ERAD 2010
10ª Escola Regional de Alto Desempenho
16 a 19 de março de 2010 - Passo Fundo - RS

# Arquiteturas Paralelas


**Prof. Ms. Marcos José Brusso**
*brusso@upf.br*
Universidade de Passo Fundo

**Prof. Dr. César A. F. De Rose**
*derose@inf.pucrs.br*
Pontifícia Universidade Católica do Rio Grande do Sul


## Apresentação – Prof. Marcos José Brusso

- Formação
  - Graduação: Ciência da Computação, UPF/1994
  - Mestrado: Ciência da Computação, UFRGS/2000
- Atividades
  - Professor Adjunto do ICEG/UPF
  - Coordenador da Especialização em Desenvolvimento de Software
  - Coordenador projeto Kelix



## Apresentação – Prof. César De Rose

- Formação
  - Graduação em Ciência da Computação - PUCRS (1990),
  - Mestrado em Ciência da Computação - UFRGS (1993)
  - Doutorado em Ciência da Computação pela Universidade de Karlsruhe, Alemanha (1998)
- Atividades
  - Professor Adjunto da PUCRS
  - Coordena o Laboratório de Alto Desempenho da PUCRS (LAD-PUCRS)



## Sumário



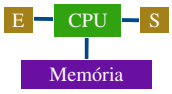


# Introdução

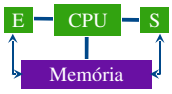


## Evolução das Arquiteturas

- Arquitetura Tradicional
  - Uma única unidade ativa

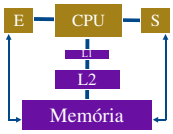



- Unidades de E/S autônomas
  - Ex.: DMA
  - Multiprogramação

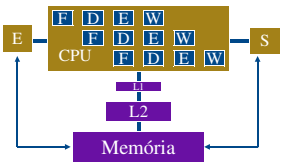


## Evolução das Arquiteturas

- Hierarquia de Memória
  - Acelerar alimentação da CPU




- Pipeline de instrução
  - Superescalar
- Pipeline de dados
  - Máquinas vetoriais






## Objetivo desta Evolução

- Acelerar o processamento dos dados pela CPU
  - Liberando CPU (delegando tarefas)
    - Controle do barramento
    - Tratamento de E/S
  - Acelerando a alimentação da CPU
    - Hierarquia de memória
  - Sobrepondo ciclos da CPU
    - Pipeline de instrução
- Acelerar o processamento dos dados construindo arquiteturas com múltiplas CPU's
  - **Arquiteturas Paralelas**



# Motivação



## Por Que Pesquisar AP ?

- Contribui para o ganho de desempenho de arquiteturas "convencionais"
- Alternativa para quando limites físicos forem atingidos
- Alternativa para aplicações com demanda imediata por alto desempenho
  - Simulação (previsão do tempo, modelos físicos, biológicos)
  - Computação gráfica
  - ...



## Por Que Pesquisar AP ?

- Solução de aplicações complexas (científicas, industriais e militares)
  - Meteorologia
  - Prospeção de petróleo
    - Análise de local para perfuração de poços de petróleo
  - Simulações físicas
    - Aerodinâmica; energia nuclear
  - Matemática computacional
    - Análise de algoritmos para criptografia
  - Bioinformática
    - Simulação computacional da dinâmica molecular de proteínas



## Por Que Estudar AP?

- Domínio da terminologia utilizada na especificação de arquiteturas
- Escolha / Construção da melhor arquitetura para o uso desejado
- Programação eficiente da máquina
- No caso de PPD, conhecimento da arquitetura da máquina influencia diretamente o desempenho da aplicação

# Classificação de Máquinas Paralelas



## Por Que Estudar Classificações ?

- Identificar o critério da classificação
  - Por que é importante
  - Quais as suas implicações
- Analisar todas as possibilidades
  - Mesmo as classes que não foram implementadas
  - Ou implementações que não deram certo
- Como ocorreu a evolução da área
  - Como pode evoluir



## Classificação de Flynn

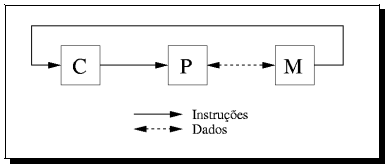
- Classificação genérica (1970)
- Diferencia se o fluxo de instruções (*instruction stream*) e o fluxo de dados (*data stream*) são múltiplos ou não

| | Single Instruction | Multiple Instruction |
|---|---|---|
| **Single Data** | SISD | MISD |
| **Multiple Data** | SIMD | MIMD |



## Classe SISD

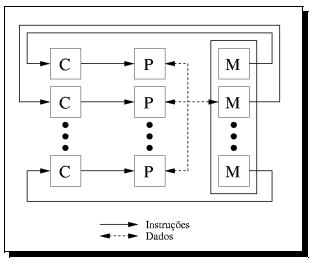
- *Single Instruction Stream, Single Data Stream*
  - Um único fluxo de instruções
  - Um único fluxo de dados
- Arquiteturas tradicionais não paralelas
  - Máquinas de von Neumann tradicionais






## Classe MISD

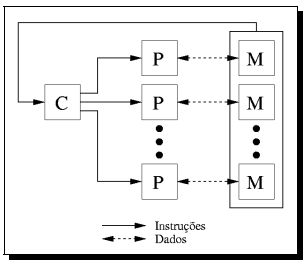
- *Multiple Instruction Stream, Single Data Stream*
  - Múltiplos fluxos de instruções
  - Um único fluxo de dados
- Ainda sem implementação





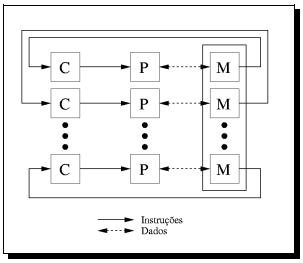

## Classe SIMD

- *Single Instruction Stream, Multiple Data Stream*
  - Um único fluxo de instruções
  - Múltiplos fluxos de dados
- Execução síncrona
- Arquiteturas Array
  - CM-2, MP-2



## Classe MIMD

- *Multiple Instruction Stream, Multiple Data Stream*
  - Múltiplos fluxos de instruções
  - Múltiplos fluxos de dados
- Vários programas sobre vários dados
- Arquiteturas Paralelas Modernas





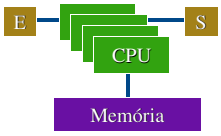

## Classificação segundo o Compartilhamento de Memória

- Dependendo da máquina paralela utilizar uma memória compartilhada por todos os processadores, pode-se diferenciar:
  - Multiprocessadores
  - Multicomputadores



## Multiprocessador

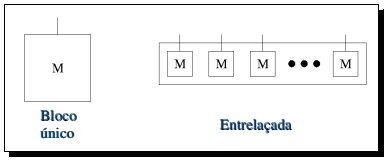
- Compartilha uma memória central
  - Arquitetura tradicional com vários processadores
  - Um único espaço de endereçamento
  - Comunicação através da memória
    - Variáveis compartilhadas






## Multiprocessador

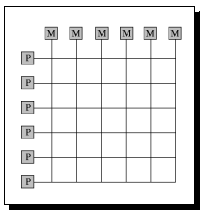
- Em um multiprocessador a memória é disputada pelos processadores
  - Muitas vezes endereços são diferentes
  - Posso quebrar memória em diferentes módulos para permitir múltiplos acessos
    - Memória Entrelaçada (*interleaved*)






## Multiprocessador

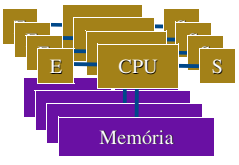
- Não adianta a memória suportar múltiplos acessos se o barramento suporta apenas uma transação por vez
- Ideal: rede não bloqueante com suporte a várias transações simultâneas
- Ex: Matriz de Chaveamento (*crossbar*)





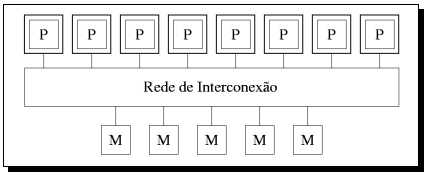

## Multicomputador

- Não compartilha memória
  - Interligação de várias arquiteturas tradicionais
  - Cada uma possui sua memória local
  - Múltiplos espaços de endereçamento privados
  - Comunicação por troca de mensagens



## Classificação Segundo o Tipo de Acesso à Memória

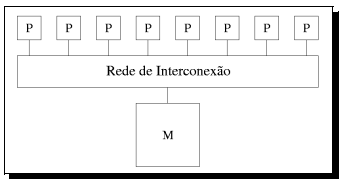
- Multiprocessadores
  - UMA
  - NCC-NUMA
  - CC-NUMA
  - SC-NUMA
  - COMA






## Multiprocessadores UMA

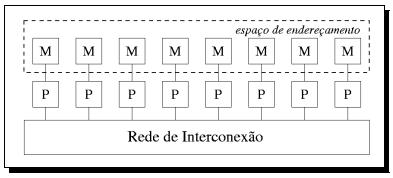
- *Uniform Memory Access*
- Memória centralizada (mesma distância de todos os processadores)
- Custo único de acesso
- Necessário tratar coerência das caches






## Multiprocessadores NUMA

- *Non Uniform Memory Access*
- Único espaço de endereçamento
- Memória distribuída (distâncias diferentes)
- Custo não uniforme de acesso à memória





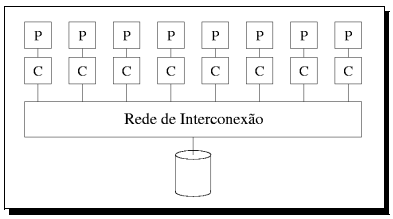

## Multiprocessadores NUMA

- Em relação ao tratamento do problema de coerência de cache
  - NCC-NUMA
    - *Non Cache-Coherent NUMA*
  - CC-NUMA
    - *Cache-Coherent NUMA*
    - Implementada em hardware
  - SC-NUMA
    - *Software-Coherent NUMA*
    - Implementada em software
    - DSM (*Distributed Shared Memory*)



## Multiprocessadores COMA

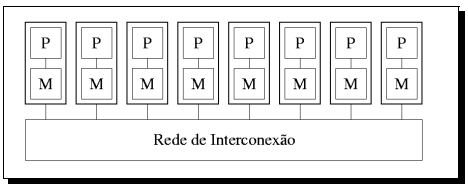
- *Cache-only Memory Architecture*
- Memórias locais são caches (COMA caches)
- Gerência de caches na MMU



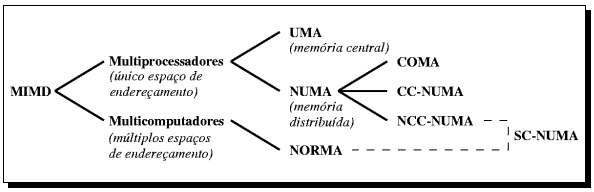



## Multicomputadores NORMA

- Non-Remote Memory Access)
- Apenas acesso local à memória



## Resumo da Classificação






# Redes de Interconexão



## Redes de Interconexão

- Implementa comunicação N:N com redes estáticas ou dinâmicas
- Estática
  - Roteamento em hardware
    - Anel
    - Torus
    - ...
- Dinâmica
  - Chaveadores (*switches*)

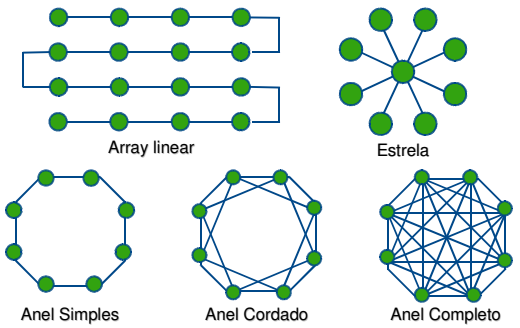


## Redes Estáticas

- Interligadas através de ligações fixas
- Entre cada componente existe ligação direta dedicada
- Topologia (estrutura de interligação) determina características da rede
- No caso das máquinas paralelas são normalmente regulares



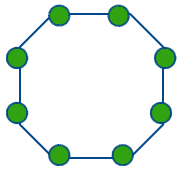
## Redes Estáticas





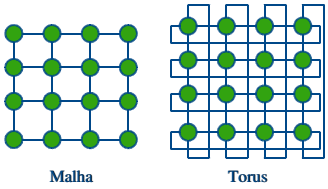

## Anel

- Baixa escalabilidade (aumento um a um até 12)
- Problemas com Tolerância a Falhas
- Uni ou bidirecional
- Grau do nó: 2



## Torus

- Malha com extremidades interligadas
- Roteamento simplificado
- Boa escalabilidade (aumento linha ou coluna até 12x12)
- Grau do nó: 4





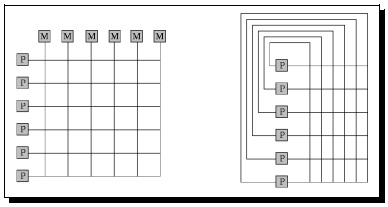

## Redes Dinâmicas

- Não há topologia fixa
- Rede adapta-se dinamicamente, por demanda
  - Barramento
  - Matriz de Chaveamento (*Crossbar*)
  - Redes Multinível



## Matriz de Chaveamento

- Crossbar
- Baixa escalabilidade (limite é o número de portas)
- Alto custo
- Bidirecional
- Grau do nó 1





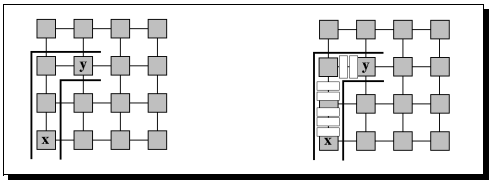
## Roteamento

- Rede de interconexão normalmente não possui ligações diretas entre todos os nós
- Mensagem precisa trafegar por nós intermediários para chegar ao destino
- A condução da mensagem é chamada de roteamento
- Duas formas de condução
  - Chaveamento de circuito (*circuit switching*)
  - Chaveamento de pacotes (*packet switching*)



## Chaveamento de Circuito

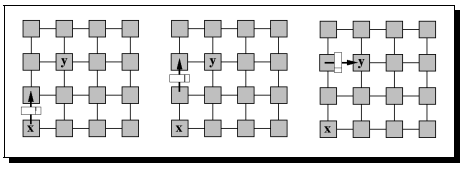
- Mais usado em redes de telecomunicações
- Estabeleço inicialmente o caminho (alto custo)
- Envio posteriormente os dados





## Chaveamento de Pacotes

- Mais comum em máquinas paralelas
- Não existe caminho pré-definido
- Sem custo inicial
- Custo adicional em cada nó
- Sem reserva de canal



# Plataformas Tradicionais para PPD



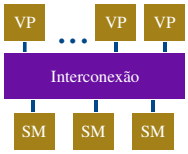
## Plataformas Tradicionais para PPD

- PVP – Processadores Vetoriais
- SMP – Multiprocessadores Simétricos com memória compartilhada
- MPP – Multicomputadores Maciçamente Paralelos com múltiplas memórias locais
- NOW – Redes de Estações de Trabalho

Arquitetura e características bem diferentes !!!



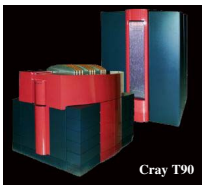
## PVP




- *Parallel Vector Processor*
- Memória compartilhada (UMA)
  - Comunicação através da memória
- Matriz de chaveamento
  - Permite acesso concorrente a memória
- Baixa escalabilidade (poucos processadores)
- Grandes registradores, sem caches
- Ex: Cray C90, Cray T-90, NEC SX-4



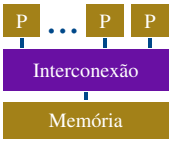
## PVP: Exemplos







## SMP




- *Symmetric Multiprocessor*
- Memória compartilhada (UMA)
  - Comunicação através da memória
- Interconexão por barramento
- Baixa escalabilidade (poucos processadores)
- Fácil programação
- Ex: SGI Power Challenge, Sun Sparc Entrerprise, Servidor x86 Dual/Quad



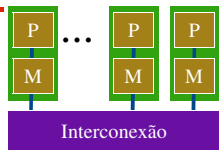
## SMP: Exemplos

**Sun SPARC Enterprise T5440 Server**
(até 4 UltraSPARC T2 Plus)

**IBM eServer p5 550**
(2 ou 4 POWER5)

## MPP

- *Massively Parallel Processors*
- Múltiplas memórias locais
  - Comunicação por troca de mensagens
- Interconexão por rede de alta velocidade (proprietária)
- Boa escalabilidade (muitos processadores)
- Programação mais complicada
- Ex: Intel Paragon, Cray T3E, Thinking Machines CM-5





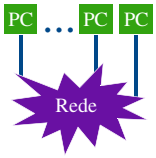

## MPP: Exemplos






## NOW

- *Network of Workstations*
- Múltiplas memórias locais
  - Comunicação por troca de mensagens
- Interconectados por rede tradicional
- Difícil programação
- Ex: PCs interligadas por rede Ethernet






## NOW: Exemplo



## Comparação

| | PVP | SMP | MPP | NOW |
|---|---|---|---|---|
| **Número de EPs** | Baixo | Baixo | Alto | Médio |
| **Escalabilidade** | Baixa | Baixa | Alta | Média |
| **Latência de Comunicação** | Baixa | Média | Baixa | Alta |
| **Programação** | Média | Fácil | Difícil | Difícil |
| | **PVP** | **SMP** | **MPP** | **NOW** |



## Surge Uma Nova Classe

- **Máquinas Agregadas**
  - **Cluster of Workstations (COW)**
- Redes de estações dedicadas ao Processamento Paralelo
- Interconectadas por novas tecnologias de redes locais (baixa latência)
- Otimização de NOW
- Procura aliar vantagens das outras quatro classes

## Máquinas Agregadas

- Baixo custo (NOW)
- Baixa latência na comunicação (MPP)
- Memória distribuída (MPP) e/ou compartilhada (SMP)
- Boa escalabilidade (MPP)



## COW: Exemplos

Clusters Amazônia e Ombrófila
CPAD-PUCRS/HP



## COW: Exemplos

HP i-cluster
Grenoble



## Construção de COW's

- Atualmente há duas tendências
- Interligadas por rede rápida
  - Impulsionada por fabricantes de placas de rede especiais
  - Alto custo por nó compromete escalabildade
  - Máquinas de pequeno e médio porte (dezenas de nós)
- Interligadas por rede Ethernet
  - Impulsionada por grandes fabricantes (HP, IBM)
  - Máquinas de grande porte (centenas de nós)



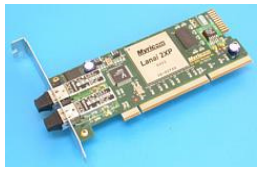
## Como Obter Baixa Latência

- Placas de interconexão (rede) otimizadas
- Conexão ponto-a-ponto entre estações
- Interconexão por redes estáticas ou dinâmicas
- Implementação de protocolos de rede em HW

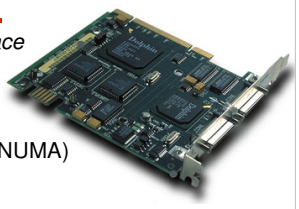


## Myrinet

- Implementa troca de mensagens (NORMA)
- Latência em torno de 2µs
- Vazão 1.2 GBytes/s
- Interligação através de switch de alto desempenho



## SCI



- *Scalable Coherent Interface*
- Padrão IEEE 1596-1992
- Implementa troca de mensagens e memória compartilhada (NORMA, NUMA)
- Latência em torno de 5µs
- Vazão 6.4 Gbits/s
- Interligação em anel ou switch de alto desempenho



## InfiniBand

- Tecnologia para comunicação
  - Computador/Computador
  - Computador/IO
- Latência em torno de 1µs
- Vazão 40 Gb/s
- Suporta *Remote Direct Memory Access* (RDMA)

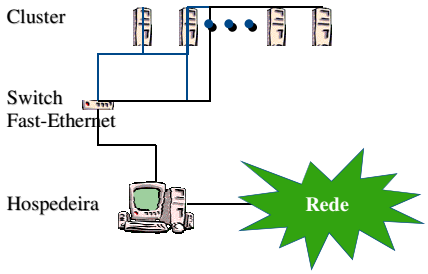


## COW - Configuração Mínima

- Aproveitamento das máquinas mais rápidas como nós (homogêneo) - 8 nós
- Aproveitamento de uma máquina como hospedeira
  - Não participa do cluster (simétrico)
  - Bloqueia acesso direto ao cluster
  - Função de console
- Sistema Operacional Linux
- Rede de interconexão de baixa latência ou uso de switch Fast-Ethernet



## Configuração Mínima



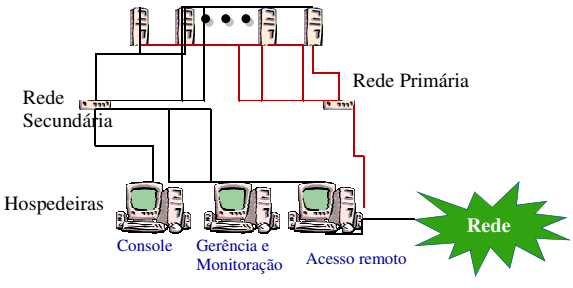



## Configuração Avançada

- Máquinas SMP como nós (dual) - 16 nós
- Redes de interconexão primária e secundária
  - Rede primária para comunicação (rede rápida)
  - Rede secundária para gerência e monitoração



## Configuração Avançada



## Lista TOP 500

- http://www.top500.org
- Benchmark: Linpack
- Última lista: novembro de 2009



## Top500: Os "Top 5"

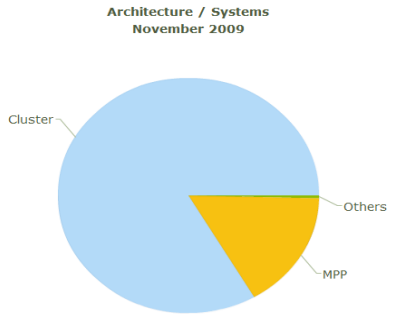
| Rank | Computer/Year Vendor | Cores | Rmax | Rpeak | Power |
|---|---|---|---|---|---|
| 1 | Jaguar - Cray XT5-HE Opteron Six Core 2.6 GHz / 2009<br>Cray Inc. | 224162 | 1759.00 | 2331.00 | 6950.60 |
| 2 | Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband / 2009<br>IBM | 122400 | 1042.00 | 1375.78 | 2345.50 |
| 3 | Kraken XT5 - Cray XT5-HE Opteron Six Core 2.6 GHz / 2009<br>Cray Inc. | 98928 | 831.70 | 1028.85 | |
| 4 | JUGENE - Blue Gene/P Solution / 2009<br>IBM | 294912 | 825.50 | 1002.70 | 2268.00 |
| 5 | Tianhe-1 - NUDT TH-1 Cluster, Xeon E5540/E5450, ATI Radeon HD 4870 2, Infiniband / 2009<br>NUDT | 71680 | 563.10 | 1206.19 | |



## Jaguar -Cray XT5

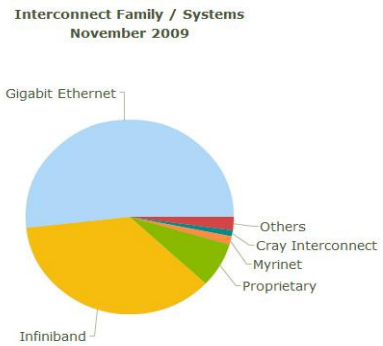



## Top500






## Top500






# Tópicos Atuais

## SMT - Simultaneous Multithreading

- Abordagem multi-thread
  - 2 ou mais threads podem executar simultaneamente no mesmo processador
  - Não há troca de contexto para execução dos threads
- Processador virtualmente duplicado
  - *n* processadores lógicos
- Objetivo: melhor utilização de recursos
- Intel comercializa como *Hyper-Threading*



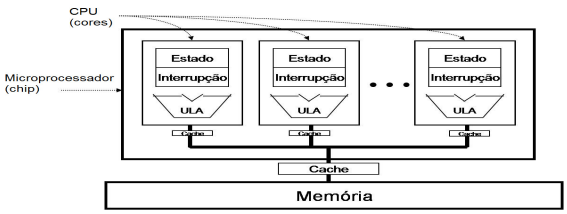
## SMT - Simultaneous multithreading

- Componentes replicados (< 5% da área do chip)
  - Contexto do processo em execução (pilha, regs de controle, etc)
    - Concorrência na execução dos processos
  - Controlador de interrupções
    - Gerência concorrente de interrupções
- Recursos compartilhados entre processos
  - Unidades de execução
  - Cache



## Tecnologia Multicore

- Múltiplos cores (núcleos de execução) integrados em um único chip
- Multiplicação total dos recursos de processamento
- Vantagem: compatibilidade com código existente!

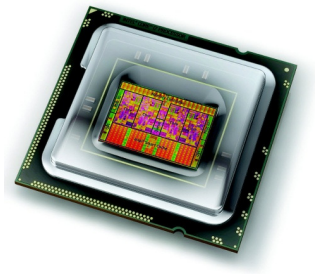



Agradecimentos: Professores Rafael Santos e Gerson Cavalheiro



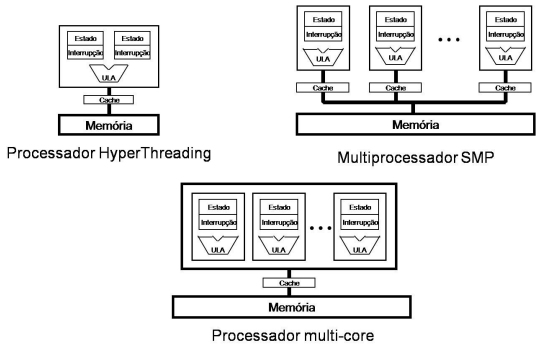
## Tecnologia Multicore

- Exemplo: Intel Core i7
  - Quad core





## Tecnologia Multicore






## Bibliografia


- De Rose, C.; Navaux, P. Arquiteturas Paralelas Editora Sagra-Luzzatto
- Hwang, Kai; Xu - Scalable parallel computing, 1998
- Culler, D.; Singh, J. - Parallel Computer Architecture, 1999
- Seitz et. al. - Myrinet, a gigabit-per-second Local Area Network. IEEE Micro, 15, 1995.
- IEEE:IEEE Standard for Scalable Cherent Interface (SCI). IEEE Standard 1596-1992