

INE 5645

Programação Paralela e Distribuída

Professor:

Frank Siqueira
INE – UFSC
frank@inf.ufsc.br

Conteúdo Programático

1. Introdução
2. Programação Paralela
3. Controle de Concorrência
4. Programação Distribuída
5. Comunicação entre Processos

Recursos Computacionais

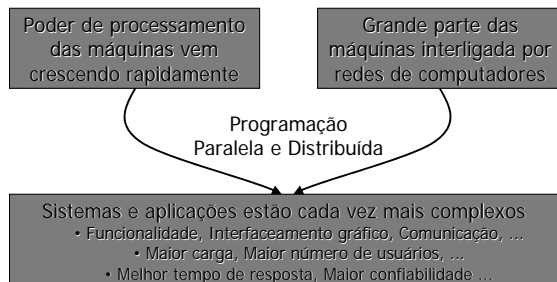
- Software
J2SE SDK
- Página da disciplina
<http://www.inf.ufsc.br/~frank/INE5645>
- Lista de e-mails
ine5645-0538@inf.ufsc.br

Unidade 1

Introdução à Programação Paralela e Distribuída

- Programação Paralela e Distribuída
- Vantagens e Dificuldades
- Plataformas de Execução
- Suporte Computacional

Panorama Atual



Programação Paralela

- O que é?
 - Consiste em executar simultaneamente várias partes de uma mesma aplicação
 - Tornou-se possível a partir do desenvolvimento de sistemas operacionais multi-tarefa, multi-thread e paralelos
 - Aplicações são executadas paralelamente:
 - Em um mesmo processador
 - Em uma máquina multiprocessada
 - Em um grupo de máquinas interligadas que se comporta como uma só máquina
- 6

Programação Distribuída

- O que é?
 - Consiste em executar aplicações cooperantes em máquinas diferentes
 - Tornou-se possível a partir da popularização das redes de computadores
- Aplicações são executadas em máquinas diferentes interligadas por uma rede
 - Intranets
 - Internet
 - Outras redes públicas ou privadas

7

Diferenças

- Acoplamento
 - Sistemas paralelos são fortemente acoplados: compartilham hardware ou se comunicam através de um barramento de alta velocidade
 - Sistemas distribuídos são fracamente acoplados
- Previsibilidade
 - O comportamento de sistemas paralelos é mais previsível; já os sistemas distribuídos são mais imprevisíveis devido ao uso da rede e a falhas

8

Diferenças

- Influência do Tempo
 - Sistemas distribuídos são bastante influenciados pelo tempo de comunicação pela rede; em geral não há uma referência de tempo global
 - Em sistemas paralelos o tempo de troca de mensagens pode ser desconsiderado
- Controle
 - Em geral em sistemas paralelos se tem o controle de todos os recursos computacionais; já os sistemas distribuídos tendem a empregar também recursos de terceiros

9

Vantagens

- Usam melhor o poder de processamento
- Apresentam um melhor desempenho
- Permitem compartilhar dados e recursos
- Podem apresentar maior confiabilidade
- Permitem reutilizar serviços já disponíveis
- Atendem um maior número de usuários
- ...

10

Dificuldades

- Desenvolver, gerenciar e manter o sistema
- Controlar o acesso concorrente a dados e a recursos compartilhados
- Evitar que falhas de máquinas ou da rede comprometam o funcionamento do sistema
- Garantir a segurança do sistema e o sigilo dos dados trocados entre máquinas
- Lidar com a heterogeneidade do ambiente
- ...

11

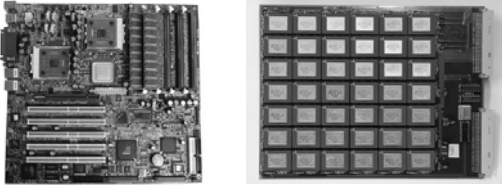
Plataformas de Execução

- Um S.O. multitarefa permite simular o paralelismo em um único processador, alternando a execução de processos
- Um processador com núcleo múltiplo permite paralelismo real entre processos, executando múltiplas instruções por ciclo



Plataformas de Execução

- Uma Placa-Mãe Multiprocessador permite que cada processador execute um processo



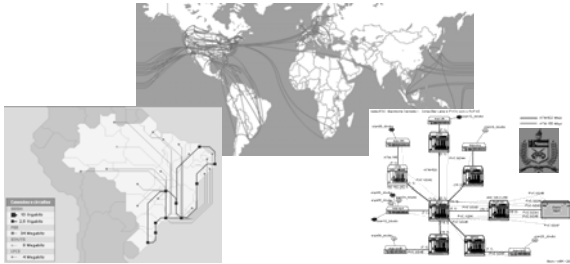
Plataformas de Execução

- Um Cluster é uma solução de baixo custo para processamento de alto desempenho



Plataformas de Execução

- Computação Distribuída é possível em redes, como numa Intranet e na Internet



Suporte Computacional

- Suportes para Computação Paralela e Distribuída devem fornecer:
 - Mecanismos para execução paralela ou distribuída de programas
 - Mecanismos para controle de concorrência
 - Mecanismos para comunicação entre processos / threads paralelos / distribuídos
 - Ferramentas e mecanismos para desenvolvimento, testes, gerenciamento, controle, segurança, tolerância a faltas, etc.

16

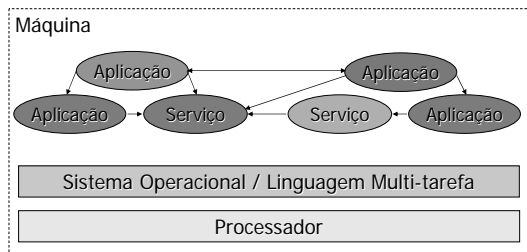
Suporte Computacional

- Suporte para Computação Paralela
 - Sistemas Operacionais Multi-Tarefa: permitem a troca de contexto entre processos / threads. Ex.: Windows, Linux, Solaris, HP-UX, AIX, etc.
 - Linguagens Multi-Tarefa: permitem escrever programas paralelos. Ex.: Pascal FC, Java, etc.
 - Sistemas Operacionais Paralelos: permitem usar vários processadores em uma máquina. Ex.: Linux, Solaris, Windows, etc.
 - Suportes para Programação Paralela: permitem criar uma máquina paralela virtual. Ex.: PVM

17

Suporte Computacional

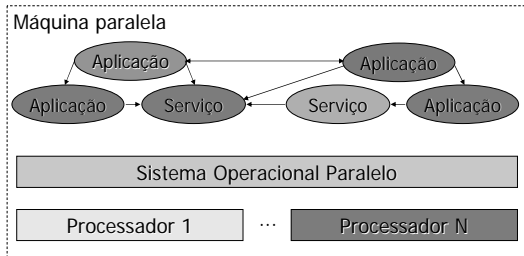
- Sist. Operacionais / Linguagens Multi-tarefa



18

Suporte Computacional

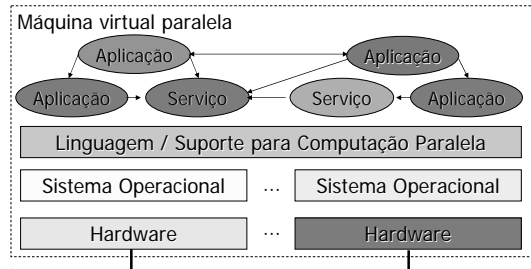
■ Sistemas Operacionais Paralelos



19

Suporte Computacional

■ Suportes para Computação Paralela



20

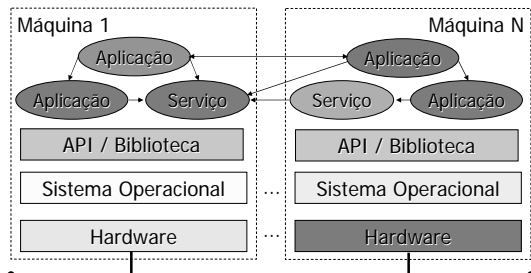
Suporte Computacional

- Suporte para Computação Distribuída
 - APIs e Bibliotecas: fornecem rotinas para comunicação entre processos
Ex.: UNIX Sockets, WinSock, java.net, etc.
 - Middleware para Programação Distribuída: fornece suporte para criar / executar programas distribuídos. Ex.: CORBA, COM, etc.
 - Servidores de Aplicação: permitem o acesso a aplicações via rede. Ex.: Tomcat, JBoss, etc.
 - Linguagens e sistemas operacionais distribuídos caíram em desuso por não suportarem heterogeneidade de ambiente

21

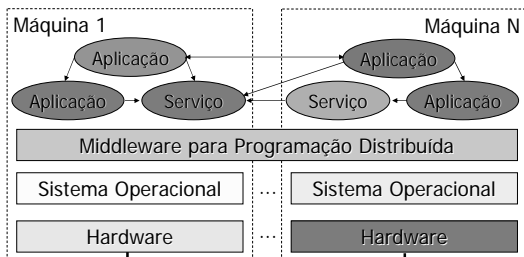
Suporte Computacional

■ APIs e Bibliotecas para Comp. Distribuída



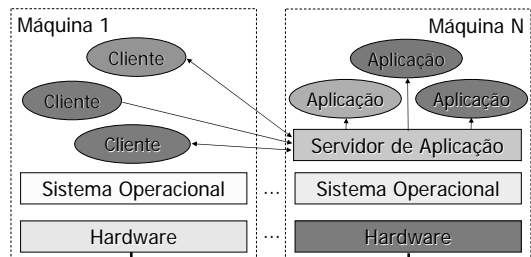
Suporte Computacional

■ Middleware para Programação Distribuída



Suporte Computacional

■ Servidor de Aplicação



24