

Tipos de Servidores

- **Com Estado X Sem Estado**
- **Sequenciais X Concorrentes**
- **Com Conexão X Sem Conexão**

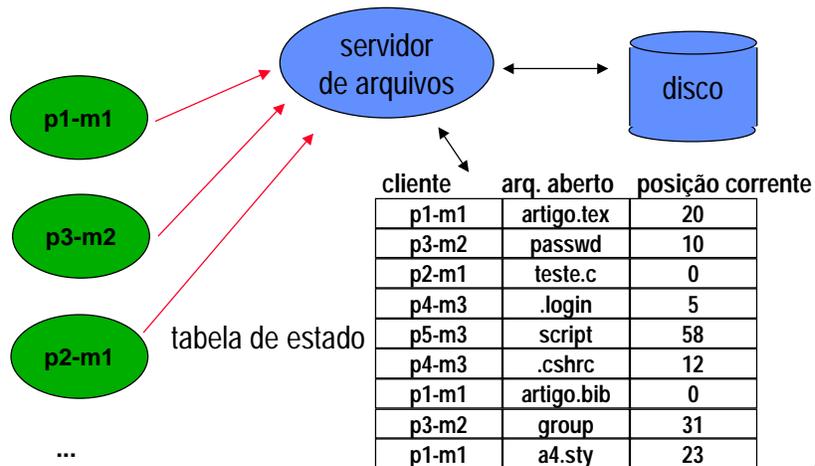
Thais Batista - UFRN - DIMAp

Servidores com estado

- **Servidores *com estado* armazenam informação sobre cada cliente.**
- **Operações podem ser implementadas de forma mais eficiente.**
- **Mensagens com pedidos podem ser menores.**

Thais Batista - UFRN - DIMAp

Servidores com estado



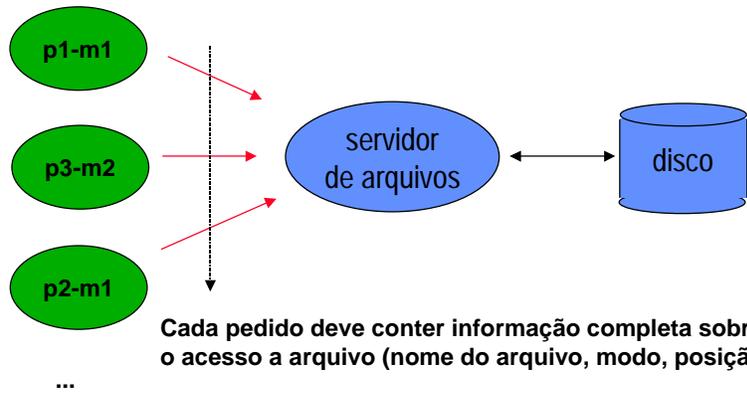
Thais Batista - UFRN - DIMAp

Servidores sem estado

- Em um servidor *sem estado* cada pedido deve conter toda a informação necessária para seu processamento.
- Servidores sem estado apresentam maior escalabilidade.
 - escalabilidade: comportamento quando o número de clientes cresce.
- A implementação de um servidor sem estado é muito mais simples que a de um servidor com estado.

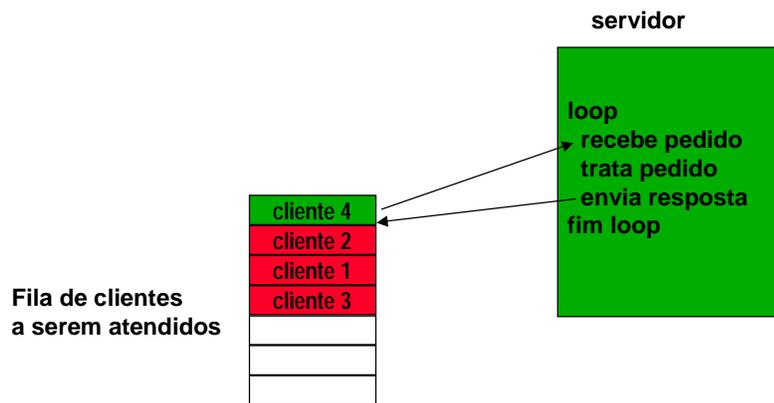
Thais Batista - UFRN - DIMAp

Servidores sem estado



Thais Batista - UFRN - DIMAp

Servidores sequenciais



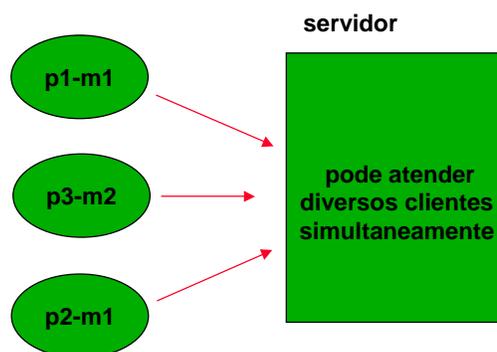
Thais Batista - UFRN - DIMAp

Servidores Sequenciais

- Fáceis de programar e modificar
- Adequados para situações onde a execução do serviço é muito rápida portanto o tempo de resposta não é alto
- Trabalham melhor com protocolos sem conexão mas podem ser usados também com protocolos com conexão (o custo de estabelecer a conexão pode não compensar)
- Em geral não guardam estado das solicitações

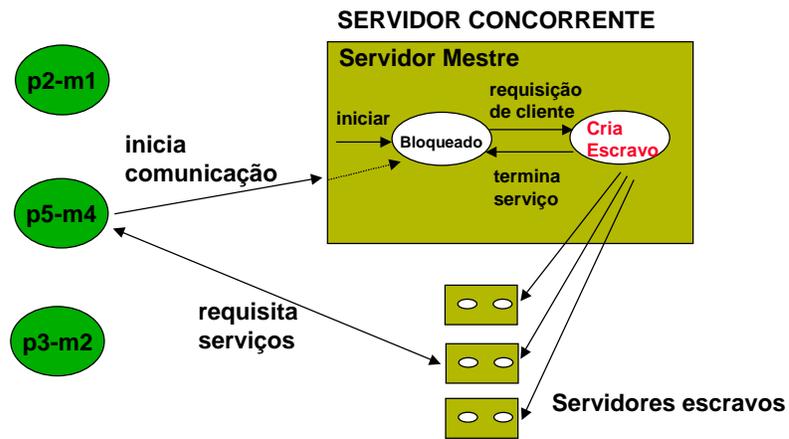
Thais Batista - UFRN - DIMAp

Servidores Concorrentes



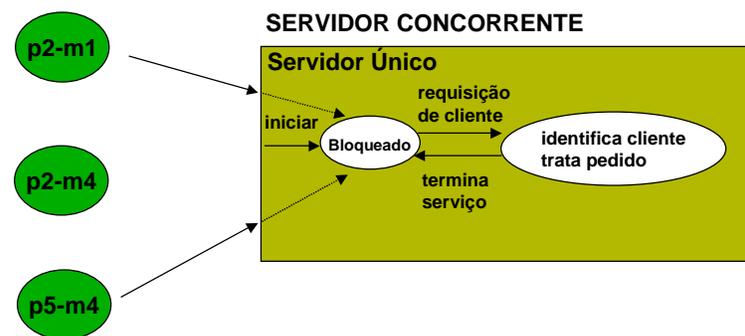
Thais Batista - UFRN - DIMAp

Servidores Concorrentes Multiprocesso



Thais Batista - UFRN - DIMAp

Servidores Concorrentes Monoprocesso



Thais Batista - UFRN - DIMAp

Servidores Concorrentes Monoprocesso

- O servidor usa um único processo para oferecer concorrência aparente para os clientes
- Adequado para aplicações onde I/O domina
 - criação de processos e troca de contextos entre processos implica em alto custo
- O servidor usa I/O assíncrono para oferecer concorrência aparente para os clientes
 - a chegada de dados dispara a execução
 - suporte para I/O assíncrono através da chamada *Select*
 - o servidor passa o conjunto de descritores de socket como argumento da chamada *Select*
- O processo servidor gerencia múltiplos sockets e faz o papel de mestre e de escravo

Thais Batista - UFRN - DIMAp

Algoritmo do Servidor Concorrente, Orientado a Conexão e Monoprocesso

1. Cria um *Socket*
 2. Associa a uma porta
 3. Adiciona o descritor do socket a uma lista
 4. Chama *Select* para observar I/O na lista de sockets
 5. Se o socket original é selecionado, chama *Accept* para obter a próxima conexão e adiciona o novo socket a lista
- Se algum outro socket é selecionado, chama *read* para obter a próxima solicitação. Faz o processamento e chama *write* para enviar a resposta
6. Continua o processamento do passo 4

Thais Batista - UFRN - DIMAp

Servidores Concorrentes multiprocesso

- **O processo mestre nunca comunica-se diretamente com o cliente**
- **Quanto aos processos Escravos:**
 - Pré-alocação de Escravos
 - Alocação de Escravos sob Demanda

Thais Batista - UFRN - DIMAp

Alocação sob Demanda

- **Escravos são criados à medida que o servidor recebe novas solicitações**
- **Os escravos são finalizados quando terminam de atender uma solicitação**
- **Vantagem: não aloca recursos do sistema sem necessidade**
- **Desvantagem: a criação do processo pode ser uma tarefa demorada (principalmente em sistemas com único processador)**

Thais Batista - UFRN - DIMAp

Pré-alocação

- **Consiste em criar N processos escravos quando o mestre começa a executar**
- **Quando uma solicitação chega, um dos processos escravos é escolhido para receber a solicitação e tratá-la**
- **Os escravos não são finalizados quando terminam de atender alguma solicitação**
- **Evita o custo de criar processos quando a solicitação chega ao servidor**
- **Pode atender os clientes de forma mais rápida que na alocação sob demanda**

Thais Batista - UFRN - DIMAp

Servidores Concorrentes

- **A maior parte usa vários processos**
- **Trabalham melhor com protocolo com conexão pois o custo de criação de processos não é desprezível. Em protocolos sem conexão o processo escravo termina após atender 1 única solicitação).**

Thais Batista - UFRN - DIMAp

Usando programas separados como Escravos

Thais Batista - UFRN - DIMAp

Tipos de Servidores

	sem estado	com estado
sequencial		
concorrente		

Thais Batista - UFRN - DIMAp

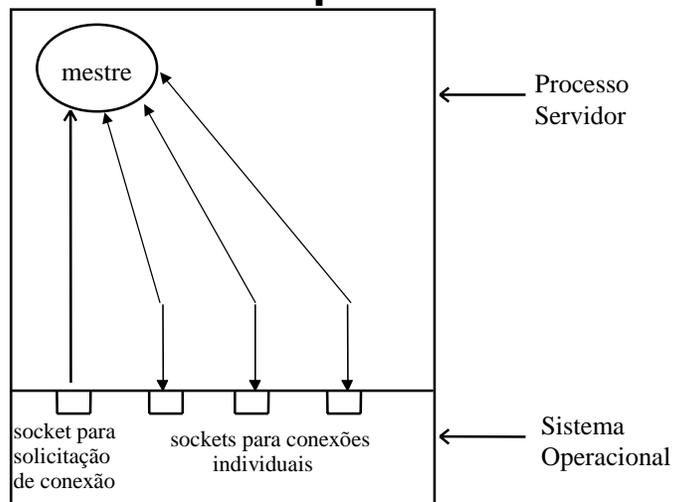
Servidores TCP/IP

- Servidores com estado em geral são implementados por comunicação com conexão.
- Interações com servidores com estado em geral são longas... necessidade de servidor concorrente!
 - modelo mais comum neste caso é o servidor concorrente multiprocesso!

	sem	com	(conexão)
sequencial	✓		
concorrente		✓	

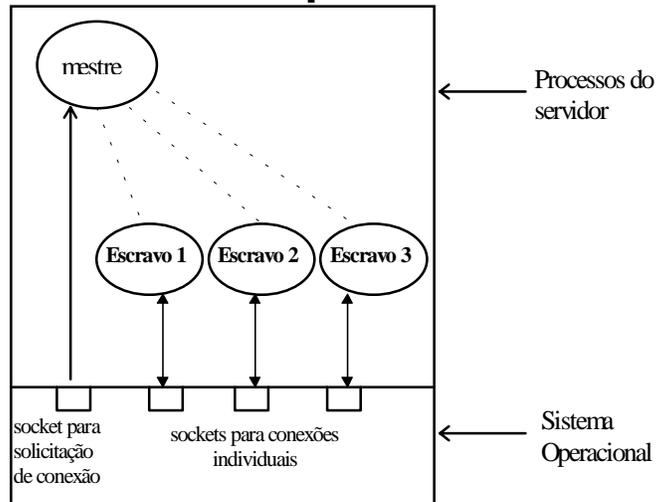
Thais Batista - UFRN - DIMAp

Servidores Concorrentes Monoprocesso



Thais Batista - UFRN - DIMAp

Servidores Concorrentes Multiprocesso



Thais Batista - UFRN - DIMAp