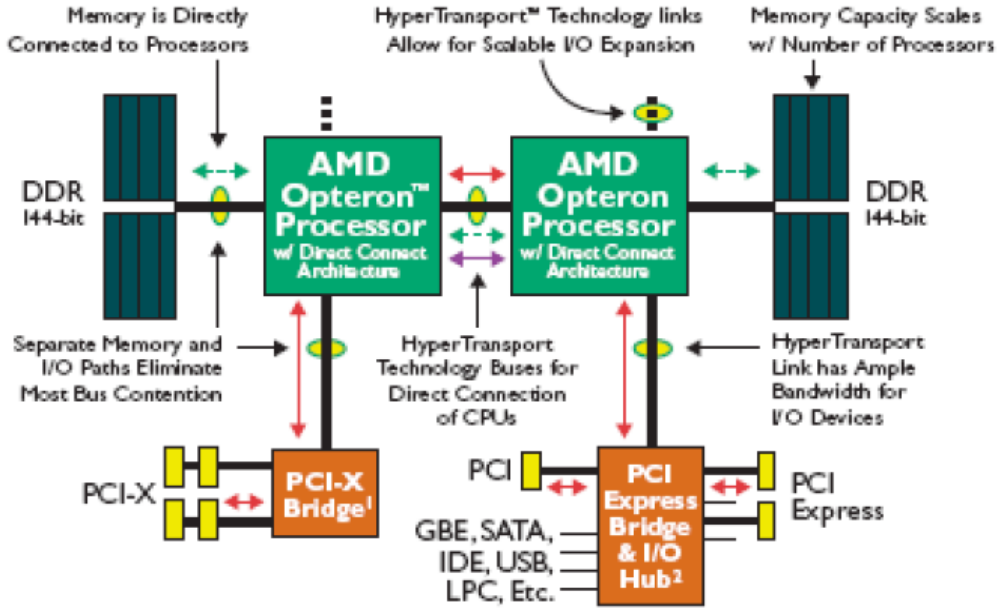
Intel north-south bridge

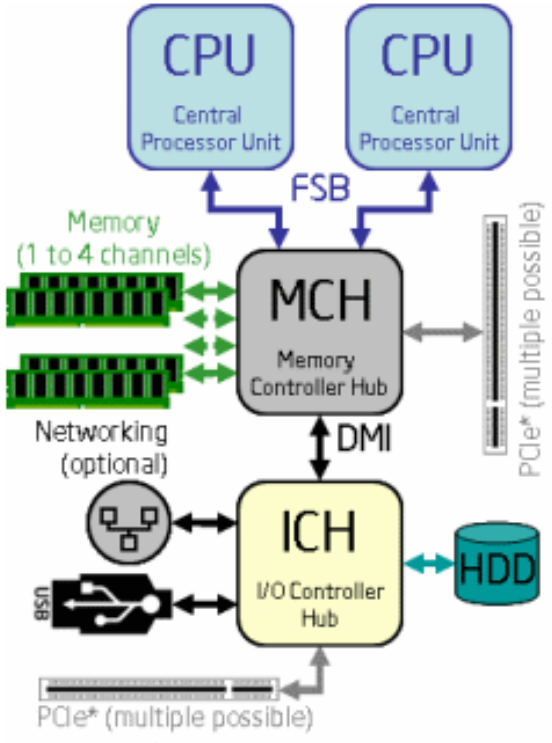




Combinando procesadores multinúcleo

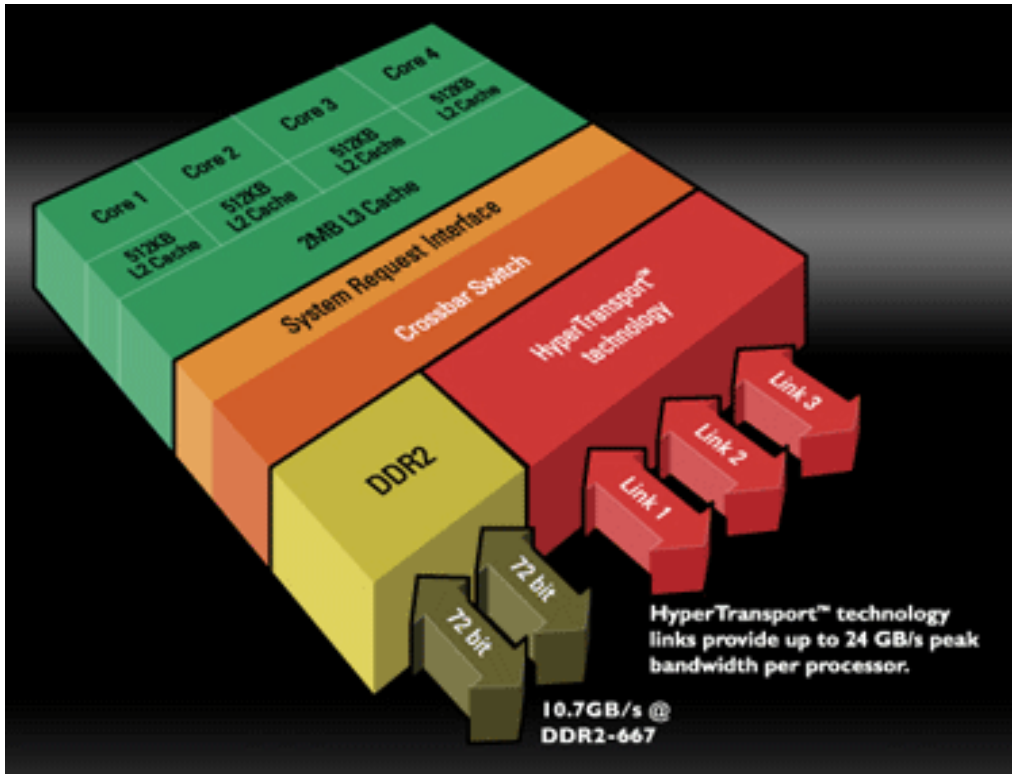
- A medida que la tecnología avanzó se incorporaron más procesadores por equipo, logrando combinar multiprocesadores con multinúcleo.
- La tecnología pasó a ser de un sistema NUMA.
- Los procesadores tienen acceso a toda la memoria (global), pero acceden a la memoria con diferentes velocidades.





Tecnologías quad-core

- El siguiente paso fue abordar los procesadores quad-core.
- En 2008 AMD propuso el chip denominado Barcelona que incluyó una memoria cache de segundo nivel (más reducida) particular de cada núcleo y una gran memoria cache de tercer nivel compartida por los cuatro núcleos.



Tecnologías quad-core

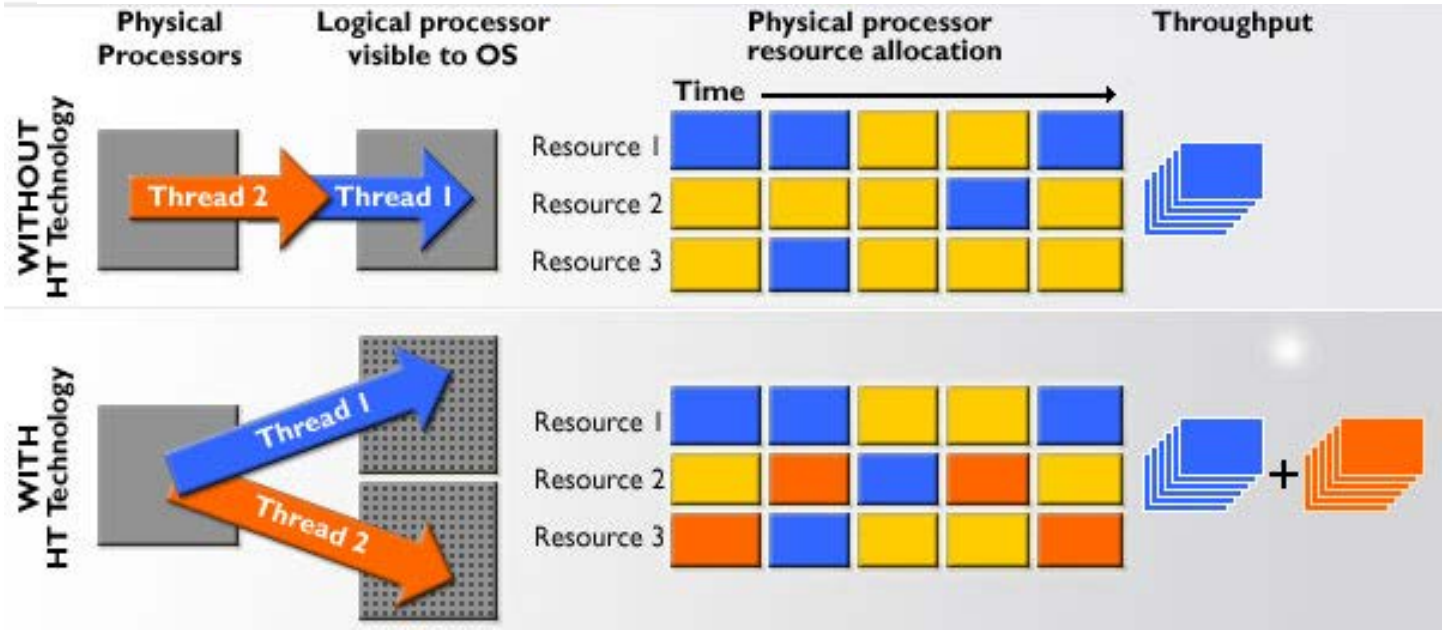
- Otro gran avance fue la inclusión de nuevas instrucciones (SSE128) en la arquitectura que permitieron operaciones de 128 bits (AMD, 2007/2008)
  - Estas operaciones permiten 4 operaciones de punto flotante de doble precisión por ciclo de reloj
  - Dos operaciones hechas a través de la multiplicación y dos a través de la suma
- Por otro lado, Intel siguió con su arquitectura de FSB pero con una cache de segundo nivel compartida por dos procesadores y de mayor tamaño
- A su vez, propuso el movimiento a procesadores de tamaño de 45nm
- Recién a partir de la familia de procesadores Nehalem (2010), Intel desecho el FSB para pasar la controladora de memoria en el procesador
  - Se incorporó una tecnología similar a la de AMD, pasando a un sistema NUMA
- Intel adoptó la tecnología de interconexión Quick-Path Interconnect (QPI) para la comunicación entre los procesadores y también para el acceso al chipset de entrada/salida

Tecnologías multicore

- Los principales procesadores multicore (de propósito general) son:
  - AMD:
    - Opteron 16 cores (2012), Bulldozer 16 cores (2017)
    - Ryzen 64 cores (2017)
    - Epyc Siena 64 cores (2018), Epyc Turin 128/192 cores (2024),
  - IBM:
    - POWER9 24 cores (2017)
    - z14 10 cores (2017)
  - Intel:
    - Atom 16 cores (2015)
    - Core i3/i5/i7/i9 hasta 18 cores (2017–2019)
    - Xeon Platinum hasta 56 cores (2021)
    - Xeon Phi hasta 72 cores (2016)
  - Sun:
    - SPARC T5 16 cores (2015)

Hyperthreading

- Implementación propietaria de Intel para multithreading; introducida en 2002 con los procesadores Xeon (servidor) y Pentium 4 (desktop)
- Consiste en hacer que cada *núcleo físico* sea considerado por el sistema operativo como 2 *núcleos virtuales* independientes
- El sistema operativo puede despachar más de un proceso o hilo simultáneamente y los recursos del núcleo físico son compartidos entre los núcleos virtuales



Multi-core y multi-threads

- Las arquitecturas multinúcleo son útiles y eficientes para implementar programas multi-threads
- Los threads (o hilos) son las unidades de procesamiento
  - Múltiples threads de un proceso son capaces de compartir estado e información (memoria y otros recursos)
  - Los threads comparten el espacio de direccionamiento (variables)
  - Los threads son capaces de comunicarse sin utilizar mecanismos explícitos de IPC
  - El cambio de contexto entre threads es más veloz que entre procesos

La programación multithreading será estudiada en el curso