

Computação Paralela e Distribuída

Profa. Lúcia Drummond

IC – Instituto de Computação
UFF – Universidade Federal Fluminense

Introdução e Conceitos Básicos

- ⊙ Por que computação paralela e distribuída
- ⊙ Computação de Alto Desempenho
- ⊙ Arquitetura de computadores
- ⊙ Ambientes de programação paralela
- ⊙ Modelos de programação paralela

Por que computação paralela e distribuída?

- ⊙ Sistemas de computadores seqüenciais cada vez mais velozes
 - velocidade de processador
 - memória
 - comunicação com o mundo externo
- ⊙ Quanto mais se tem, mais se quer.....
 - Demanda computacional está aumentando cada vez mais: visualização, base de dados distribuída, simulações, etc.
- ⊙ limites em processamento seqüencial
 - velocidade da luz, termodinâmica

Por que computação paralela e distribuída?

- ⊙ que tal utilizar vários processadores?
- ⊙ dificuldades encontradas
 - mas como?
 - paralelizar uma solução?

Existem vários desafios em Computação Paralela e Distribuída

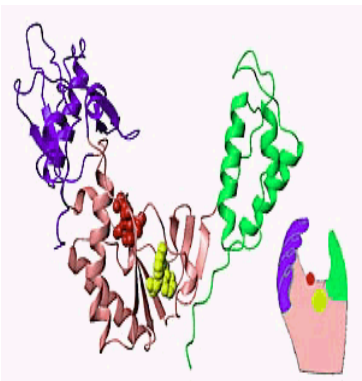
Computação de Alto Desempenho

Os grandes desafios (Levin 1989):

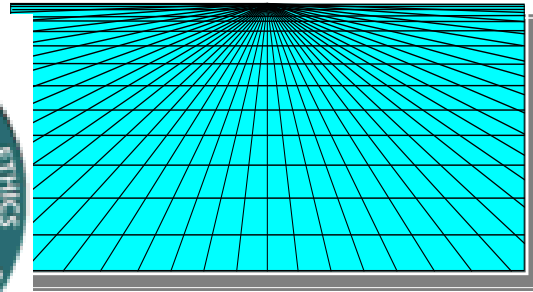
- química quântica, mecânica estatística e física relativista;
- cosmologia e astrofísica;
- dinâmica e turbulência computacional dos fluídos;
- projeto de materiais e supercondutividade;
- biologia, farmacologia, seqüência de genomas, engenharia genética, dobramento de proteínas, atividade enzimática e modelagem de células;
- medicina, modelagem de órgãos e ossos humanos;
- clima global e modelagem do ambiente

Computação de Alto Desempenho

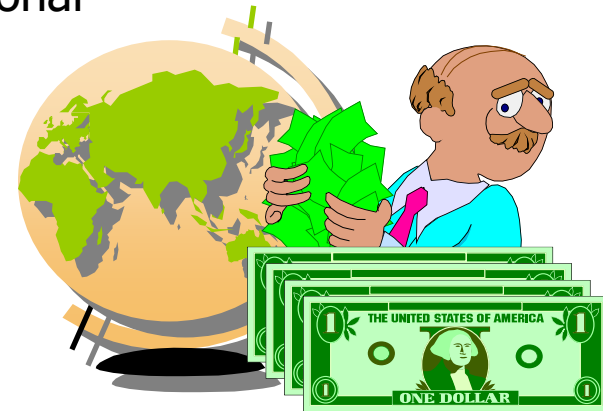
- utilizando modelagem, simulação e análise computacional



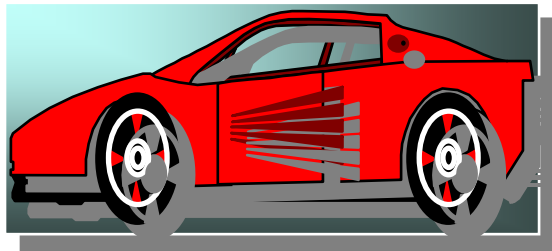
Life Sciences



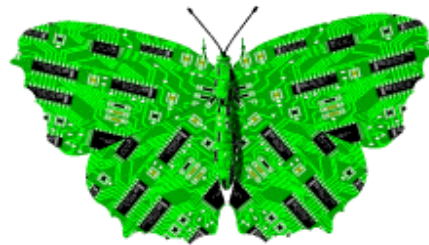
Aerospace



Internet & Ecommerce



CAD/CAM



Digital Biology



Military Applications

Definindo melhor alguns conceitos

- **Concorrência**

- termo mais geral, um programa pode ser constituído por mais de um thread/processo concorrendo por recursos

- **Paralelismo**

- uma aplicação é executada por um conjunto de processadores em um ambiente único (dedicados)

- **Computação distribuída**

- aplicações sendo executadas em plataformas distribuídas

Definindo melhor alguns conceitos

Qualquer que seja o conceito, o que queremos?

- estabelecer a solução do problema
- lidar com recursos independentes
- aumentar desempenho e capacidade de memória
- fazer com que usuários e computadores trabalhem em espírito de colaboração

O que paralelizar?

- ◉ Concorrência pode estar em diferentes níveis de sistemas computacionais atuais
 - hardware
 - Sistema Operacional
 - Aplicação

- ◉ As principais questões que são focadas são
 - Desempenho
 - Corretude
 - possibilidade de explorar o paralelismo

Por que paralelizar?

Aplicação Paralela

- várias tarefas
- vários processadores
 - redução no tempo total de execução

Modelos de Programação Paralela

- ⊙ Criação e gerenciamento de processos
 - estático ou dinâmico
- ⊙ Comunicação
 - **memória compartilhada**
 - visão de um único espaço de endereçamento global
 - **memória distribuída**
 - troca explícita de mensagens

Modelos de Programação Paralela

- ⊙ Expressão de Paralelismo: Paradigmas
 - SPMD
 - MPMD

- ⊙ Metas
 - aumento no desempenho
 - maior eficiência

Objetivos

- ◉ Visão geral
 - arquitetura de computadores
 - ambientes de programação paralela
 - modelos de programação paralela
- ◉ Motivar ⇒ Sistemas de Alto Desempenho

Arquitetura de Computadores

Classificação de Computadores

- Computadores Convencionais
- Memória Centralizada
- Memória Distribuída

Arquitetura de Computadores

- ⊙ Sistema Paralelo
 - vários processadores
 - vários módulos de memória
 - comunicação: estruturas de interconexão

Plataforma de Execução Paralela

Conectividade \Rightarrow rede de interconexão

Heterogeneidade \Rightarrow hardware e software distintos

Compartilhamento \Rightarrow utilização de recursos

Imagem do sistema \Rightarrow como usuário o percebe

Escalabilidade \Rightarrow + nós > desempenho/eficiência

Classificação de Sistemas Paralelos

- ◉ Proposta por Flynn
 - quantidade de instruções e dados processados em um determinado momento

SISD (single instruction single data)

- Um contador de programa
- Computadores seqüenciais

SIMD (single instruction multiple data)

- Um contador de programa, uma instrução executada por diversos processadores sobre diferentes dados
- Computadores

Classificação de Sistemas Paralelos

- ◉ Proposta por Flynn

MISD (multiple instructions single data)

- Não aplicável

MIMD (multiple instructions multiple data)

- Vários contadores de programa
- Diferentes dados
- Os vários computadores paralelos e distribuídos atuais

Plataforma de Execução Paralela

⦿ Diferentes plataformas do **MIMD** de acordo com os seguintes critérios

- espaço de endereçamento
- mecanismo de comunicação

⦿ Podem ser agrupadas em quatro grupos

SMPs (*Symmetric MultiProcessors*)

MPPs (*Massively Parallel Processors*)

Cluster ou **NOWs** (*Network Of Workstations*)

Grades Computacionais

SMPs

◉ SMPs ou Multiprocessadores

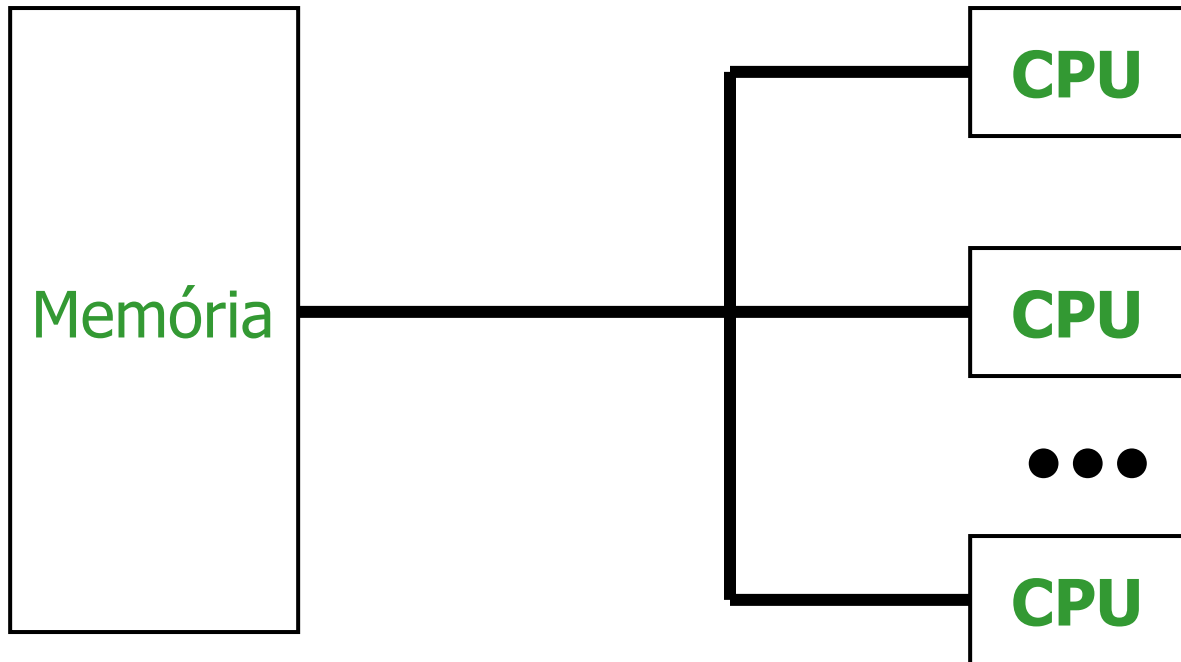
- único espaço de endereçamento lógico
 - mecanismo de hardware (memória centralizada)
- comunicação \Rightarrow espaço de endereçamento compartilhado
 - operações de *loads* e *stores*
 - Acesso a memória é realizada através de leitura (load) e escrita (store), caracterizando desta forma, a comunicação entre processadores

SMPs

- ⊙ Sistema homogêneo
- ⊙ Compartilhamento
 - Compartilhamento total da mesma memória
- ⊙ Uma única cópia do Sistema Operacional
- ⊙ Imagem única do sistema
- ⊙ Excelente conectividade
 - fortemente acoplados
- ⊙ Não escalável
- ⊙ Exemplos:
 - Sun HPC 10000 (StarFire), SGI Altix, SGI Origin, IBM pSeries, Compac AlphaServer

SMPs

Multiprocessadores



MPPs (Multicomputadores)

- ⊙ Diferem quanto a implementação física
- ⊙ Módulos ou elementos de processamento contendo:
 - múltiplos processadores com memória privativa
 - computadores completos
- ⊙ Espaço de endereçamento
 - não compartilhado - **memória distribuída**
- ⊙ Comunicação
 - troca de mensagens
- ⊙ Rede de interconexão
 - diferentes topologias
- ⊙ Fracamente acoplados
- ⊙ Escaláveis

MPPs

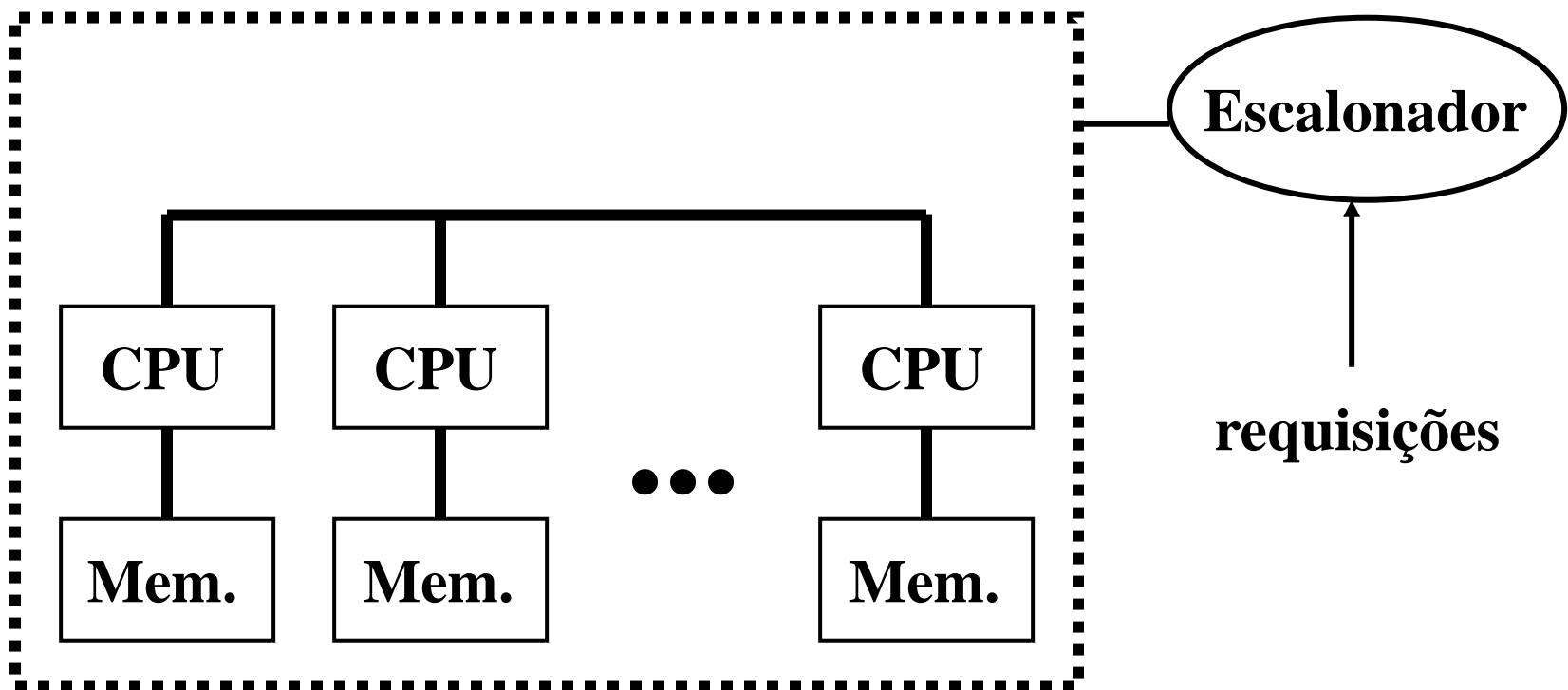
- ⦿ Sistema homogêneo ou heterogêneo
- ⦿ Interconexão: redes dedicadas e rápidas
- ⦿ Cada nó executa sua própria cópia do Sistema Operacional
- ⦿ Imagem única do sistema
 - visibilidade dos mesmos sistemas de arquivo
- ⦿ Um escalonador de tarefas
 - partições diferentes para aplicações diferentes

MPPs

- ⦿ Partições dedicadas a cada aplicação
- ⦿ Aplicações não compartilham recursos
 - Pode ocorrer que uma aplicação permaneça em estado de espera
- ⦿ Exemplos:
 - Cray T3E, IBM SP2s, *clusters* montados pelo próprio usuário, com propósito de ser um MPP

MPPs

- **Multicomputadores**



Cluster de computadores ou NOWs

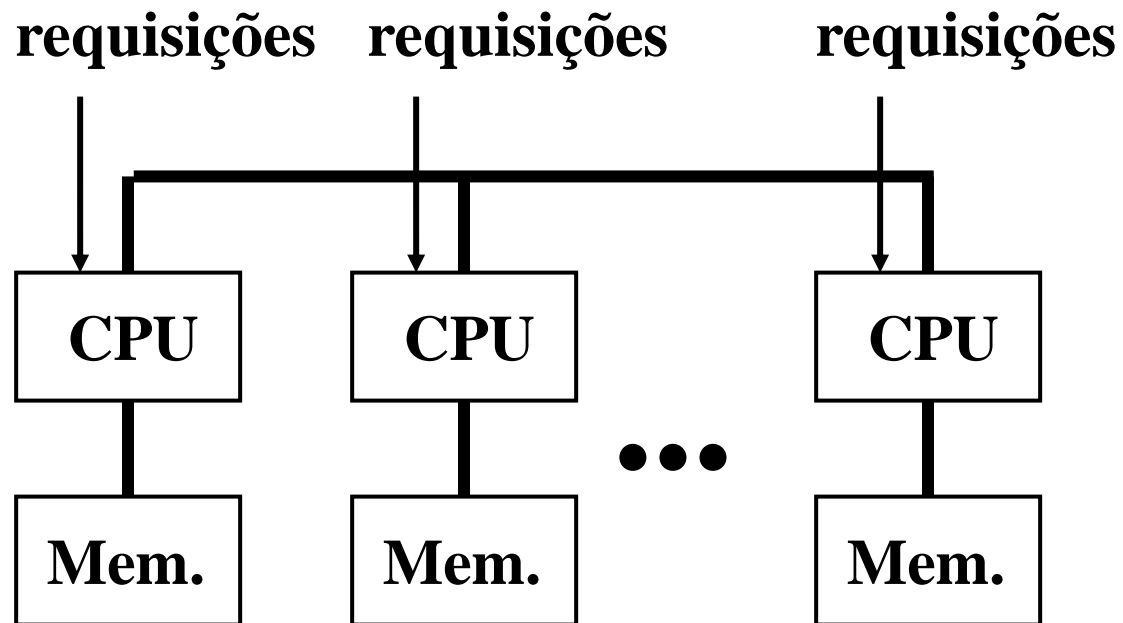
- ⊙ Conjunto de estações de trabalho ou PCs
- ⊙ Interconexão: redes locais
- ⊙ Nós: elementos de processamento = processador + memória
- ⊙ Diferenças em relação a MPPs:
 - não existe um escalonador centralizado
 - redes de interconexão tendem a ser mais lentas

Cluster de computadores ou NOWs

- ⊙ Resultado das diferenças:
 - Cada nó tem seu próprio escalonador local
 - Compartilhamento de recursos ⇒ sem partição dedicada a uma aplicação
 - Aplicação ⇒ deve considerar impacto no desempenho
 - ⇒ não tem o sistema dedicado

 - Possibilidade de compor um sistema de alto desempenho e um baixo custo (principalmente quando comparados com MPPs).

Cluster ou NOWs



Grades Computacionais (*Computational Grids*)

- ⊙ Utilização de computadores
 - independentes
 - geograficamente distantes

- ⊙ Diferenças: *clusters* X grades
 - heterogeneidade de recursos
 - alta dispersão geográfica (escala mundial)
 - compartilhamento
 - múltiplos domínios administrativos
 - controle totalmente distribuído

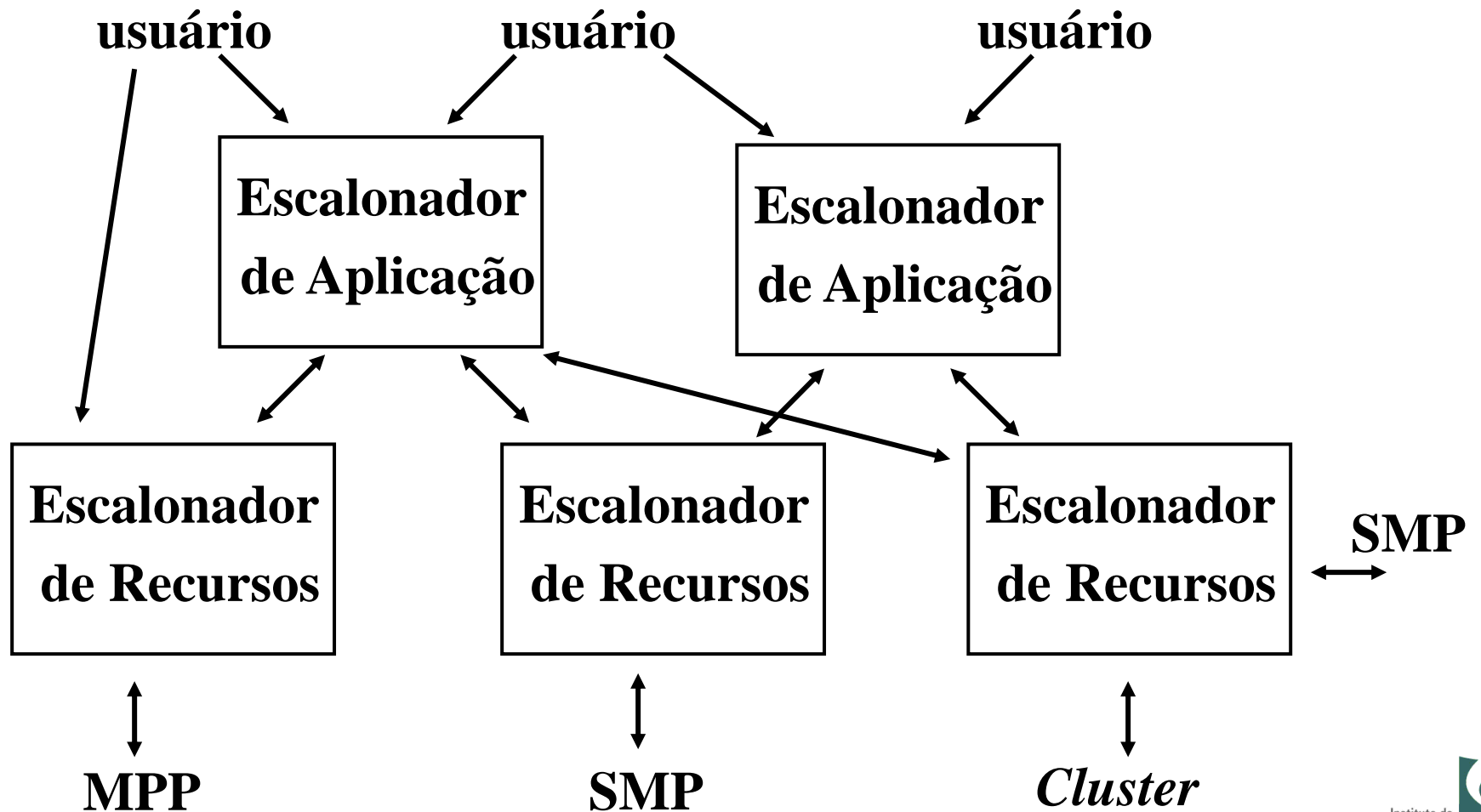
Grades Computacionais

- ⊙ Componentes
 - PCs, SMPs, MPPs, *clusters*
 - controlados por diferentes entidades ⇒ diversos domínios administrativos
- ⊙ Não têm uma imagem única do sistema a princípio
 - Vários projetos tem proposto o desenvolvimento de middlewares de gerenciamento ⇒ camada entre a infra-estrutura e as aplicações a serem executadas na grade computacional
- ⊙ Aplicação deve estar preparada para:
 - Dinamismo
 - Variedade de plataformas
 - Tolerar falhas

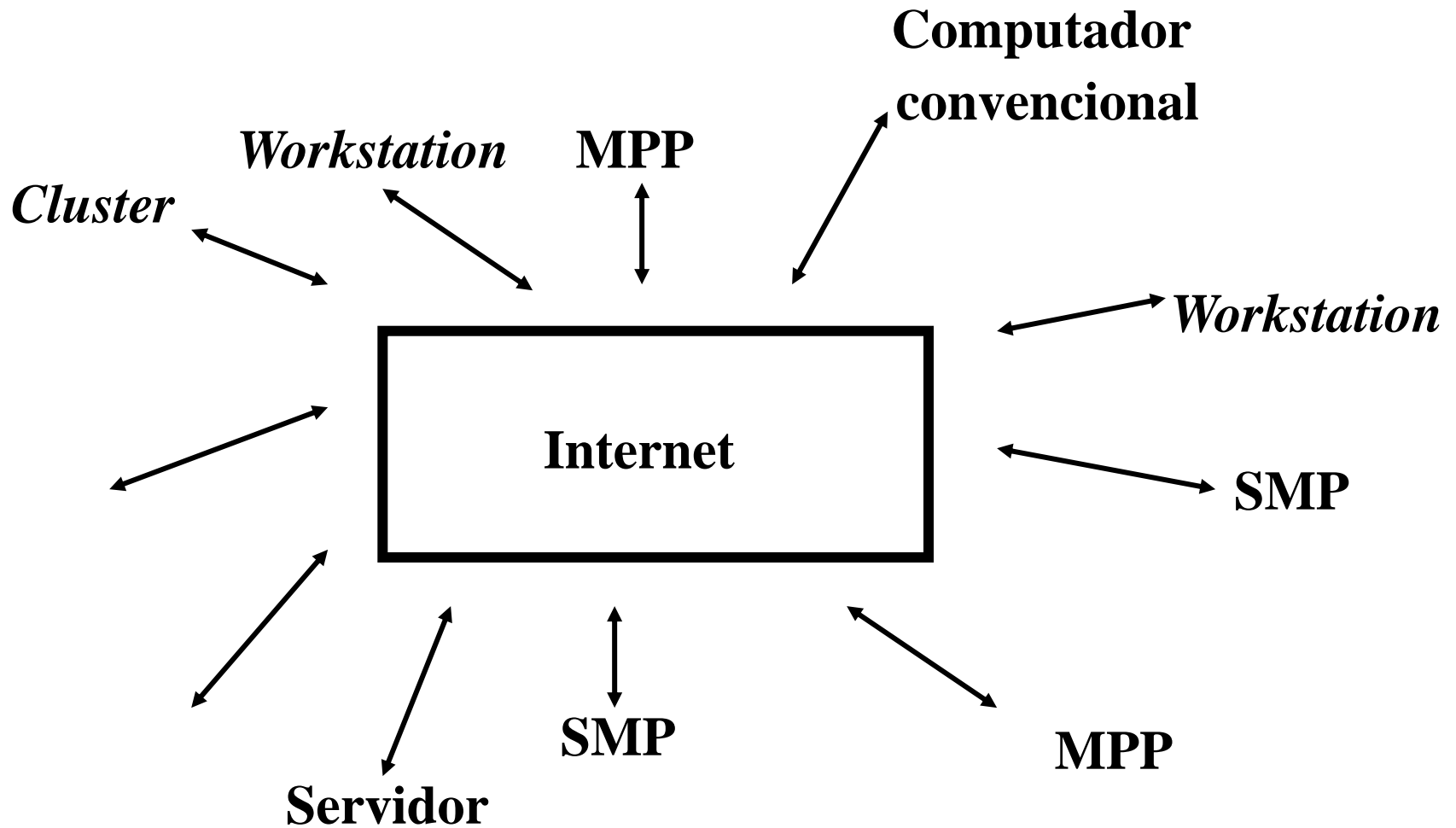
Grades Computacionais

- ⊙ Sistema não dedicado e diferentes plataformas
 - Usuários da grades devem obter autorização e certificação para acesso aos recursos disponíveis na grade computacional
- ⊙ Falhas nos recursos tanto de processamento como comunicação são mais freqüentes que as outras plataformas paralelas
 - Mecanismos de tolerância a falhas devem tornar essas flutuações do ambiente transparente ao usuário
- ⊙ Para utilização eficiente da grade computacional
 - Gerenciamento da execução da aplicação através de políticas de escalonamento da aplicação ou balanceamento de carga
 - Escalonamento durante a execução da aplicação se faz necessário devido as variações de carga dos recursos da grade

Grades Computacionais



Grades Computacionais



Resumo

⦿ Plataformas de Execução Paralela

Características	SMPs	MPPs	NOWs	Grids
Conectividade	excelente	muito boa	boa	média/ruim
Heterogeneidade	nula	baixa	média	alta
Compartilhamento	não	não	sim	sim
Imagem do Sistema	única	comum	comum	múltipla
Escalabilidade	10	1.000	1.000	100.000

Top500 Supercomputer (atualizada)

	Site	Computer	Procs	Year	R _{max}	R _{peak}
1	DOE/NNSA/LLNL United States	BlueGene/L - eServer Blue Gene Solution IBM	131072	2005	280600	367000
2	NNSA/Sandia National Laboratories United States	Red Storm - Sandia/ Cray Red Storm, Opteron 2.4 GHz dual Cray Inc.	26544	2006	101400	127411
3	IBM Thomas J. Watson Research Center United States	BGW - eServer Blue Gene Solution IBM	40960	2005	91290	114688
4	DOE/NNSA/LLNL United States	ASC Purple - eServer pSeries p5 575 1.9 GHz IBM	12208	2006	75760	92781
5	Barcelona Supercomputing Center Spain	MareNostrum - BladeCenter JS21 Cluster, PPC 970, 2.3 GHz, Myrinet IBM	10240	2006	62630	94208
6	NNSA/Sandia National Laboratories United States	Thunderbird - PowerEdge 1850, 3.6 GHz, Infiniband Dell	9024	2006	53000	64972.8
7	Commissariat a l'Energie Atomique (CEA) France	Tera00 - NovaScale 5160, Itanium2 1.6 GHz, Quadrics Bull SA	9968	2006	52840	63795.2
8	NASA/Ames Research Center/NAS United States	Columbia - SGI Altix 1.5 GHz, Voltaire Infiniband SGI	10160	2004	51870	60960
9	GSIC Center, Tokyo Institute of Technology Japan	STSUBAME Grid Cluster - Sun Fire x4600 Cluster, Opteron 2.4/2.6 GHz and ClearSpeed Accelerator, Infiniband NEC/Sun	11088	2006	47380	82124.8
10	Oak Ridge National Laboratory United States	Jaguar - Cray XT3, 2.4 GHz Cray Inc.	10424	2006	43480	54204.8

R_{max} Maximal LINPACK performance achieved

R_{peak} Theoretical peak performance

GFlops Computação

Top500 Supercomputer (Máquinas Brasileiras)

	Site	Computer	Procs	Year	R _{max}	R _{peak}
273	Petroleum Company (C) Brazil	xSeries Cluster Xeon 3.06 GHz - Gig-E IBM	1024	2004	3755	6266.88
275	PETROBRAS Brazil	Rbwr1 Cluster platform 3000 DL140G3 Xeon 3.06 GHz GigEthernet Hewlett-Packard	1300	2004	3739	7956
363	University of San Paulo Brazil	BladeCenter JS21 Cluster, PPC970, 2.5 GHz, Myrinet IBM	448	2006	3182.38	4480
418	PETROBRAS Brazil	bw7 - Cluster platform 3000 DL140G3 Xeon 3.06 GHz GigEthernet Hewlett-Packard	1008	2004	2992	6169

Rmax Maximal LINPACK performance achieved

Rpeak Theoretical peak performance

GFlops  Instituto de Computação

Computação em Cluster

- Um conjunto de computadores (PCs)
- não necessariamente iguais → heterogeneidade
- Filosofia de imagem única
- Conectadas por uma rede local

Para atingir tais objetivos, necessidade de uma camada de software ou *middleware*

Computação em Grid

- Computação em Cluster foi estendido para computação ao longo dos sites distribuídos geograficamente conectados por redes metropolitanas

Grid Computing

- Heterogêneos
- Compartilhados
- Aspectos que devem ser tratados
- Segurança
- Falhas de recursos
- Gerenciamento da execução de várias aplicações

Computação em Grid

O sonho do cientista (The Grid Vision)

- ⊙ Computação em Grid adota tanto o nome quanto o conceito semelhantes aqueles da Rede de Potência Elétrica para capturar a noção ou a visão de:
 - Oferecer desempenho computacional eficientemente;
 - De acordo com a demanda;
 - A um custo razoável;
 - Para qualquer um que precisar.
- ⊙ O sucesso da computação em grid depende da comunidade de pesquisadores
 - A possibilidade de construir tal ambiente (hardware e software)
 - Necessidade de atingir seus objetivos.

Computação em Grid



Computação em Grid

- *Grid middlewares*: tem como objetivo facilitar a utilização de um ambiente grid
 - APIs para isolar usuários ou programas da complexidade deste ambiente
 - Gerenciar esses sistemas automaticamente e eficientemente para executar aplicações no ambiente grid (grid-enabled applications)

E as aplicações não habilitadas a execução em ambiente grids?

Computação em Grid

Como o usuário (dono da aplicação) escolhe?

- ⊙ Vários *middlewares* existem, qual o mais apropriado?
- ⊙ Vários estão ainda sendo desenvolvidos
- ⊙ Não há a garantia de suporte
- ⊙ Pouca comparação entre os *middlewares*, por exemplo, desempenho, grau de intrusão.
- ⊙ É difícil encontrar grids com o mesmo tipo de software instalado