POLI
ESCOLA POLITÉCNICA
DE PERNAMBUCO

Organização e arquitetura de computadores

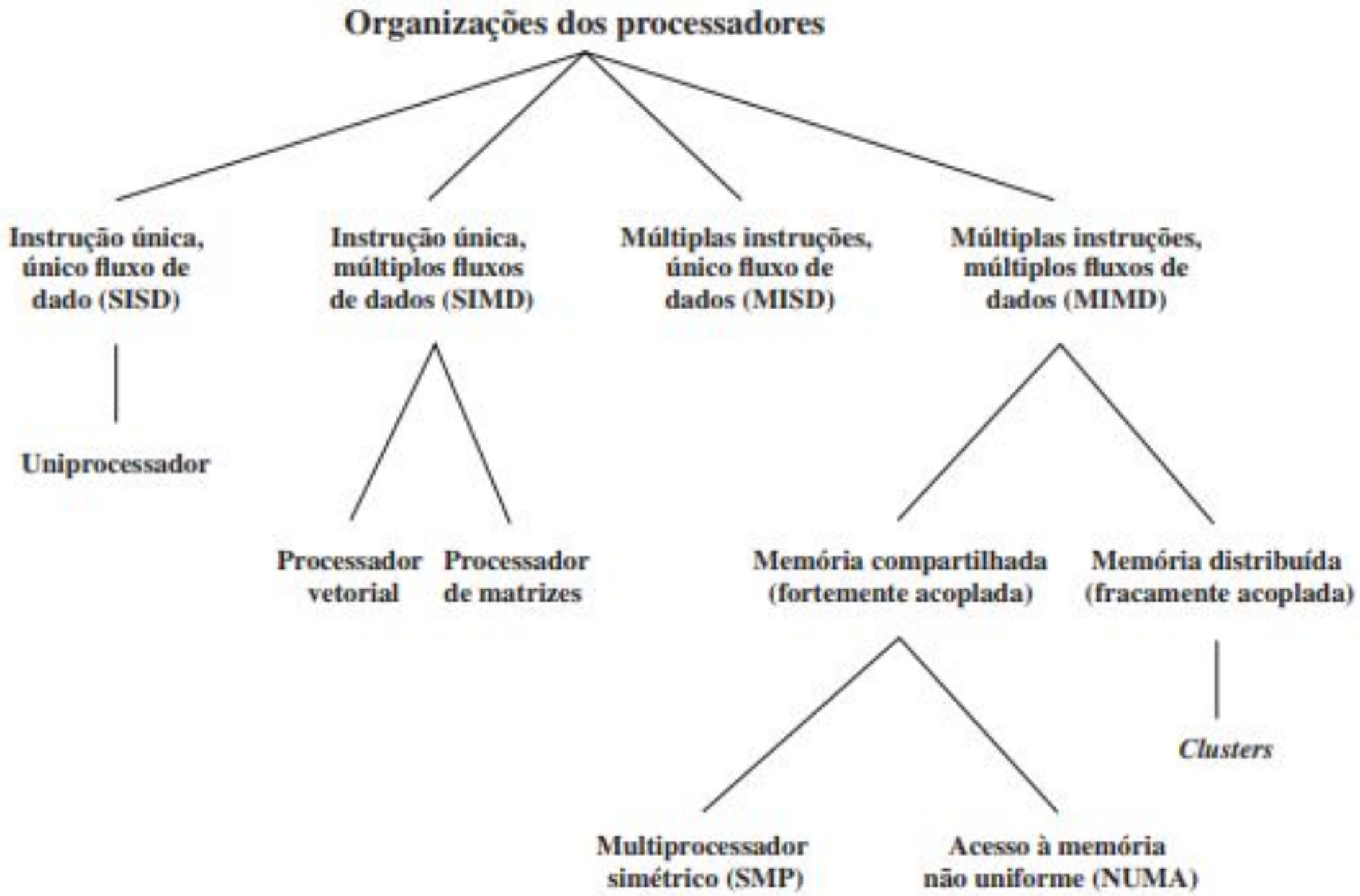
Organização Paralela

Carlos Neto
Pablo Cavalcante
Isabela de Queiroz

UPE
UNIVERSIDADE
DE PERNAMBUCO

Organizações de múltiplos processadores

- Tipos de sistemas de processadores paralelos
  - SISD — do inglês, Single Instruction, Single Data
    - Único processador; única sequência; única memória
  - SIMD — do inglês, Single Instruction, Multiple Data
    - única instrução; vários elementos de processamento
  - MISD — do inglês, Multiple Instruction, Single Data
    - Uma sequência de dados; conjunto de processadores
  - MIMD — do inglês, Multiple Instruction, Multiple Data
    - Conjunto de processadores; sequências de instruções diferentes; diferentes conjuntos de dados

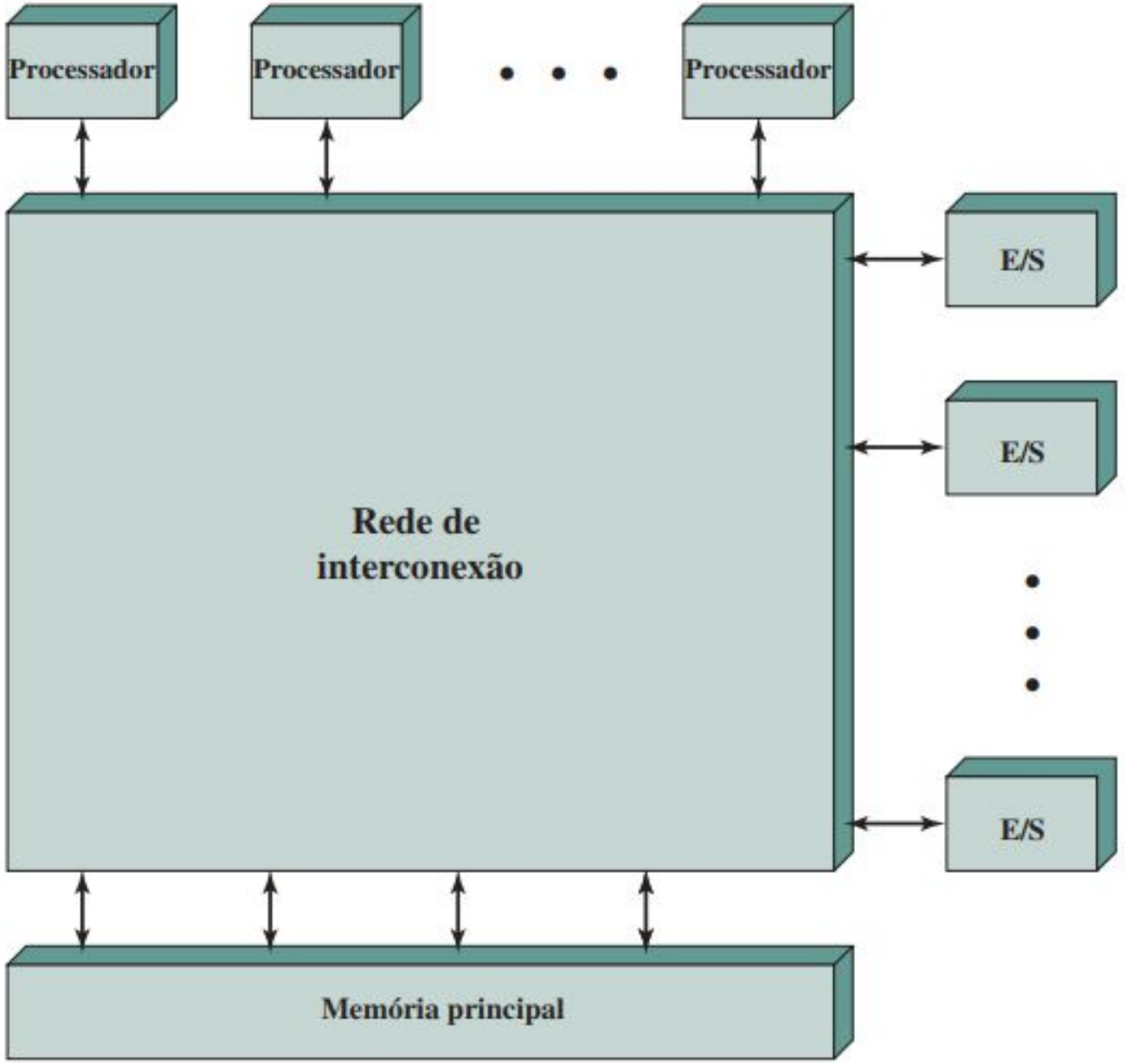

Organizações dos processadores
Instrução única, único fluxo de dado (SISD)
Instrução única, múltiplos fluxos de dados (SIMD)
Múltiplas instruções, único fluxo de dados (MISD)
Múltiplas instruções, múltiplos fluxos de dados (MIMD)
Uniprocessador
Processador vetorial
Processador de matrizes
Memória compartilhada (fortemente acoplada)
Memória distribuída (fracamente acoplada)
Clusters
Multiprocessador simétrico (SMP)
Acesso à memória não uniforme (NUMA)

Multiprocessadores simétricos (SMP)

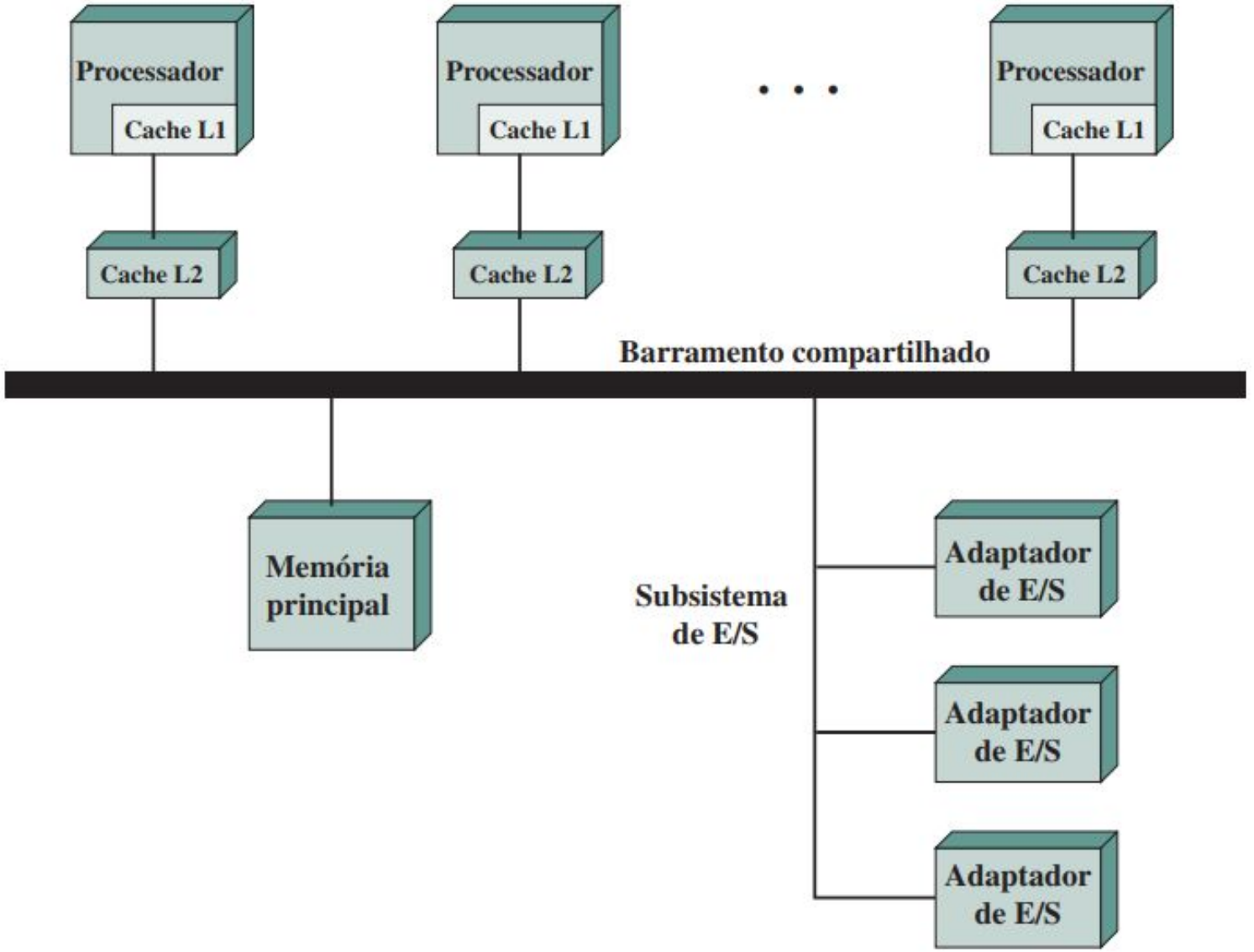
- Sistema de computação independente

- Características
  - Dois ou mais processadores semelhantes
  - Mesma memória principal e mesmos recursos de E/S
  - Acesso aos dispositivos de E/S compartilhado
  - Mesmas funções
  - Sistema operacional integrado

- Vantagens
  - Desempenho
  - Disponibilidade
  - Crescimento incremental
  - Escalabilidade



- Cada processador é autossuficiente
- Acesso a memória principal compartilhada
- Processadores podem trocar sinais diretamente



- Cada processador é autossuficiente
- Acesso a memória principal compartilhada
- Processadores podem trocar sinais diretamente

- Recursos
  - Endereçamento: diferenciar os módulos no barramento
  - Arbitração: Qualquer módulo E/S pode ser o “mestre”
  - Tempo compartilhado: Um módulo controla o barramento por vez

- Organização do barramento
  - Simplicidade
  - Flexibilidade
  - Confiabilidade

- Desvantagem: desempenho
  - barramento muito acessado
  - limitação na velocidade do sistema

Considerações de projeto de SO

- O SO tem a responsabilidade de escalonar a execução
- O SO deve fornecer a funcionalidade e os recursos
- Questões de projeto
  - Processos concorrentes simultâneos
  - Escalonamento
  - Sincronização
  - Gerenciamento de memória
  - Confiabilidade e tolerância a falhas

Coerência de cache e protocolo MESI

- Problema: cópias do mesmo dado em caches diferentes

- Considerações
  - Write back: escrita apenas na cache
  - Write Through: gravação na memória principal e na cache

- Protocolo MESI
  - Dois bits para cada estado:
    - Modificada
    - Exclusiva
    - Compartilhada
    - Inválida

	M Modificada	E Exclusiva	S (*shared*) Compartilhada	I Inválida
Essa linha da cache está válida?	Sim	Sim	Sim	Não
A cópia da memória está...	desatualizada	válida	válida	—
Há cópias em outras caches?	Não	Não	Talvez	Talvez
Uma escrita nessa linha...	não vai para o barramento	não vai para o barramento	vai para o barramento e atualiza a cache	vai diretamente para o barramento

LEITURA COM FALHA (*READ MISS*) Quando ocorre uma falha de leitura em uma cache local, o processador inicia uma leitura de memória para ler a linha da memória principal que contém o endereço que está faltando. O processador insere um sinal no barramento que avisa todos os outros processadores/unidades de cache para monitorarem a transação. Há vários desfechos possíveis:

- Se outra cache possui uma cópia limpa (não modificada desde a leitura da memória) da linha no estado exclusivo, ela retorna um sinal indicando que compartilha essa linha. O processador que respondeu passa o estado da sua cópia de exclusiva para compartilhada e o processador que iniciou lê a linha da memória principal e passa a linha na sua cache de inválida para compartilhada.
- Se uma ou mais caches têm uma cópia limpa da linha no estado compartilhado, cada uma delas sinaliza que compartilha essa linha. O processador que iniciou lê a linha e passa-a na sua cache de inválida para compartilhada.
- Se outra cache tem uma cópia modificada da linha, então essa cache bloqueia a leitura de memória e fornece a linha para a cache que requisitou por meio do barramento compartilhado. A cache que respondeu muda, então, a sua linha de modificada para compartilhada[1]. A linha enviada para a cache requisitante é também recebida e processada pelo controlador de memória, que guarda o bloco na memória.
- Se nenhuma outra cache tem uma cópia da linha (limpa ou modificada), então nenhum sinal é retornado. O processador que iniciou lê a linha e passa-a na sua cache de inválida para exclusiva.

Multithreading e chips multiprocessadores

- Taxa de execução das instruções

$$\text{Taxa MIPS} = f \times IPC$$

- Multithreading
  - Alto grau de paralelismo
  - Sem aumento de complexidade
  - Sem aumento do consumo de energia

Multithreading e chips multiprocessadores

- Considerações
  - Processo: instância de um programa executando em um computador
    - Posse do recurso
    - Escalonamento/execução
  - Troca de processos
  - *Thread*
  - Troca de *thread*

Multithreading e chips multiprocessadores

- Multithreading explícito
  - Multithreading intercalado: O processador lida com dois ou mais contextos de thread ao mesmo tempo, trocando de um thread para outro a cada ciclo de clock.
  - Multithreading bloqueado: As instruções executadas sucessivamente até ocorrer um evento que cause um atraso. Esse evento induz uma troca para outro thread
  - Multithreading simultâneo (SMT): instruções são enviadas simultaneamente a partir de múltiplos threads para unidades de execução de um processador superescalar
  - Chip multiprocessador: o processador inteiro é replicado em um único chip e cada processador lida com threads separados

Clusters

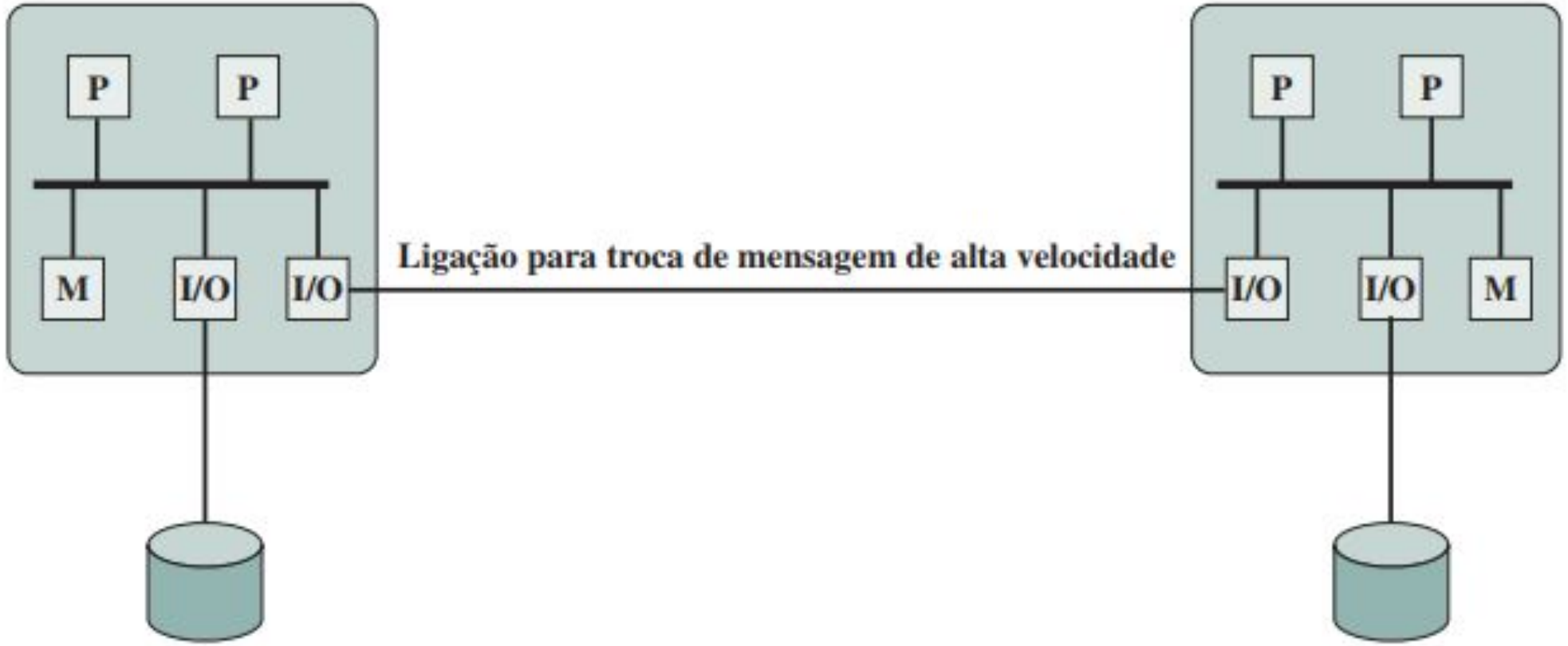
- “Um grupo de computadores completos interconectados trabalhando juntos, como um recurso computacional unificado que pode criar a ilusão de ser uma única máquina.”

- Agrupamento de computadores (clustering)
  - Alternativa para SMP
  - Alto desempenho e disponibilidade

- Benefícios
  - Escalabilidade absoluta
  - Escalabilidade incremental
  - Alta disponibilidade
  - Preço/desempenho superior

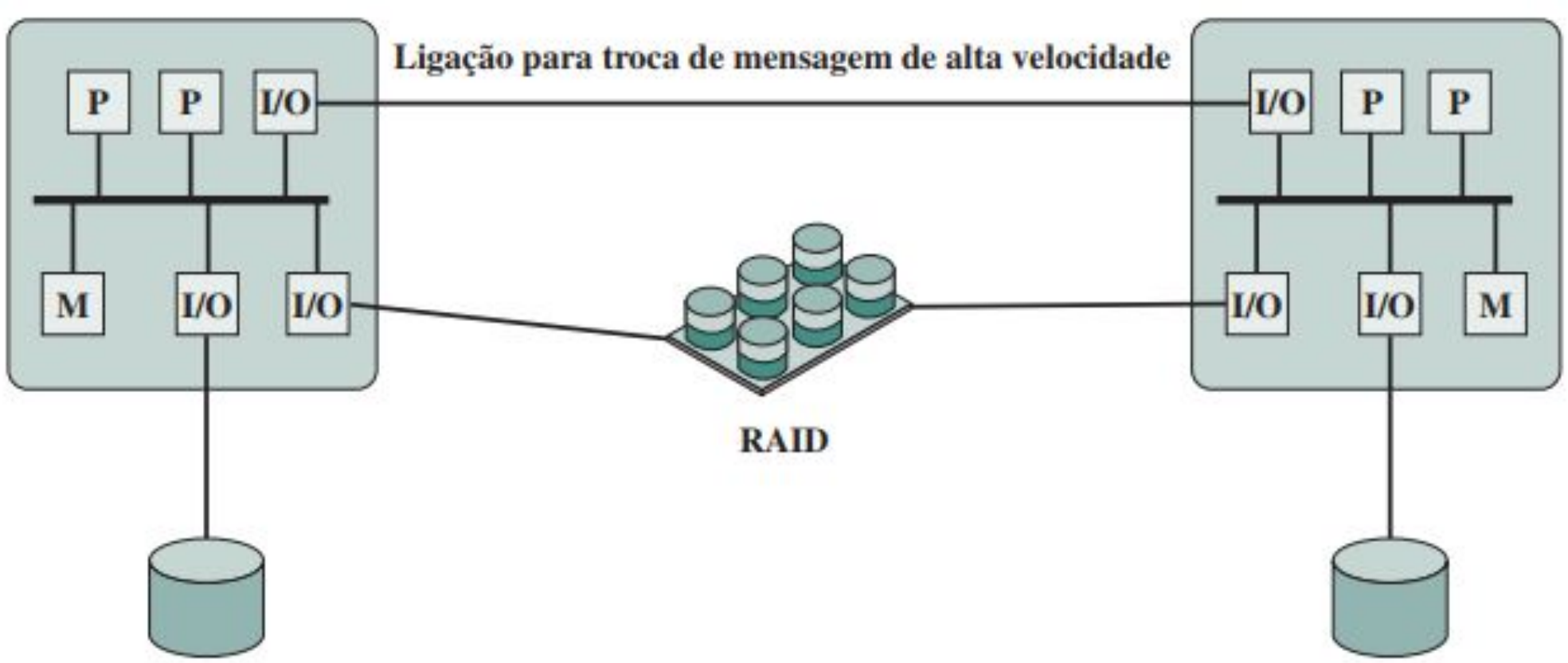
Clusters

- Configurações de Cluster



Clusters

- Configurações de Cluster



Clusters

Método de *clustering*	Descrição	Benefícios	Limitações
Secundário passivo (*passive standby*)	Um servidor secundário assume em caso de falha do servidor primário.	Fácil de implementar.	Custo alto porque o servidor secundário está indisponível para outras tarefas de processamento.
Secundário ativo	O servidor secundário é usado também para tarefas de processamento.	Custo reduzido porque servidores secundários podem ser usados para processamento.	Complexidade aumentada.
Servidores separados	Possuem seus próprios discos. Dados são copiados continuamente do servidor primário para o secundário.	Alta disponibilidade.	Grande sobrecarga de rede e servidores por causa das operações de cópia.
Servidores conectados aos discos	Servidores são ligados aos mesmos discos, mas cada servidor possui seus discos. Se um servidor falha, seus discos são assumidos por outro servidor.	Carga de rede e servidores reduzida por causa da eliminação das operações de cópia.	Costuma requerer espelhamento de discos ou tecnologia RAID para compensar o risco da falha de disco.
Servidores que compartilham discos	Vários servidores compartilham simultaneamente o acesso a discos.	Baixa carga de rede e servidores. Risco reduzido de inatividade causada por falha de disco.	Requer software de gerenciamento de bloqueio. Normalmente usado com tecnologia de espelhamento ou RAID.

Clusters

- Classificações de Clusters
  - Servidores separados
    - Cada computador tem seus próprios discos
    - Alto desempenho e disponibilidade
    - Software de gerenciamento ou escalonamento
    - Tolerância a falhas
  - Sem Compartilhamento
    - Discos particionados em volumes
    - Cada volume é propriedade de um único computador

Clusters

- Clusters x SMP
  - SMP: mais fácil de gerenciar e configurar; ocupa menos espaço físico; produtos estáveis
  - Clusters: melhor escalabilidade; melhor disponibilidade

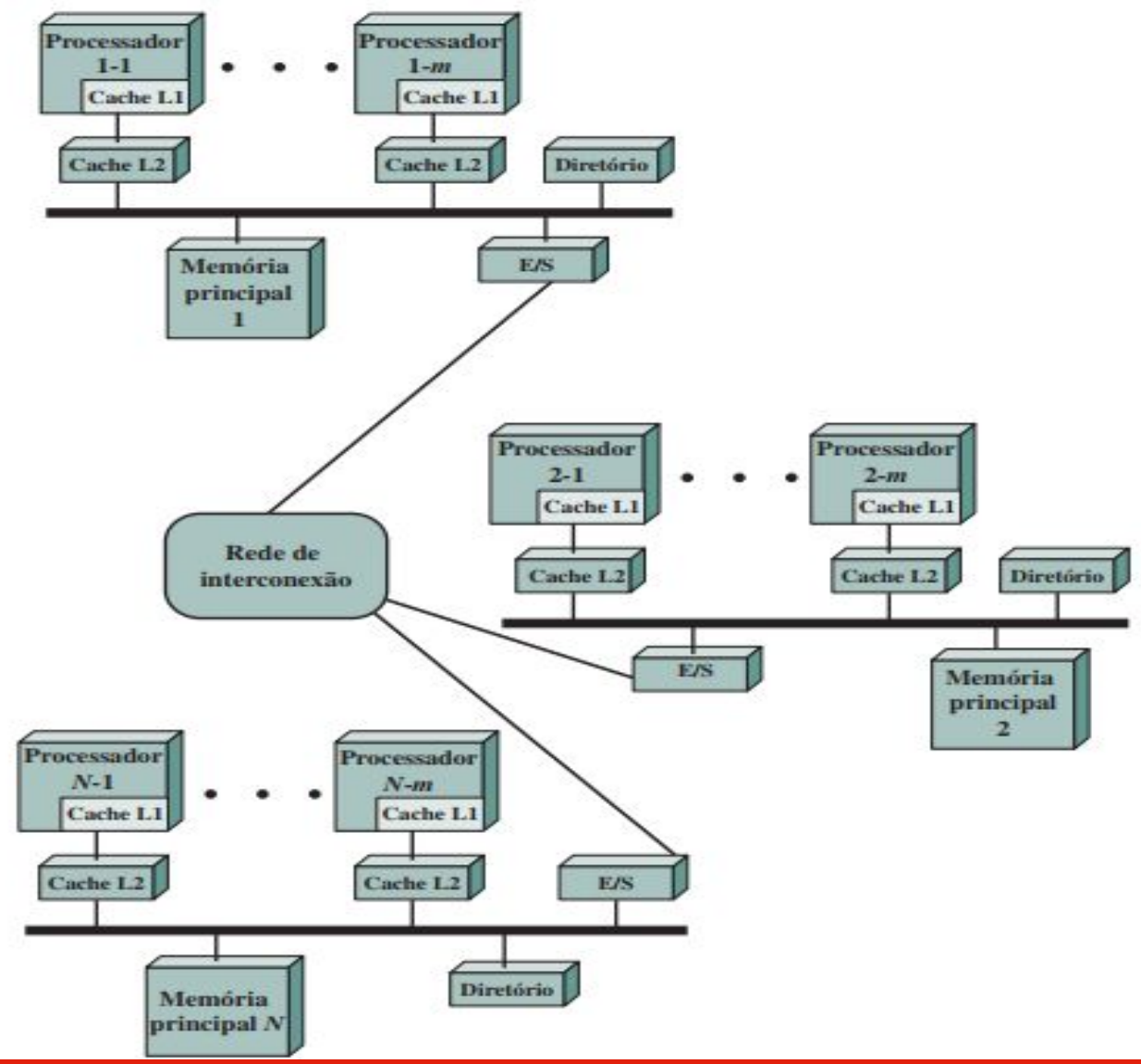
Acesso não uniforme à memória

- No SMP há um certo limite para o número de processadores
  - Mais processadores, menor desempenho
- NUMA: Multiprocessamento em grande escala mantendo as características do SMP

Acesso não uniforme à memória

- Definições
  - Acesso uniforme à memória (UMA)
    - processadores podem acessar todas as partes da memória principal
    - Mesmo tempo de acesso
  - Acesso não uniforme à memória (NUMA)
    - processadores podem acessar todas as partes da memória principal
    - tempo de acesso depende da região da memória a ser acessada
  - NUMA com coerência de cache: coerência de cache mantida entre os processadores

Acesso não uniforme à memória



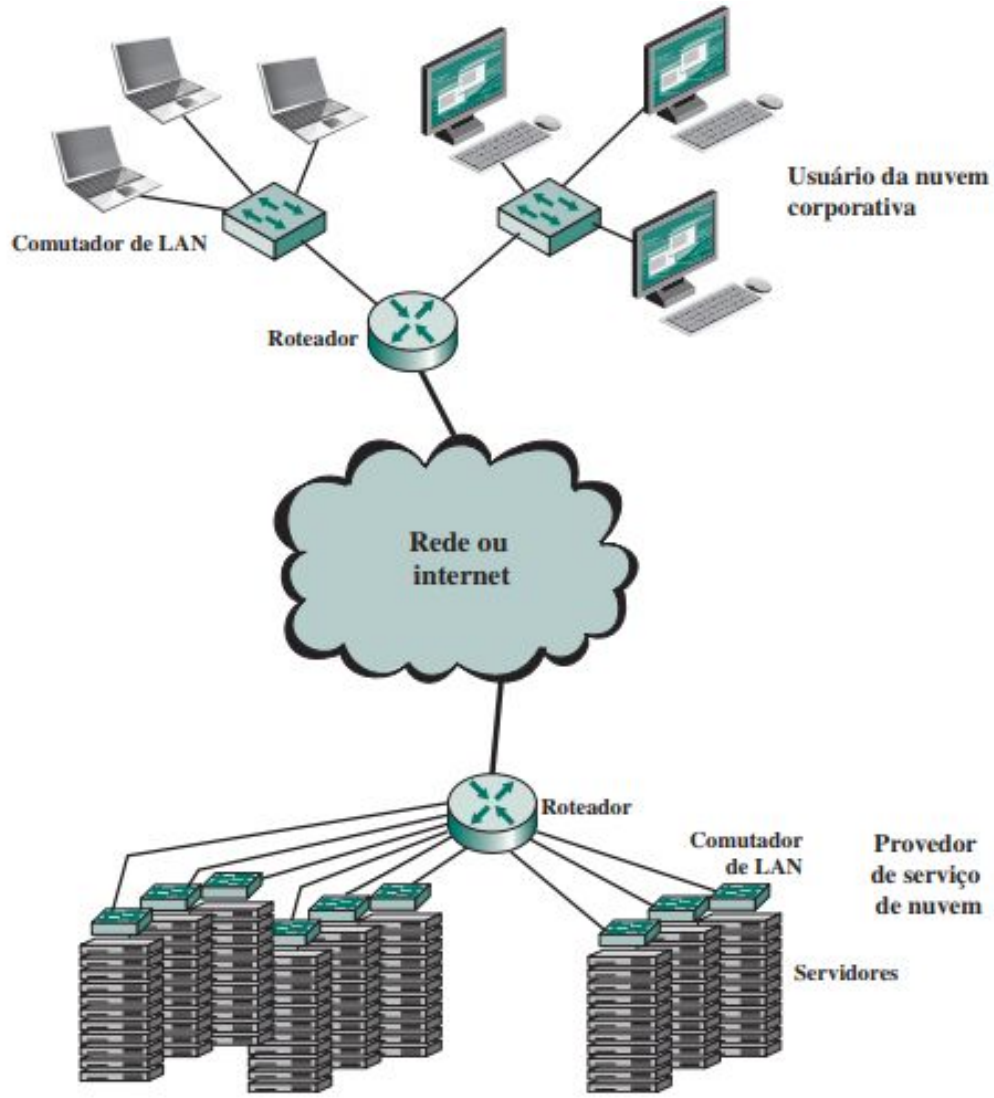
Acesso não uniforme à memória

- Prós e Contras
  - Desempenho mais eficiente que no SMP
  - Tráfego no barramento depende da limitação do mesmo
  - Não é transparente como o SMP
  - Disponibilidade não é eficiente

Computação em nuvem

- Características
  - Acesso abrangente à rede
  - Elasticidade rápida
  - Serviço mensurado
  - Autoatendimento sob demanda
  - Agrupamento de recursos
- Modelos de serviço
  - Software como um serviço (SaaS)
  - Plataforma como um serviço (PaaS)
  - Infraestrutura como um serviço (IaaS)
- Modelos de implantação
  - Nuvem pública
  - Nuvem privada
  - Nuvem comunitária
  - Nuvem híbrida

Arquitetura da computação em nuvem



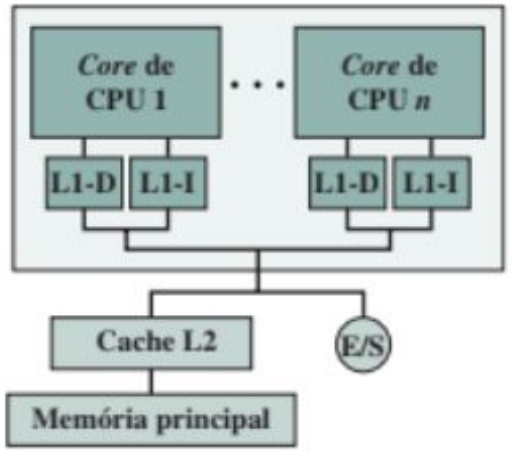
Computação em nuvem

- "Atores"
  - Consumidor de nuvem
  - Provedor de nuvem
  - Auditor de nuvem
  - Agente de nuvem
  - Operador de nuvem

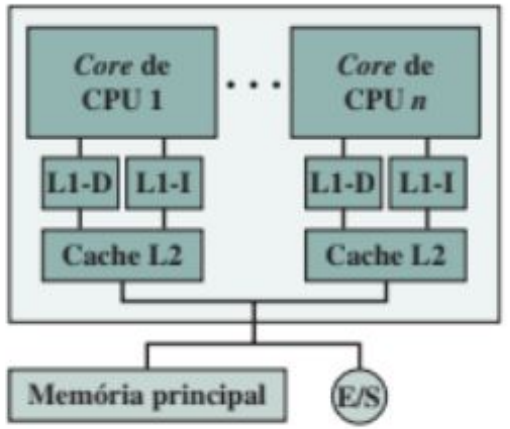
COMPUTADORES MULTICORE

- Chip multiprocessador
- Duas ou mais unidades de processador em uma peça única de silício
- Questões sobre desempenho do hardware
  - Aumento no paralelismo e na complexidade
    - Pipeline
    - Superescalar
    - Multithreading simultâneo (SMT)

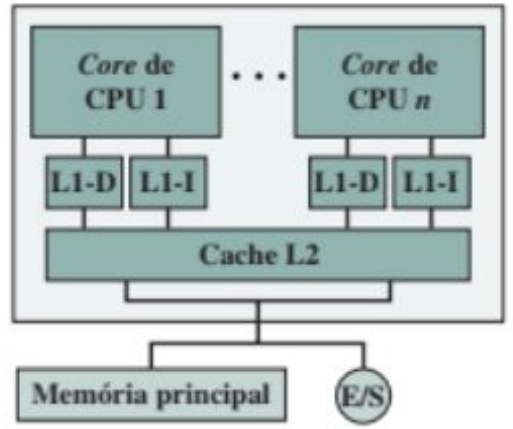
- Organização multicore
  - Aspectos gerais
  - Níveis de Cache



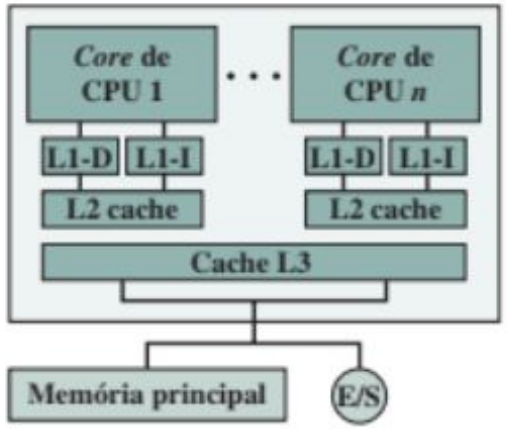
(a) Cache L1 dedicada



(b) Cache L2 dedicada



(c) Cache L2 compartilhada



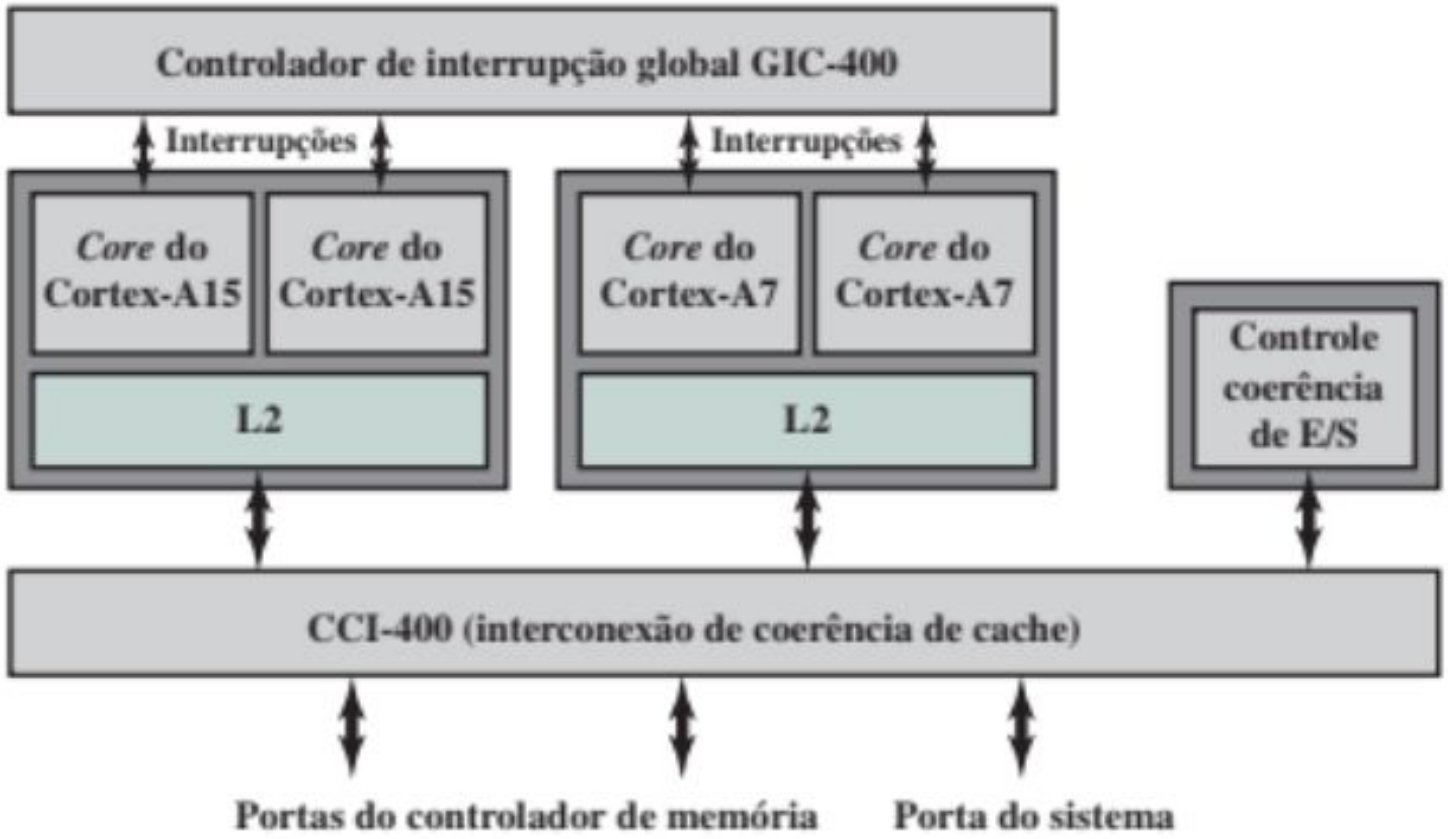
(d) Cache L3 compartilhada

- Multithreading simultâneo
  - ex: multicore com quatro cores
  - SMT que suporta quatro threads simultâneos em cada core

- Organização multicore heterogênea
  - Arquiteturas de conjunto de instruções diferentes
    - Exemplo
      - Multicore CPU/GPU
      - Multicore CPU/DSP

  - Arquiteturas de conjunto de instruções equivalentes
    - EXEMPLO: arquitetura do big.Little da ARM

big.Little da ARM



- Intel Core i7-990X
- ARM Cortex-A15 MPCore
    - aplicações → computação móvel, servidores domésticos de ponta e infraestrutura sem fio
    - Elementos-chave
        - Controlador de interrupção genérico (GIC)
        - Timer genérico
        - Rastreamento
            - monitoramento de desempenho
            - Ferramentas de rastreamento do programa
        - Core
        - Cache L1
        - Cache L2
        - Unidade de Controle de Monitoração (SCU — do inglês, Snoop Control Unit)
- Mainframe do zEnterprise EC12 da IBM
    - oito chips / 7.356 conexões. / mais de 23 bilhões de transistores
    - Elementos-chave
        - Unidade do processador (PU)
        - Controle de armazenamento (CA)

Unidades de processamento gráfico de uso geral

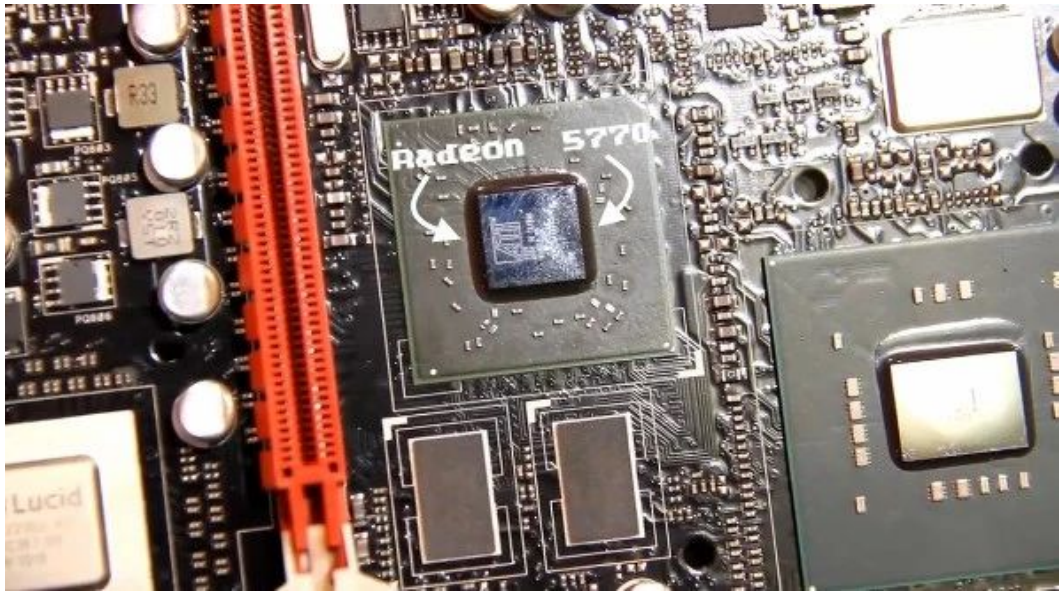
- GPU (Graphics Processing Unit) x Placa de vídeo
    - o termo GPU se refere apenas à unidade de processamento e não à placa de vídeo como um todo
- CPU x GPU
    - Ambas são voltadas para o processamento
    - GPU é voltada para atividades gráficas (jogos, edição de vídeo, etc)
    - GPU funciona de forma paralela à CPU
- GPU onboard
- GPU offboard

Paralelismo

- Exemplo do supermercado

GPU onboard

- Vem soldado com a placa mãe;
- Vantagens:
  - Mais simples
  - Consomem menos energia
  - Diminui o custo
- Desvantagens:
  - Se queimar, não tem como trocar por uma melhor
  - Menor desempenho



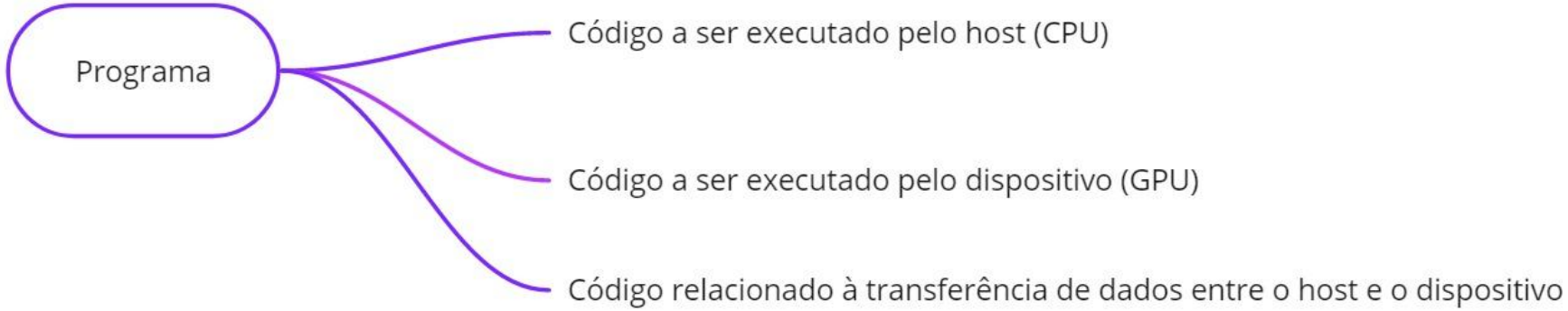
GPU offboard

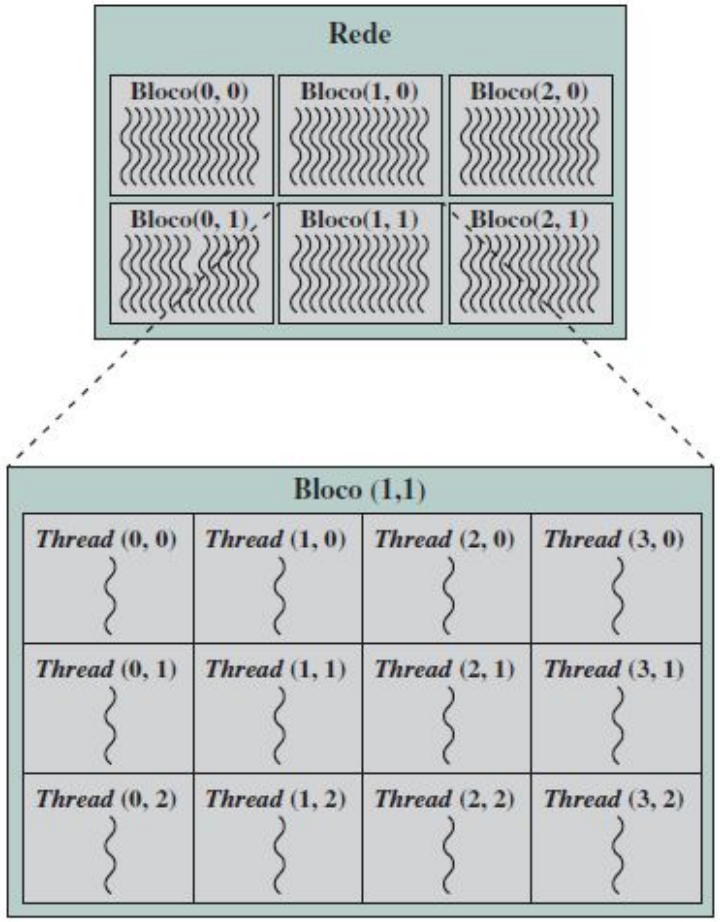
- Vantagens:
  - Adaptabilidade: o usuário pode montar a própria configuração de acordo com a necessidade
  - Mais rápida
- Desvantagem:
  - Mais cara
  - Precisa instalar drivers

CUDA (Compute Unified Device Architecture)

- Plataforma de programação paralela NVDIA e implementada nas suas GPUs;
- Core CUDA (Explicação nos próximos slides)
- CUDA C:
    - Desenvolvedores podem usar o potencial de processamento paralelo de uma GPU.
    - Baseada em C++

Programa CUDA C





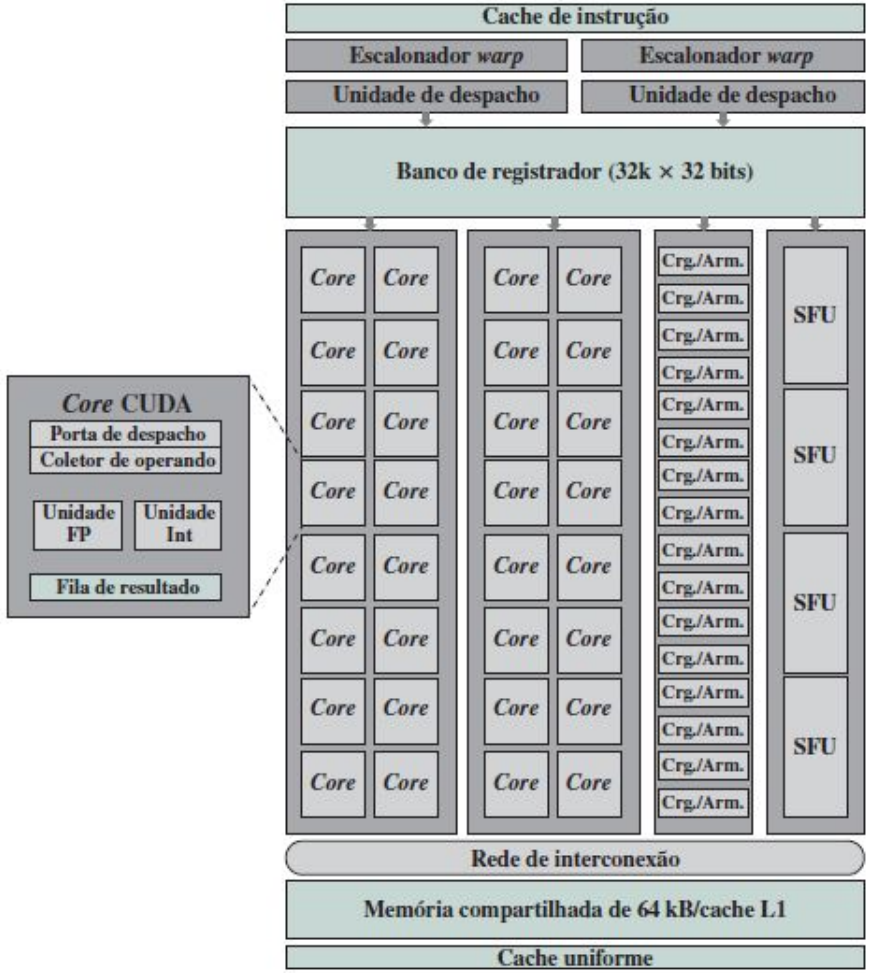
Termo de CUDA	Definição	Componente equivalente de hardware de GPU
Kernel	Código paralelo na forma de uma função a ser executada na GPU	Não se aplica
Thread	Uma instância do kernel na GPU	*Core* processador de uma GPU/CUDA
Bloco	Um grupo de *threads* atribuído a um MS em particular	Multiprocessador de CUDA (MS)
Rede	A GPU	GPU

CPUXGPU

- *"GPU usa uma arquitetura SIMD massivamente paralela (única instrução e múltiplos dados) para executar, sobretudo, operações matemáticas. Sendo assim, uma GPU não necessita das mesmas capacidades complexas da lógica de controle da CPU (ou seja, execução fora de ordem, previsão de desvio, hazard de dados etc.)."*

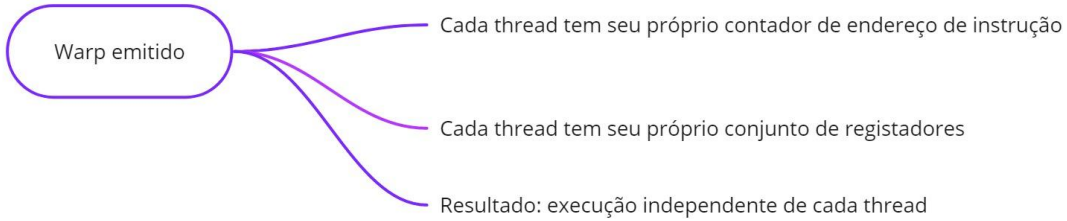
Arquitetura de uma GPU

Arquitetura MS única.



Escalonador warp

- Warps: São agrupamentos de 32 threads que começam no mesmo endereço inicial e seus IDs de thread são consecutivos
- Escalonador de warp: Separa cada bloco de thread em warps.



- A GPU é mais eficiente quando está processando tantos warps quanto possível para manter os cores

Quando usar GPU como um coprocessador?

- Programas que tem muitas partes que ´podem ser paralelizadas!

Questões

1) Quais são as diferenças entre kernel, thread e bloco?

2) Quais as questões de projeto de um SO para uma organização paralela? Comente brevemente sobre estas.

3) Resuma a diferença entre pipeline simples de instruções, superescalar e multithreading simultâneo.