Universidade Federal do Rio de Janeiro
Pós-Graduação em Informática
DCC/IM - NCE/UFRJ

Arquiteturas de Sistemas de Processamento Paralelo

Introdução

Gabriel P. Silva

Bibliografia

- **Advanced Computer Architecture**
  - Dezsõ Sima, Terence Fountain, Péter Kacsuk
  - Addison-Wesley, 1997
- **Computer Architecture: A Quantitative Approach**
  - John L. Henessy e David A. Patterson
  - Morgan Kaufman Publishers, 3 edição, 2003
- **Parallel Computer Architecture – A Hardware/Software Approach**
  - David E. Culler e Jaswinder Pal Singh
  - Morgan Kaufman Publishers, 1999

Aplicações Paralelas

Aplicações Paralelas

- Genoma Humano
- Turbulência dos Fluidos
- Dinâmica de Veículos
- Circulação de Oceanos
- Dinâmica de Fluidos Viscosos
- Modelagem de Supercondutores
- Cromodinâmica Quântica
- Visão por Computador
- Farmacêutica
- Biologia Estrutural
- Previsão do Tempo (+ 72 hs)

Aplicações Paralelas

- **Simulando as Correntes Oceânicas**
  - Modelo do clima da terra precisa saber como a atmosfera interage com os oceanos, que ocupam ¾ da superfície da Terra.
  - Estão envolvidos neste estudo diversas forças físicas: efeitos atmosféricos, vento e fricção com o fundo dos oceanos.
  - O oceano é divido em planos e cada plano possui uma grade de pontos igualmente espaçados (uma matriz) com informações como velocidade e pressão, entre outras.
  - Além disto, todos esses planos são simulados para diversos intervalos de tempo, também igualmente espaçados.

Aplicações Paralelas

- **Simulando as Correntes Oceânicas**
  - Para um oceano como o Atlântico, de 2000 km x 2000 km, uma grade de 100 x 100 pontos implica uma distância de 20 km entre os pontos, o que não é uma resolução muito fina.
  - Para simular o comportamento dos oceanos para um período de 5 anos, atualizados a cada 8 horas, serão necessários cerca de 5500 intervalos de tempo.
  - A demanda computacional para alta acurácia é enorme e a necessidade de multiprocessamento é clara.
  - Por sorte a aplicação naturalmente permite bastante concorrência, pois há bastante independência de dados entre as diversas fases da computação.

Aplicações Paralelas

- **Simulando a Evolução das Galáxias**
  - Para este tipo de problema seria necessário calcular a interação gravitacional para cada par de estrelas em diversos intervalos de tempo.
  - Este método de solução tem complexidade $O(n^2)$ o que o tornaria impossível de ser aplicado para os milhões de estrelas de uma galáxia.
  - Mas levando-se em conta que a força gravitacional diminui com o quadrado da distância, pode-se usar um algoritmo de complexidade $O(n \log n)$.

Aplicações Paralelas

- Simulando a Evolução das Galáxias
  - Neste caso grupos de estrelas mais distantes são consideradas como uma única estrela com massa equivalente e situada no centro de massa dessas galáxias.
  - Este algoritmo hierárquico recebe o nome de Barnes-Hut
  - Ampla concorrência existe entre estrelas dentro de um intervalo de tempo, mas como são padrões bastantes irregulares e variantes no tempo, é um desafio explorar esta concorrência em arquiteturas paralelas.

Computadores Paralelos

Classificação dos Processadores (2003)

#	Fabr.	Nome	GFlops	# Proc.
1	NEC	Earth-Simulator, NEC Vector SX6	35860	5120
2	Hewlett-Packard	ASCI Q - AlphaServer SC ES45	13880	8192
2	Linux Networx	MCR Linux Cluster Xeon 2.4 GHz - Quadrics	7634	2304
4	IBM	ASCI White, SP Power3 375 MHz	7304	8192
5	IBM	SP Power3 375 MHz 16 way	7304	6656
6	IBM	xSeries Cluster Xeon 2.4 GHz - Quadrics	6586	1920
7	Fujitsu	PRIMEPOWER HPC2500	5406	2304
8	Hewlett-Packard	SR8000/MPP	4881	1540

Classificação dos Processadores (2005)

#	Fabr.	Nome	GFlops	# Proc.
1	IBM	IBM BlueGene/L	35860	65536
2	IBM	IBM BlueGene/L	91290	40960
3	SGI	SGI Altix 3700	51870	10160
4	NEC	Earth-Simulator, NEC Vector SX6	35860	5120
5	IBM	JS20 Cluster, PPC 970	27910	4800
6	IBM	IBM BlueGene/L	27450	12288
7	Intel	NOW - Itanium2 Tiger4	19940	4096
8	IBM	IBM BlueGene/L	18200	8192

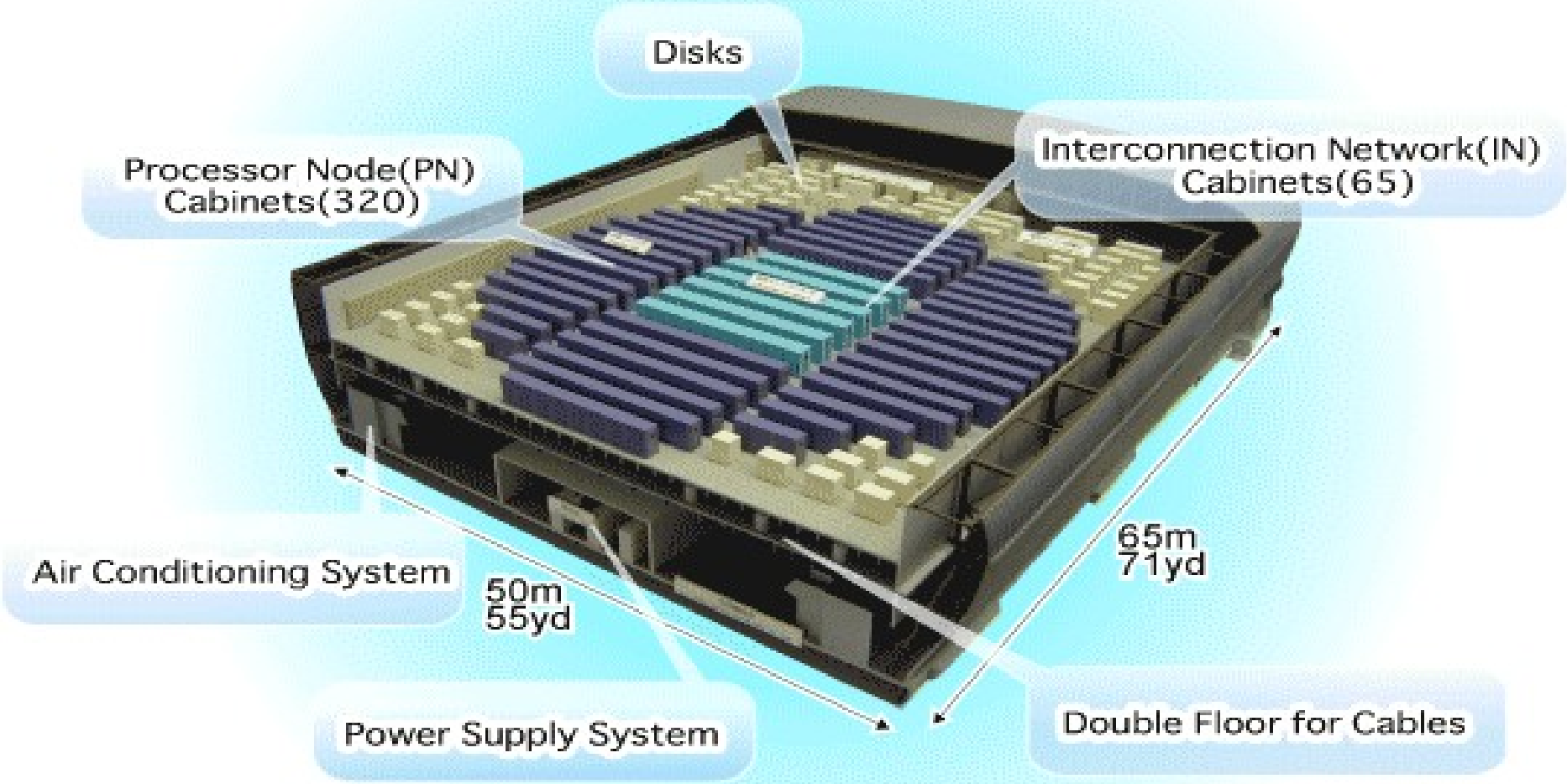
Iniciativa de Computação Estratégica Acelerada (ASCI)

- A Iniciativa de Computação Estratégica Acelerada (ASCI) foi iniciada em 1996 para assegurar que os EUA manteriam a dianteira no desenvolvimento de computadores de alto desempenho. A iniciativa foi realizada por alguns dos maiores laboratórios do Departamento de Energia (DOE): Sandia, Los Alamos, e Lawrence Livermore.
- Três fabricantes (Intel, IBM and SGI) ganharam contratos para desenvolver sistemas que poderiam alcançar velocidades de 10--30 Tflop/s no nível de aplicação por volta de 1999--2001 e cerca de 100 Tflop/s em 2003--2004.
- Um dos objetivos era o uso de componentes "off-the-shelf" de baixo custo e facil aquisição para os sistemas poderem serem facilmente comercializados.

O mais rápido em 2001

- ASCI White

O mais rápido em 2003


Disks
Processor Node(PN)
Cabinets(320)
Interconnection Network(IN)
Cabinets(65)
Air Conditioning System
50m
55yd
65m
71yd
Power Supply System
Double Floor for Cables

O mais rápido em 2005

Conceitos Básicos

Conceitos Básicos

- As arquiteturas paralelas permitem a execução das tarefas em menor tempo, através da execução em paralelo de diversas tarefas.
- O paralelismo pode ser obtido em diversos níveis, com ou sem o uso de linguagens de programação paralelas.
- Arquiteturas de diversos tipos, elaboradas para aplicações específicas, podem ser utilizadas para acelerar a execução dessas aplicações.

Processo

- **Processo:**
  - Um processo é criado para execução de um programa pelo sistema operacional
- **A criação de um processo envolve os seguintes passos:**
  - Preparar o descritor de processo
  - Reservar um espaço de endereçamento
  - Carregar um programa no espaço reservado
  - Passar o descritor de processo para o escalonador
- **Nos modernos S.O.s, um processo pode gerar cópias, chamadas de processos "filhos".**

Threads

- Threads são partes autônomas de código, criadas dentro de um mesmo processo.
- Todas as threads de um processo compartilham os mesmos recursos, em particular o mesmo espaço de endereçamento.
- A custo de criação, comunicação e sincronização entre threads é bem menor que entre processos.

Escalonamento

- Há dois tipos básicos de escalonamento:
  - baseado em processos
  - baseado em threads
- Em qualquer dos casos o controle de atividade é feito por um diagrama com pelo menos três estados básicos:
  - Pronto
  - Executando
  - Aguardando
- As linguagens de programação possuem primitivas específicas para a criação de threads e processos.

Conceitos Básicos

- Execução concorrente está associada a idéia de um servidor atendendo a vários clientes através de uma política de escalonamento no tempo.
- Execução paralela está associada ao modelo de vários servidores atendendo a vários clientes simultaneamente no tempo.
- As linguagens de programação podem então ser classificadas como seqüenciais, concorrentes e paralelas.
- Seqüenciais:
  - C, Pascal, Fortran
- Concorrentes:
  - Ada, Pascal Concorrente, Modula-2, PROLOG Concorrente
- Paralelas:
  - Occam-2, 3L Parallel C, Strand-88

Níveis de paralelismo

- Nível de instrução (granulosidade fina)
  - Arquiteturas Pipelined, Super Escalares e VLIW
- Nível de "thread" (granulosidade média)
  - Arquiteturas SMT
- Nível de processo (granulosidade grossa)
  - Arquiteturas Multiprocessadores e Multicomputadores

Avaliação de Desempenho

Medidas de Desempenho

- **Velocidade: tempo de resposta, vazão e utilização:**
  - **Vazão *(Throughput):* taxa na qual os pedidos são atendidos (servidos) pelo sistema.**
  - **Utilização: fração do tempo em que o recurso permanece ocupado atendendo os pedidos dos usuários**
  - **Tempo de resposta: tempo decorrido entre o pedido e o início/conclusão da realização do serviço (latência).**
- **Confiabilidade**
  - **Probabilidade de erro**
  - **Intervalo entre erros**
- **Disponibilidade**
  - **Duração da falha**
  - **Intervalo entre falhas**

Vazão

- Taxa na qual os pedidos são atendidos (servidos) pelo sistema.
- Exemplos:
  - Sistemas em lotes: jobs por segundo
  - Sistemas interativos: pedidos por segundo
  - CPUs: MIPs ou MFLOPs
  - Redes: pacotes por segundo (pps) ou bits por segundo (bps)
  - Sistemas de Processamento de Transações: Transações por segundo (TPS)

Medidas de Desempenho em Processamento Paralelo

- **Speed-up:**
  - Mede a razão entre o tempo gasto para execução de um algoritmo ou aplicação em um único processador e o tempo gasto na execução com *n* processadores

$$S(n) = T(1)/T(n)$$

- **Eficiência:**

$$E(n) = S(n)/n$$

Medidas de Desempenho

- **Escalabilidade:**
  - **Um sistema é dito *escalável* quando sua eficiência se mantém constante à medida que o número de processadores (*n*) aplicado à solução do problema cresce.**
  - **Se o tamanho do problema é mantido constante e o número de processadores aumenta, o "overhead" de comunicação tende a crescer e a eficiência a diminuir.**
  - **A análise da escalabilidade considera a possibilidade de se aumentar proporcionalmente o tamanho do problema a ser resolvido à medida que *n* cresce de forma a contrabalançar o natural aumento do "overhead" de comunicação quando *n* cresce.**

Técnicas de Avaliação

- Medição
- Modelagem Analítica
- Simulação

Modelagem Analítica

- É uma técnica aproximada, ou seja, aproxima a realidade por um modelo.
- Se o modelo for simples e a aproximação boa, é possível avaliar facilmente compromissos entre alternativas.
- **Teoria das filas**
- **Filas associadas a recursos**
- **Caracterização:**
  - Processo de chegada
  - Processo de atendimento
  - Número de servidores
  - Tamanho máximo da fila
  - Política de atendimento da fila

Simulação

- Simulação de eventos discretos.
- Cada evento (ex.: chegada de usuário, término de serviço, etc.) é tratado quando do instante de sua ocorrência.
- Simula o comportamento de um sistema real.
- Em geral, é possível construir um modelo muito mais próximo da realidade do que com a teoria das filas.

Programas de Avaliação

- **É o processo de comparação entre dois ou mais sistemas através de medições em que são utilizados programas de avaliação (*benchmarks*) como cargas de trabalho *(workloads).* Podem ser de vários tipos:**
  - **Aplicações Reais: compiladores, processadores de texto e imagem são alguns exemplos de programas utilizados. Sofrem de problemas de portabilidade.**
  - **Aplicações Modificadas: E/S removida. "Scripts" são utilizados para simular interatividade.**
  - **Kernels: pequenos trechos de código de programas reais são extraídos de programas reais. Ex: Livermoore loops e Linpack.**

Programas de Avaliação

- ***Benchmarks* de brinquedo:** tipicamente entre 10 e 100 linhas de código para produzir um resultado previamente conhecido. Exemplos são o "Crivo de Eratóstenes", Quicksort e Puzzle. São fáceis de digitar e executam em quase qualquer computador.

- ***Benchmarks* sintéticos:** tentam imitar a freqüência de execução média das instruções esperada em uma aplicação de um determinado tipo, mas não são extratos de programas reais. Whetstone (ponto flutuante) e Dhrystone (inteiro) são os exemplos mais populares.

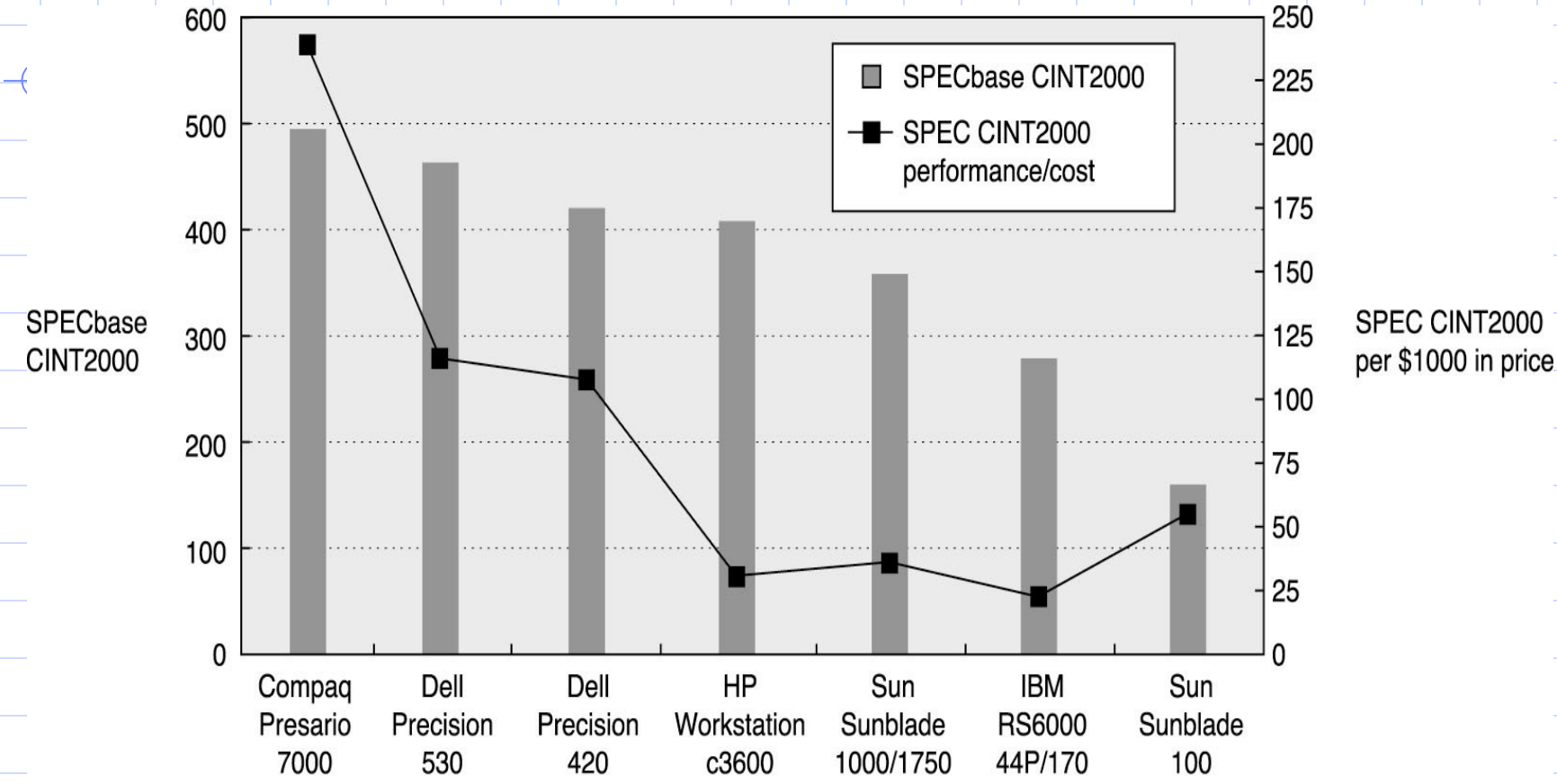
Programas de Avaliação

- **Suítes de Avaliação:** coleção de programas de avaliação que tentam medir o desempenho dos processadores para uma variedade de aplicações.
- O exemplo de maior sucesso é o SPEC, que possui versões inteiras e de ponto flutuante, com edições em 1989, 1992, 1995 e 2000. Estão divididos em duas grandes categorias: inteiros e de ponto flutuante.
- O SPEC CPU2000 é constituído por 11 programas de avaliação inteiros (CINT2000) e 14 de ponto-flutuante (CFP2000).
  - http://www.spec.org
  - http://www.top500.org

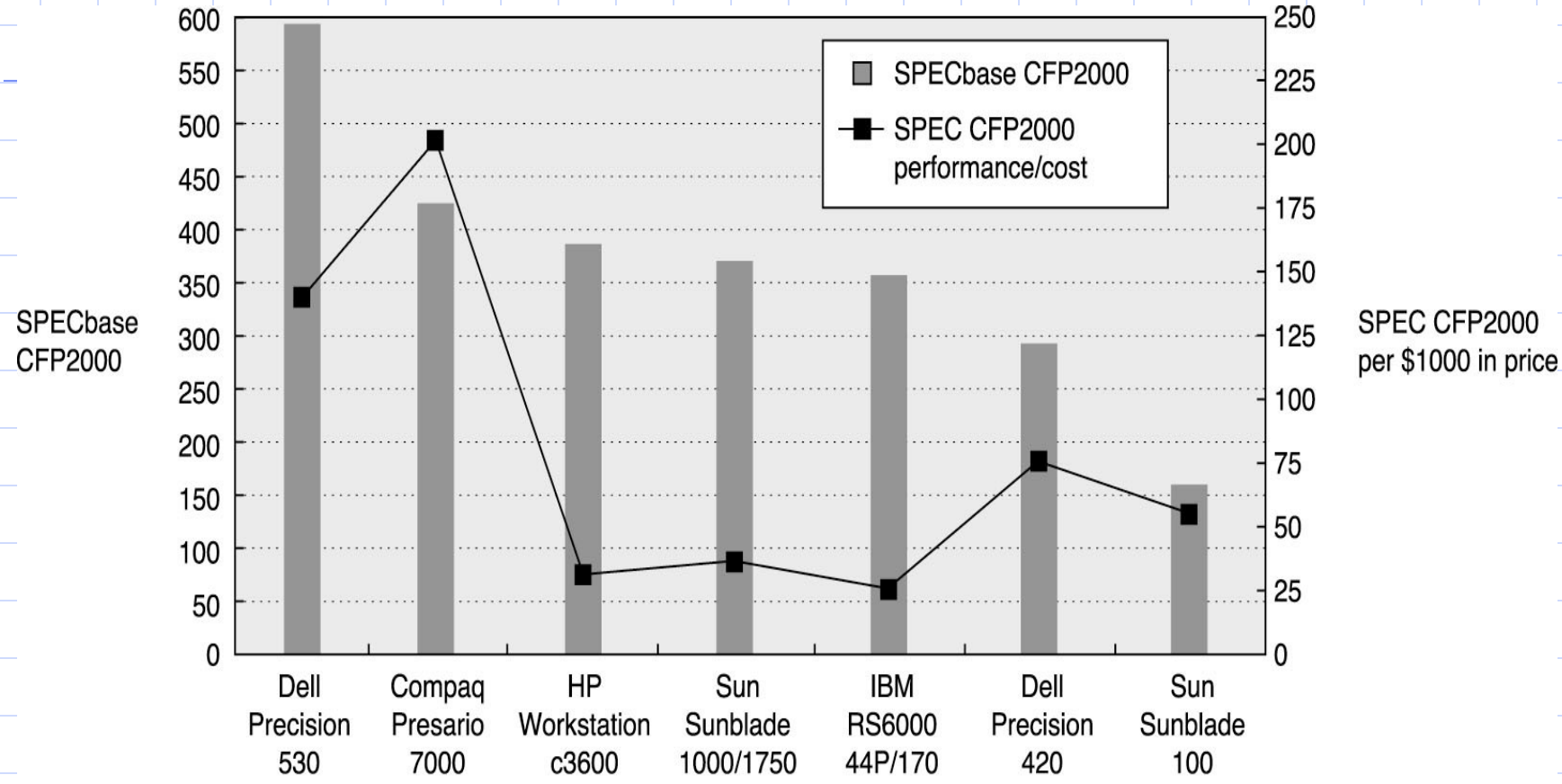
Medição

- Para efetuarmos medições (com os "*benchmarks*" – programas de avaliação) é preciso termos à disposição ao menos um protótipo do sistema;
- Neste cado, normalmente é difícil comparar alternativas;
- O método mais utilizado é compilar os "*benchmarks*" e executá-los no sistema que se deseja avaliar.
- Os programas devem ter características de acordo com os parâmetros que se seja avaliar (CPU, E/S, Gráfico, etc.).

Porgramas de Avaliação



Porgramas de Avaliação



Técnicas de Avaliação

Critério	Modelagem Analítica	Simulação	Medição
Estágio	Qualquer	Qualquer	Protótipo
Tempo Necessário	Pouco	Médio	Variado
Ferramentas	Analistas	Linguagens de Programação	Instrumen-tação
Precisão	Pouca	Moderada	Variada
Avaliação de Compromissos	Fácil	Moderada	Difícil
Custo	Baixo	Médio	Alto
Escalabilidade	Baixa	Média	Alta

Classificação

Paralelismo

- **Níveis de paralelismo :**
  - **nível de instrução (granulosidade fina): arquiteturas pipelined, superescalares e VLIW**
  - **nível de "thread" (granulosidade média): arquiteturas multithreading, SMT**
  - **nível de processo (granulosidade grossa): multiprocessadores e multicomputadores**

Modelos de Programação

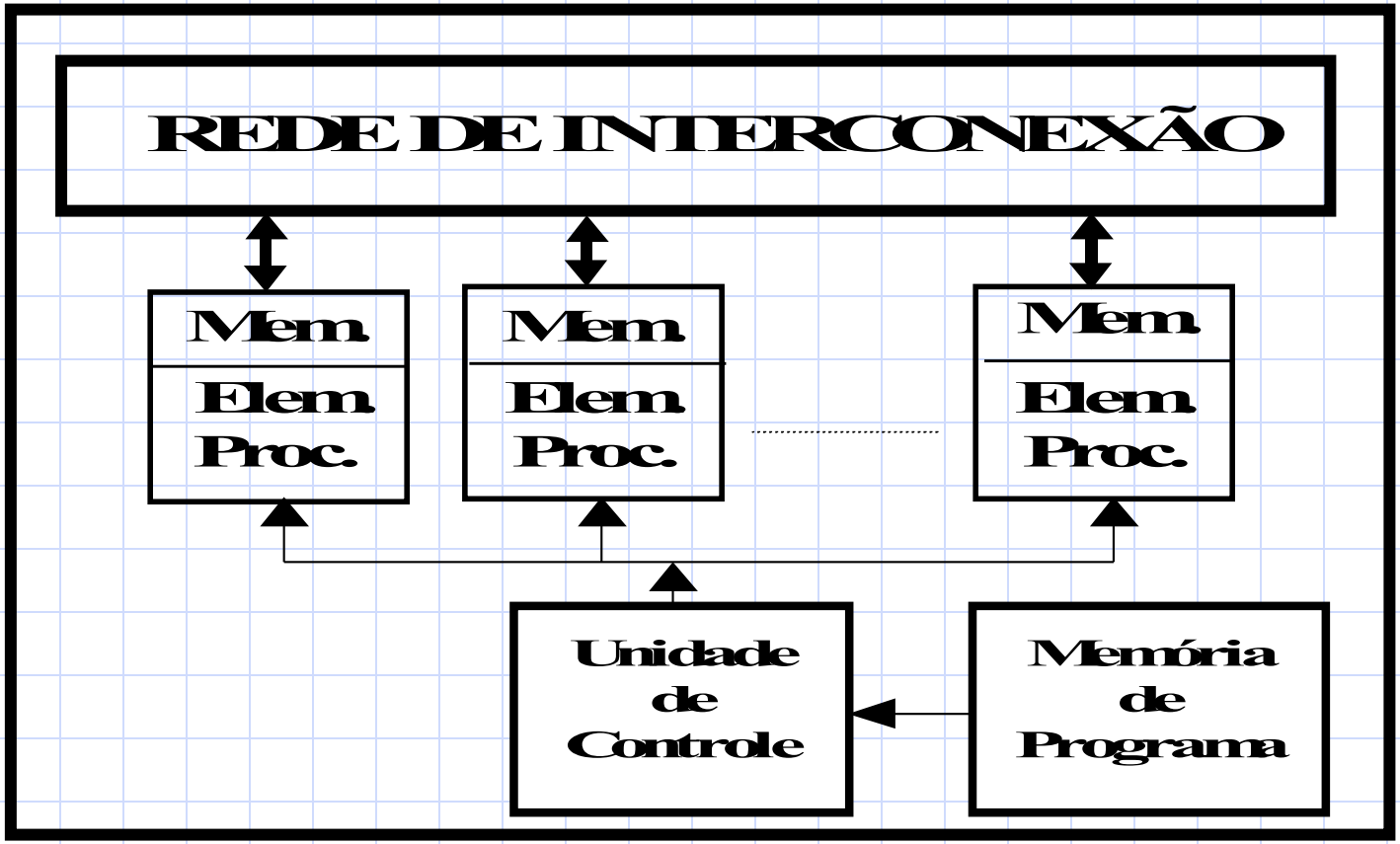
- **Modelos Básicos de Programação:**
  - **Memória Compartilhada:**
    - Todos os processos/threads compartilham de um espaço de endereçamento comum;
    - Comunicação através da memória;
    - Uso de semáforos para sincronização;
    - Linguagens: Pascal e "C" Concorrente.
  - **Memória Distribuída:**
    - Os processos/threads não dispõem de um espaço comum para sincronização e comunicação por troca de mensagens;
    - Linguagens: PVM, MPI, Occam-3.

Paralelismo

- Classificação de Flynn (1966):
  - Single Instruction Stream (SI)
  - Multiple Instruction Streams(MI)
  - Single Data Stream (SD)
  - Multiple Data Streams (MD)
- Categorias de Arquiteturas
  - SISD (Processadores Convencionais)
  - SIMD (Processadores Vetoriais)
  - MIMD (Multiprocessadores)

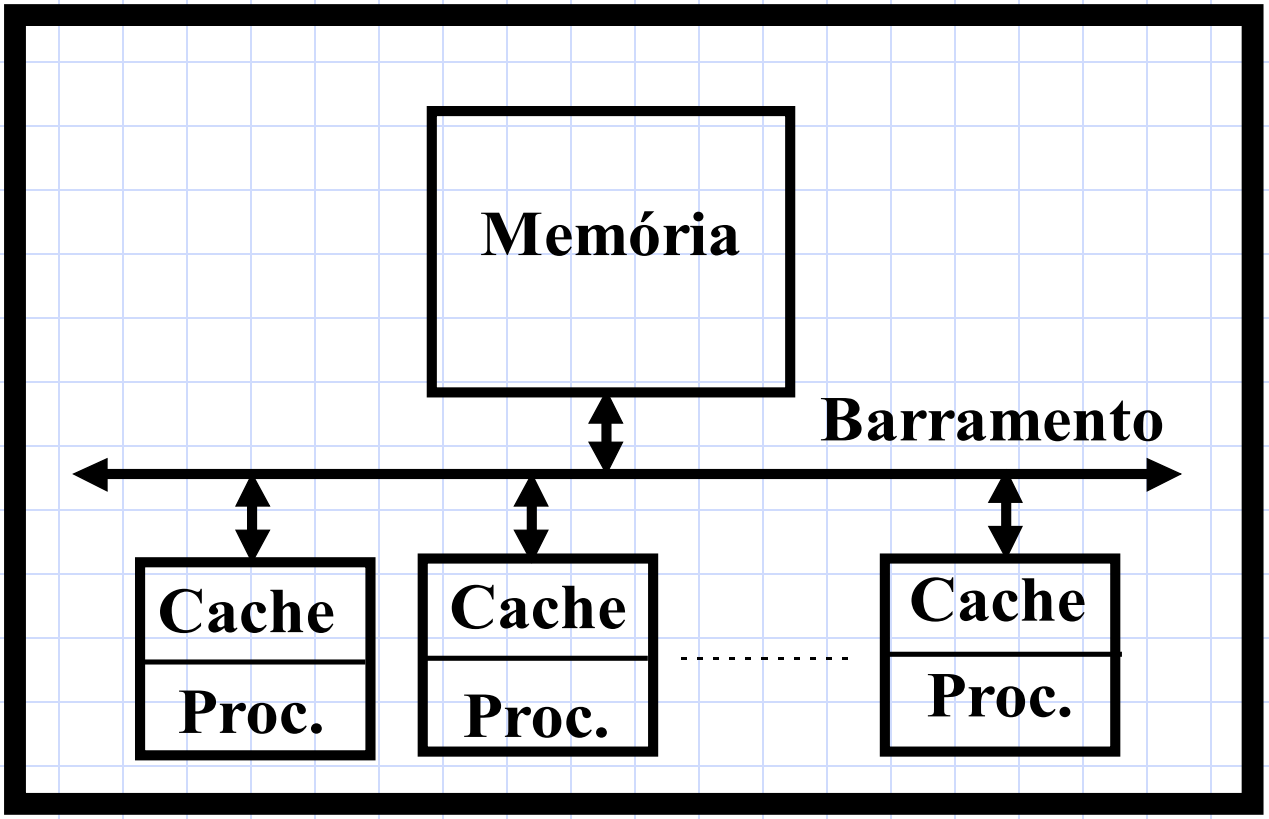
Arquitetura SIMD

- Arquitetura SIMD Básica



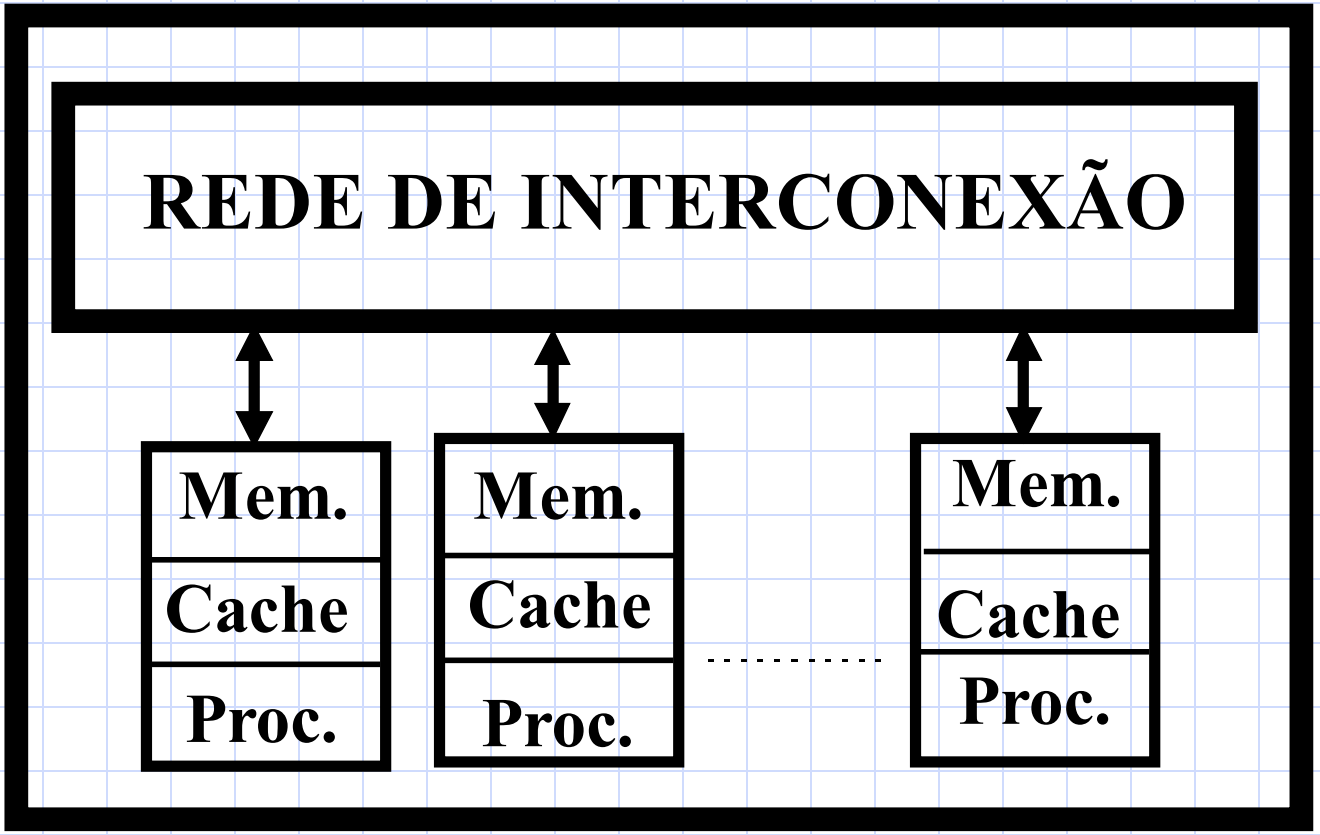
Arquitetura MIMD

- Arquitetura de Memória Centralizada



Arquitetura MIMD

- Arquitetura de Memória Distribuída



Técnicas Básicas de Paralelismo

- **Pipelining**
  - Várias unidades funcionais colocadas em seqüência para realizar uma tarefa.
- **Duplicação de Recursos**
  - Várias unidades funcionais colocadas em paralelo para realizar uma tarefa

Classificação

Tipo de Processador	Pipelining	Duplicação
Vetoriais	X	
Sistólicos	X	
SIMD		X
Pipelined	X	
VLIW		X
Superescalares	X	X
Multithread	X	X
Multicomputadores	X	X
Multiprocessadores		X