Universidade Estadual de Mato Grosso do Sul
Curso de Ciência da Computação
Disciplina de Algoritmos Paralelos e Distribuídos

Aula 4 – Processos: Clientes, Servidores e Migração

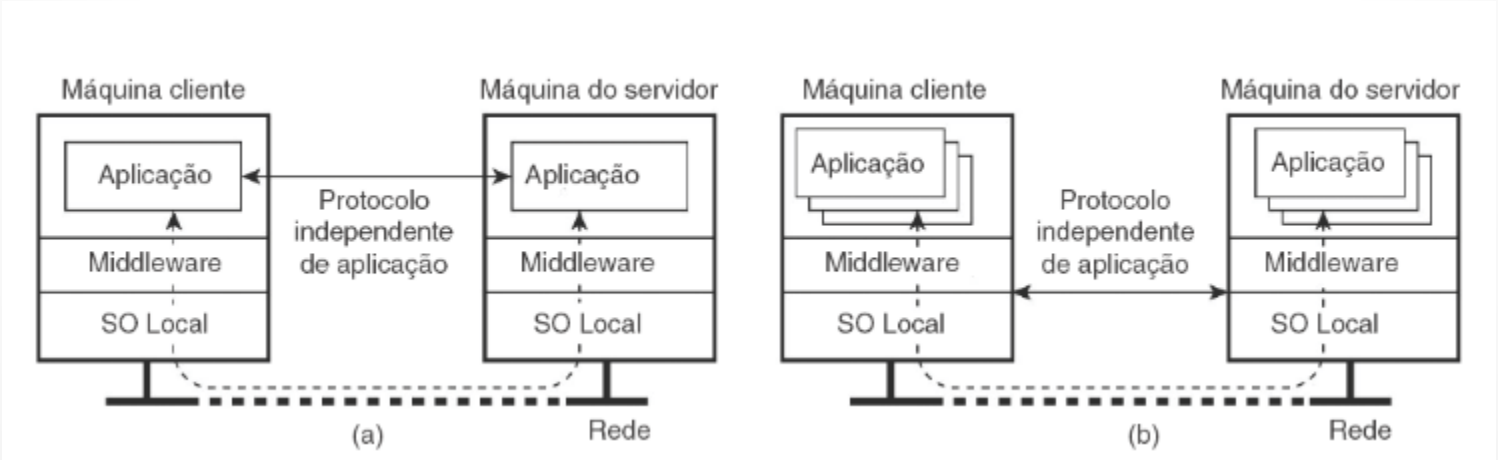
Clientes

- A tarefa das máquinas clientes é proporcionar aos usuários meios de interagir com servidores remotos.
- Há dois modos:
  1. Para cada serviço remoto, a máquina cliente terá um servidor separado com o qual possa estabelecer contato. Algumas operações são feitas no próprio cliente.

  2. Ou então, fornecer apenas uma interface de usuário conveniente. A máquina cliente é usada somente como um terminal sem nenhuma necessidade de armazenamento.

Clientes

- A) Aplicação em rede com seu próprio protocolo.
- B) Solução geral para permitir acesso a aplicações remotas.

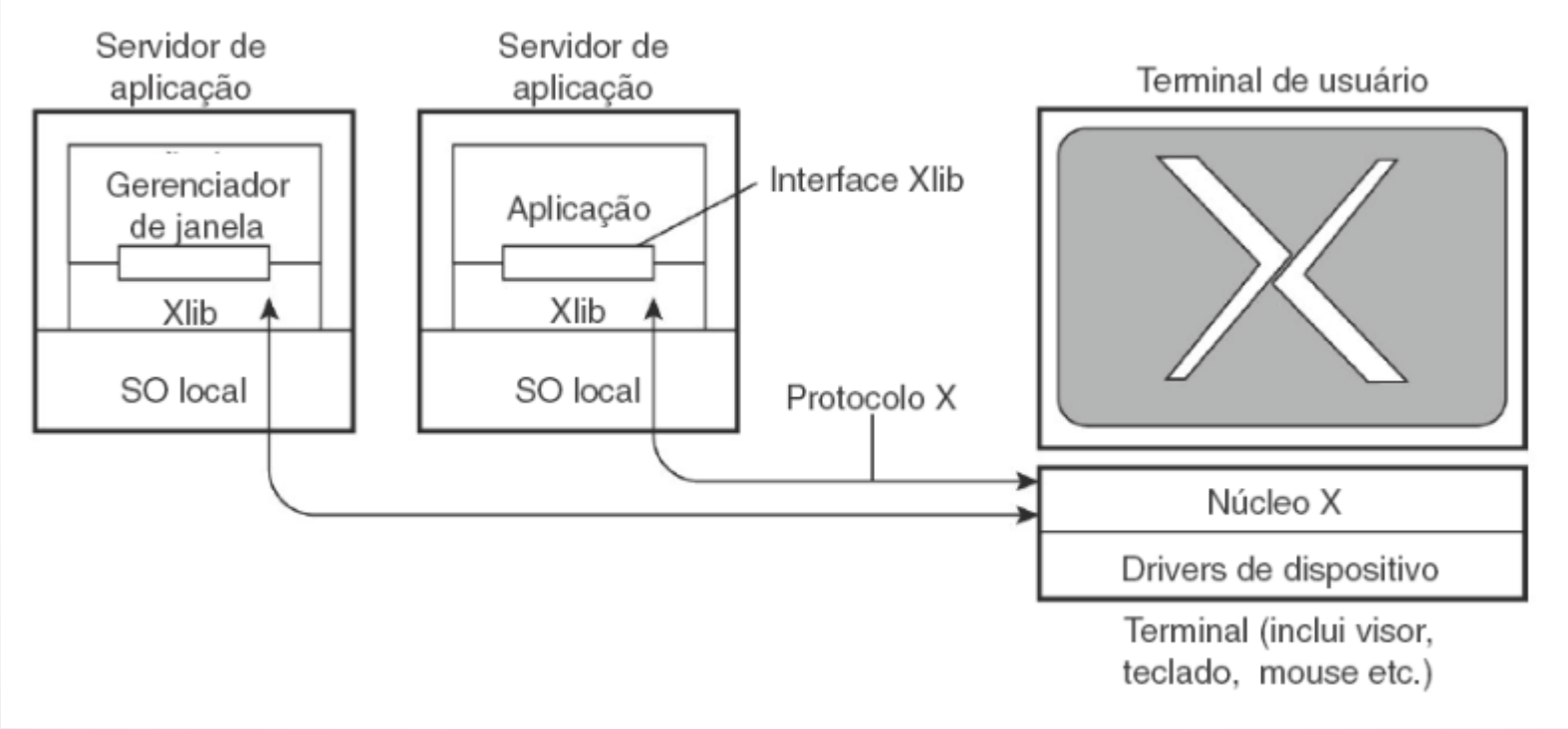


Sistema X Window

- O sistema X é baseado no modelo Cliente-Servidor
  - Servidor X é um programa que executa no sistema e é responsável por todos os acessos ao hardware gráfico.
  - O cliente se comunica com o servidor, enviando requisições como 'desenhe uma linha' ou 'preste atenção na entrada do teclado'.
- Pode ser visto como uma parte do sistema operacional que controla o terminal.
- O cerne do sistema é formado pelo núcleo X
- O núcleo X oferece uma interface de nível relativamente baixo para controlar a tela e também para capturar eventos do teclado e do mouse
- Interface disponibilizada para as aplicações é definida pela biblioteca Xlib

Sistema X Window

- Organização básica do sistema X Window



Sistema X Window

- O núcleo X e as aplicações X precisam necessariamente residir na mesma máquina.
- É fornecido o protocolo X, que é um protocolo de comunicação de chamada de aplicação pelo qual uma instância de Xlib pode trocar dados e eventos com o núcleo X.
  - Exemplo: Xlib pode enviar requisições ao núcleo X para criar ou encerrar uma janela, estabelecer cores e definir o tipo de cursor. O núcleo X reagirá a eventos locais como entrada de teclado e mouse devolvendo pacotes de eventos a Xlib.
- O núcleo X recebe requisições para manipular o visor.
- O núcleo X age como um servidor, enquanto as aplicações desempenham o papel de clintes.

Clientes – Transparência de Distribuição

- O software do lado do cliente pode ser mais do que uma interface.
- Em alguns casos, parte do nível de processamento e dados são executados do lado do cliente.
- Transparência de Migração:
  - Se um cliente já está vinculado a um servidor, ele pode ser informado quando o servidor mudar de localização.
  - Middleware no cliente → oculta do usuário a corrente localização do servidor.
- Transparência a Falha:
  - Middleware cliente pode ser configurado para tentar a conexão com um servidor repetidas vezes, ou tentar um outro servidor após várias tentativas.

Servidores – Questões de Projeto

- Cada servidor é organizado do mesmo modo:
  - Espera por uma requisição do cliente.
  - Assegura o atendimento.
  - Espera pela próxima requisição.

- Questões a serem levantadas:
  - Como tratar as requisições?
  - Como os clientes contatam um servido?
  - Como interromper a comunicação com o servidor?
  - O servidor deve guardar o estado dos clientes?

Servidores – Tratamento de Requisições

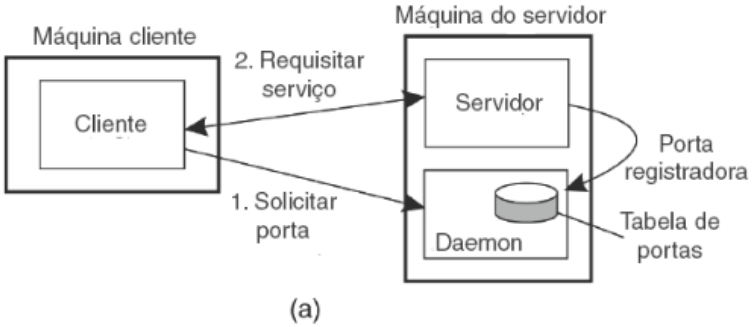
- Servidor Iterativo
  - O próprio servidor manipula a requisição e, se necessário, retoma a resposta ao cliente
- Servidor Concorrente
  - Não manipula por si próprio a requisição
  - Exemplo: Servidor Multithread
  - Processos ou Threads que devolvam a resposta ao cliente

Servidores – Como o cliente contata?

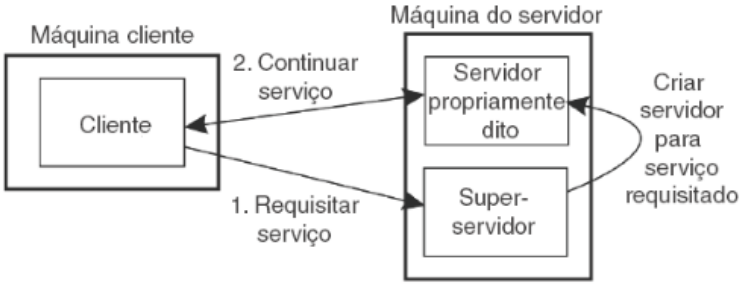
- Os clientes enviam as requisições a uma porta, na máquina em que o servidor está rodando.
- Para alguns serviços conhecidos (HTTP, FTP) existem portas pré-definidas
- Para serviços que não exigem portas pré-definidas, uma opção é ter um *daemon* e o sistema operacional informa dinamicamente uma porta.
- Outra maneira de manipular portas é o uso de superservidores:
  - Para os casos onde diversos servidores que possuem uma porta específica estão rodando em uma mesma máquina.
  - O monitoramento passa a ser realizado por um **superservidor**. Tem-se uma escuta em cada porta associada como um serviço específico.
  - Exemplo: Daemon inetd no Unix.

Servidores – Como o cliente contata?

- A) Vinculação cliente-a-servidor usando um daemon.
- B) Vinculação cliente-a-servidor usando um superservidor



(a)



(b)

Servidores – Interrompendo a comunicação

- Abordagem mais simples:
  - O usuário sai abruptamente da aplicação cliente → o servidor encerrará a conexão antiga.

- Abordagem mais completa:
  - Dados "urgentes" (fora da banda):
    - Pacotes TCP possuem no *header* o compo URG.
    - O servidor ao receber um pacote com o campo URG marcado é interrompido, através de um sinal.

Servidores – Manutenção de Estado

- Há três tipos:
  - Sem estado
  - Estado flexível
  - Com estado

Servidores sem Estado

- Não mantém informações sobre os estados dos clientes
- Pode mudar seu próprio estado sem ter de informar a nenhum cliente
- Servidor Web: após processar uma requisição, esquece o cliente.
- Mesmo sem estado, o servidor Web guarda informações dos clientes.
  - A perda não resulta na interrupção do serviço.

Servidores com Estado Flexível

- Servidor promete manter estado em nome do cliente, por um tempo limitado.
- Após o tempo limitado, o servidor descarta estas informações.
- Exemplo:
  - Servidor promete manter um cliente informado sobre atualizações

Servidores com Estado

- Mantém informações persistentes sobre seus clientes
- Informações precisam ser explicitamente removidas pelo servidor.
- Exemplo:
  - Servidor de arquivos que permite a um cliente manter cópia local de um arquivo.
  - Servidor mantém uma tabela com entradas (cliente, arquivo).
  - Permite monitorar as permissões sobre um arquivo, etc

  - Possível problema: Recuperação de Falhas.

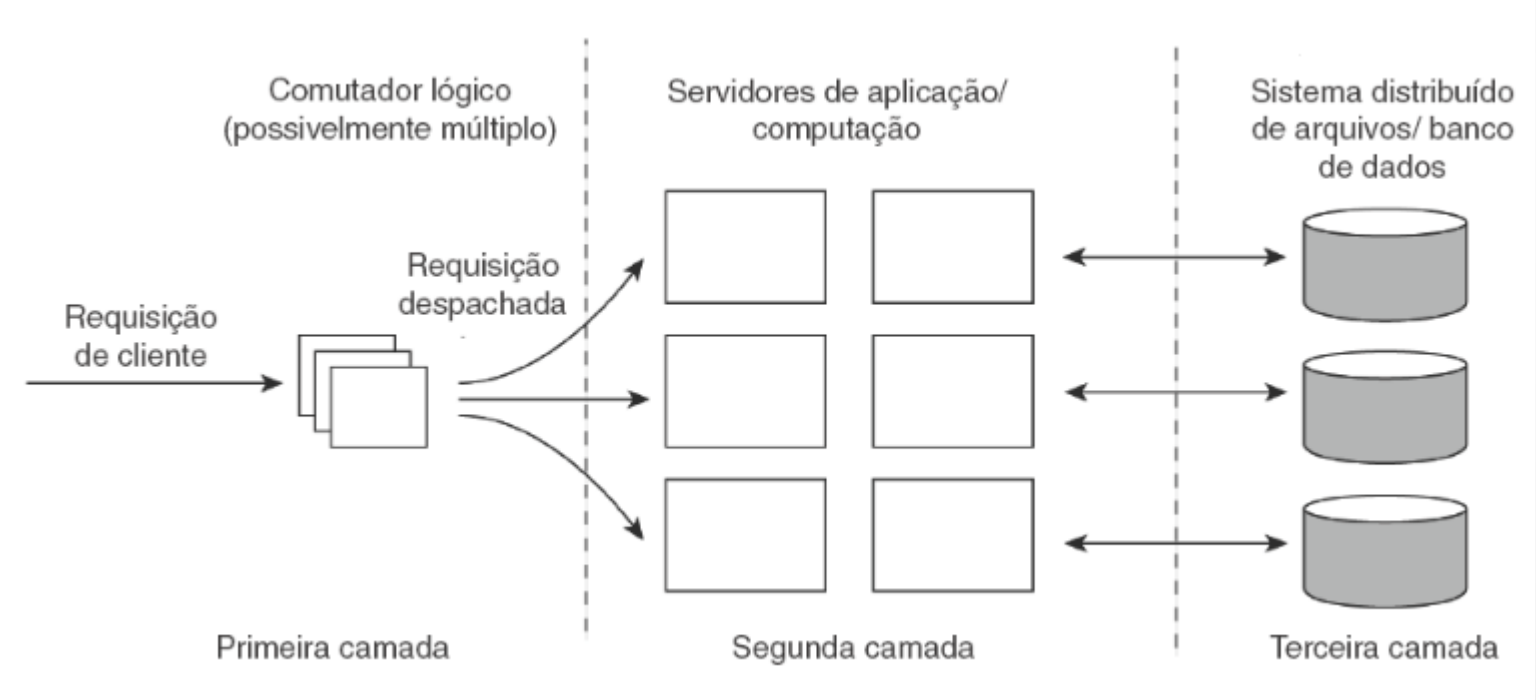
Clusters de Servidores

- Conjunto de máquinas conectadas por uma rede, no qual cada máquina executa um ou mais servidores.

- Estas máquinas estão conectadas por uma rede local, com alta largura de banda e baixa latência.

Clusters de Servidores

- Organização geral de um cluster de servidores de três camadas.

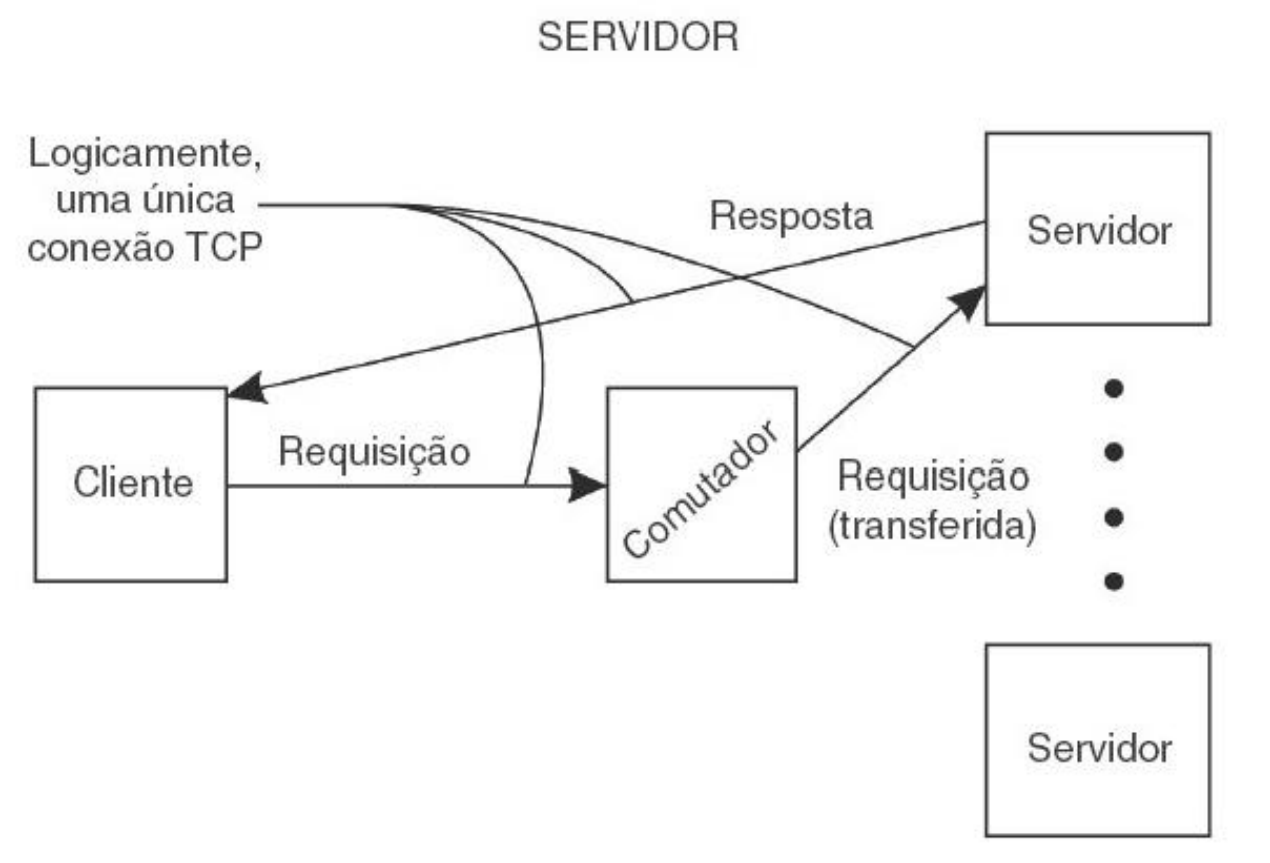


Clusters de Servidores Comutador

- Forma o ponto de entrada para o Cluster de Servidores, oferecendo um único endereço de rede.
- Destacando o TCP, o comutador passa a aceitar as requisições e a transferi-las a um dos servidores.
- Identifica o melhor servidor a tratar a requisição.

Clusters de Servidores Comutador

- Princípio da transferência do protocolo TCP



Clusters de Servidores Distribuídos

- Conjunto de máquinas que possivelmente muda dinamicamente, com vários pontos de acesso variáveis.
- Essas máquinas se apresentam ao "mundo" externo como uma única e poderosa máquina.
- Pontos de destaque:
  - Vários pontos de acesso;
  - Estabilidade no acesso;
  - Alto desempenho;
  - Flexibilidade.

Migração de Código

- Somente se aborda a passagem de dados entre diferentes máquinas.
- Em alguns casos é importante migrar o código de uma máquina para outra.
- Dúvida: Como fazer essa tarefa em sistemas heterogêneos?

Por que Migrar Código?

- A principal razão é o aumento de desempenho!!

- O envio de processos de máquinas sobrecarregadas para máquinas com cargas mais leves.
- Evitar grande quantidade de mensagens trocadas entre aplicações cliente-servidor.
  - Exemplo: operações de banco de dados que envolvem uma grande quantidade de dados (aplicação cliente → servidor).
  - Exemplo: formulários enviados do servidor para o cliente.

Modelos para Migração de Código

- A migração de código em si é algo muito mais amplo:
  - Podemos também migrar o *status* de um programa, sinais pendentes e outras partes do ambiente.
- Modelos para Migração de Código:
  - O processo consiste em três segmentos:

    1. Segmento de código: Contém o conjunto de instruções que compõem o programa que está em execução.
    2. Segmento de recursos: Contém referências a recursos externos (arquivos, impressoras).
    3. Segmento de execução: Armazenar o estado em execução de um processo no momento da migração (dado privados, pilha, contador de programa).

Modelos para a Migração de Código

- A mobilidade pode ser de dois tipos:

1. Mobilidade Fraca: Possível transferir somente o segmento de código, junto com alguns dados de inicialização. Requer somente que a máquina alvo possa executar o código (portabilidade).

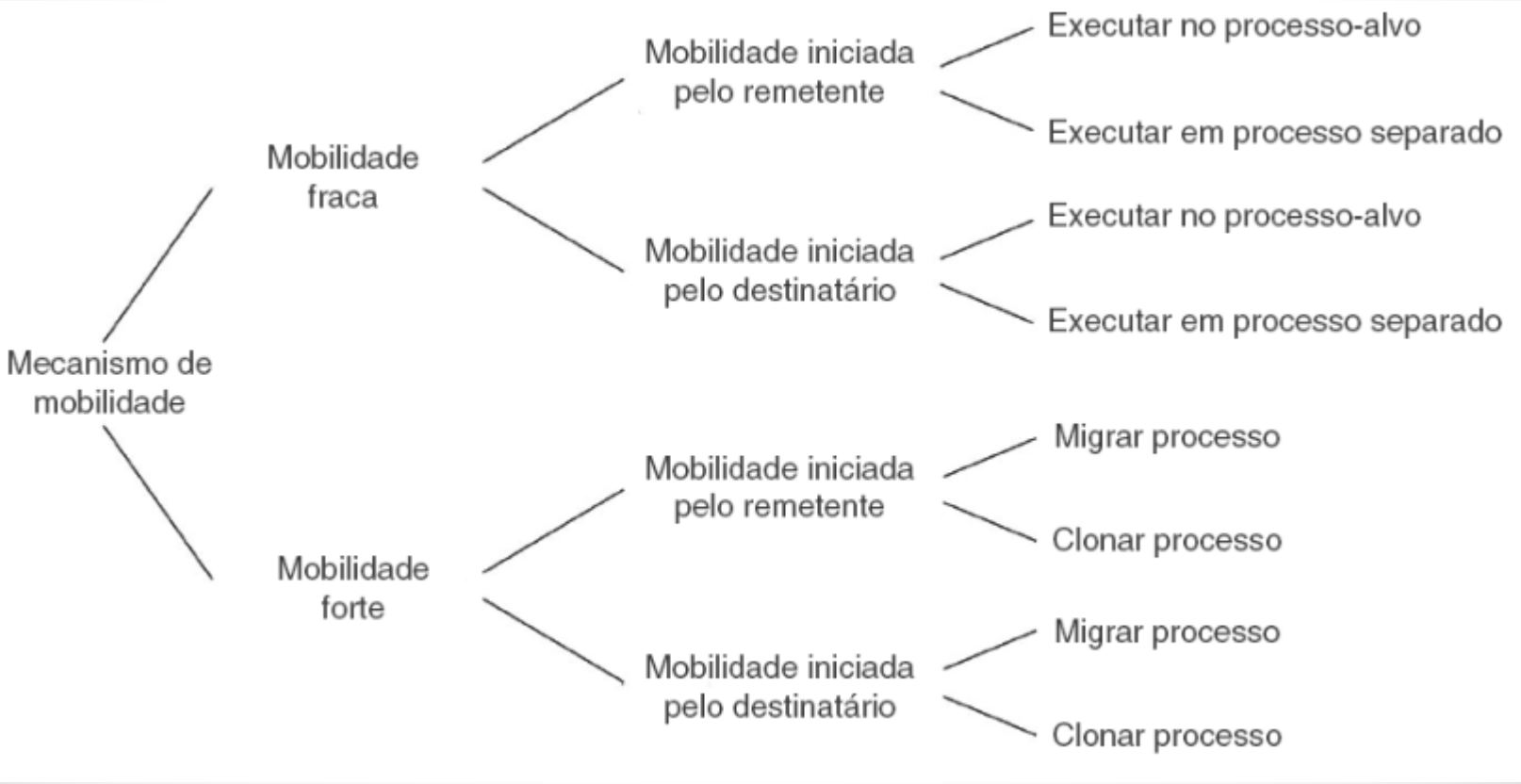
2. Mobilidade Forte: O segmento de execução também pode ser transferido. O processo em execução pode ser parado e movido para uma outra máquina e retomar a execução no ponto original.

Modelos para Migração de Código

- Em relação ao ponto de início da migração, podemos ter:
  - Iniciada pelo remetente: a migração é iniciada na máquina em que o código está em execução no momento em questão.
    - Exemplo: Enviar programa de busca pela internet a um servidor de banco de dados para a realização de pesquisas.
  - Iniciada pelo destinatário: A iniciativa da migração de código é tomada pela máquina alvo.

Modelo para a Migração de Código

- Esquema de alternativas para a migração de código.



Migração e Recursos Locais

- Segmentos de recursos requerem uma atenção especial.
  - Exemplo: Comunicação de um processo sendo feita através de uma porta TCP. Ao mudar de localização, este processo deverá devolver a porta e requisitar uma nova no destinatário.

- Há três tipos de vinculações processo – recurso:
  1. Vinculação por identificador: o processo requer exatamente o recurso referenciado (Exemplo: a URL de um site);
  2. Vinculação por valor: se outro recurso fornece o mesmo valor, a execução não é afetada (bibliotecas padronizadas);
  3. Vinculação por tipo: O processo requer um recurso de um tipo específico (monitores, impressoras).

Migração e Recursos Locais

- Há três tipos de vinculações recurso – máquina:

1. Recursos não ligados: recursos podem ser movidos com facilidade entre máquinas diferentes (arquivos de dados).

2. Recursos amarrados: recursos podem ser movidos ou copiados, mas só a custos relativamente altos (banco de dados locais).

3. Recursos fixos: os recursos estão intimamente vinculados a uma máquina ou ambiente específico e não podem ser movidos (porta).

Migração de Recursos Locais

- Referência Global
  - Caso vários processos referenciem o mesmo recurso, uma alternativa é estabelecer uma referência global, que atravesse as fronteiras das máquinas.
    - URL (baixa complexidade)
    - Memória compartilhada entre processos (referência global significa memória compartilhada).
- Há várias combinações entre vinculação processo-recurso e recurso-máquina.
  - Exemplo: O processo admite que a memória pode ser compartilhada entre processos (recursos fixo, vinculação de valor).
  - Exemplo: Bibliotecas de tempo de execução (recurso amarrado, vinculação de valor).

Migração e Recursos Locais

- A complexidade está ligada aos tipos de vínculos processo-recurso e recurso-máquina.
  - Recursos não ligados: Melhor solução é copiar ou mover o recurso para o novo destino.
  - Vinculação por tipo: A solução é vincular novamente o processo a um recurso do mesmo tipo disponível no local.

Migração em Sistemas Heterogêneos

- Uma migração em sistemas heterogêneos requer:
  - Que o segmento de código possa ser executado em cada plataforma;
  - Que o segmento de execução possa ser adequadamente representado em cada plataforma;
  - Efeito Global: Migrar sistema operacional inteiro, em vez de migrar processos.

Leitura, Estudo e Exercícios

- Este material tem sua fonte no livro:
  - Sistemas Distribuídos: Princípios e Paradigmas.
  - Andrew Tanenbaum e Marteen Steen
  - Capítulo 3

  - Distributed System – Concepts and Design
  - George Coulouris, Jean Dollimore, Tim Kindberg, Gordon Blair.
  - Capítulo 4