

Arquitetura de Servidores WEB

- Introdução
- Formas de Organização
- Questões de Desempenho
- Implementação de Servidores WEB
 - Servidores WEB Multiprocessos
 - Servidores WEB Multithreads
 - Servidores WEB SPED

Introdução

- Servidores Web tradicionais são implementados no espaço do usuário e são atualmente os mais conhecidos e utilizados.
- Recentemente, tem surgido uma nova tendência de implementação de servidores Web objetivando melhor desempenho: a implementação direta.
- O servidor, portanto, deve ser projetado de modo a atender o maior número de requisições que lhe for possível. As diversas arquiteturas apresentadas a seguir procuram justamente melhorar o nível de concorrência que é possível alcançar no servidor.

Formas de Organização

- **Servidor Interativo**

- Este seria o tipo de servidor mais simples, que apenas fica aguardando as requisições dos clientes e as trata uma por vez, na ordem de sua chegada, sem nenhum tipo de concorrência. Sua inclusão aqui se dá apenas por finalidade didática, pois, na prática, seria um servidor muito ineficiente.

Formas de Organização

- **Processo por Requisição**
 - Existe um processo principal (o “pai”) que fica aguardando requisições em uma porta pré-definida. Ao chegar uma nova requisição, ele cria uma cópia e passa a esse “filho” a responsabilidade de tratar a solicitação, retornando em seguida ao seu laço de espera.
 - Nessa arquitetura, em um determinado momento, existirão inúmeras cópias executando concorrentemente, limitadas apenas pela capacidade da máquina que as abriga. Evidentemente, existe um limite para o número de processos concorrentes no servidor, sob pena de bloquear sua execução, pois cada processo filho ocupa memória da máquina e ciclos de CPU, além de possuir uma conexão de rede aberta com um cliente.

Formas de Organização

- **Pool de Processos**

- Esta solução tenta aproveitar a simplicidade da abordagem anterior, ao mesmo tempo em que procura eliminar suas desvantagens. Neste caso, o servidor, ao iniciar sua execução, cria um número mínimo de processos, que constituem o *process pool*.
- Os processos do *pool*, após atenderem uma requisição, não encerram sua execução; eles ficam de prontidão, esperando até que sejam convocados para atuar novamente. Esta solução é usada nas arquiteturas de servidores web como o Apache.

Formas de Organização

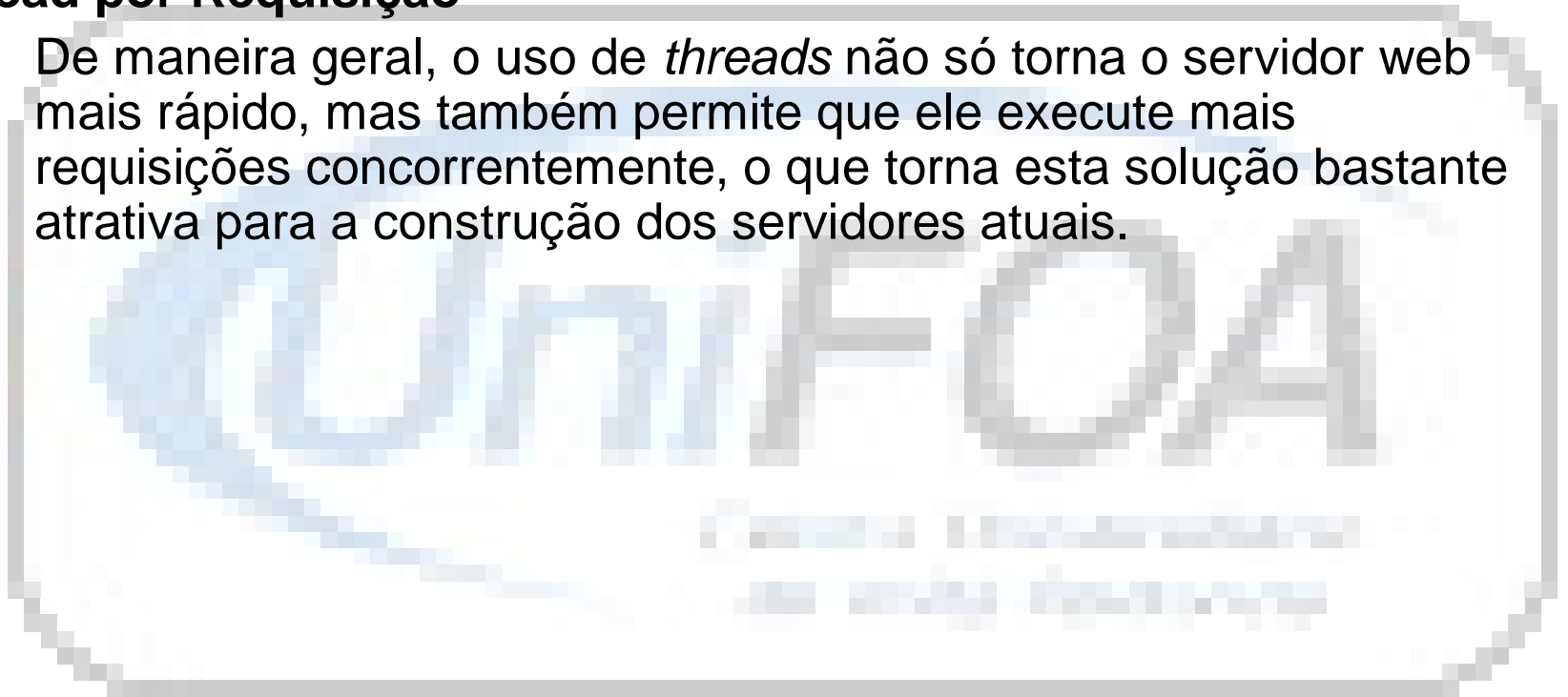
- **Thread por Requisição**

- Para cada nova requisição que chega é criada uma nova *thread* (linha de execução) para tratá-la. A vantagem é que uma *thread* consome muito menos recursos da máquina que um processo, pois diferentes *threads* podem compartilhar o mesmo espaço de endereçamento, incluindo código e dados globais. E, como as *threads* se encontram dentro de um mesmo processo, a mudança de contexto entre elas também se dá de forma mais rápida.
- A utilização de *threads* é bastante indicada para o caso de tarefas que frequentemente precisam esperar por I/O, pois é preciso aguardar pela leitura dos dados no disco e também pela recepção e envio dos pacotes pela rede.

Formas de Organização

- **Thread por Requisição**

- De maneira geral, o uso de *threads* não só torna o servidor web mais rápido, mas também permite que ele execute mais requisições concorrentemente, o que torna esta solução bastante atrativa para a construção dos servidores atuais.



Formas de Organização

- **Pool de Threads**

- Finalmente, outra alternativa possível é a utilização de um *pool* de *threads*, que são criadas quando o servidor web é iniciado. Dessa forma, elimina-se a sobrecarga de criação de uma nova *thread* para cada requisição que chega ao servidor, a exemplo do caso da utilização do *pool* de processos, além de permitir um controle refinado do nível de multiprogramação utilizado.

Questões de desempenho

- Técnicas para melhoria de desempenho na implementação da arquitetura de servidores Web se focam nos seguintes objetivos:
 - eliminação de cópia e leitura de dados;
 - redução de escalonamento e *overhead* causado pela notificação de eventos;
 - redução da comunicação entre as várias camadas da pilha de protocolos de rede.

Implementação de Servidor WEB

- Esta arquitetura é a tradicional para a implementação de aplicações em geral.
- A arquitetura de servidores Web é somente mais uma aplicação que se utiliza de recursos do sistema, com privilégios limitados, através da interface de *system calls*.
- Desta forma, o servidor Web é completamente dependente da interface fornecida pelo Sistema Operacional, e quão otimizada é implementada essa interface, baseando-se nas questões de desempenho.

Implementação de Servidor WEB

- Devido à demanda por serviços mais rápidos, tem-se notado a descoberta de formas de melhoria do desempenho e escalabilidade do servidor Web através da maior integração entre servidores Web e Sistemas Operacionais.
- Esta tendência tem provocado uma busca pela otimização das interfaces e implementações existentes nos Sistemas Operacionais especialmente para atender o serviço Web.
- Por exemplo, implementações de primitivas como `select()` e `poll()` tem sido otimizadas em Sistemas Unix para se reduzir o *overhead* de notificação de eventos em arquiteturas que se utilizam de chamadas não bloqueantes.

Implementação de Servidor WEB

- Os servidores implementados no espaço do usuário dividem-se basicamente em três grupos, por evolução na implementação:

- servidores Web multiprocessos (MP);
- servidores Web *multithreads* (MT);
- servidores Web SPED.(Single Process Event Driven)

servidores Web multiprocessos (MP)

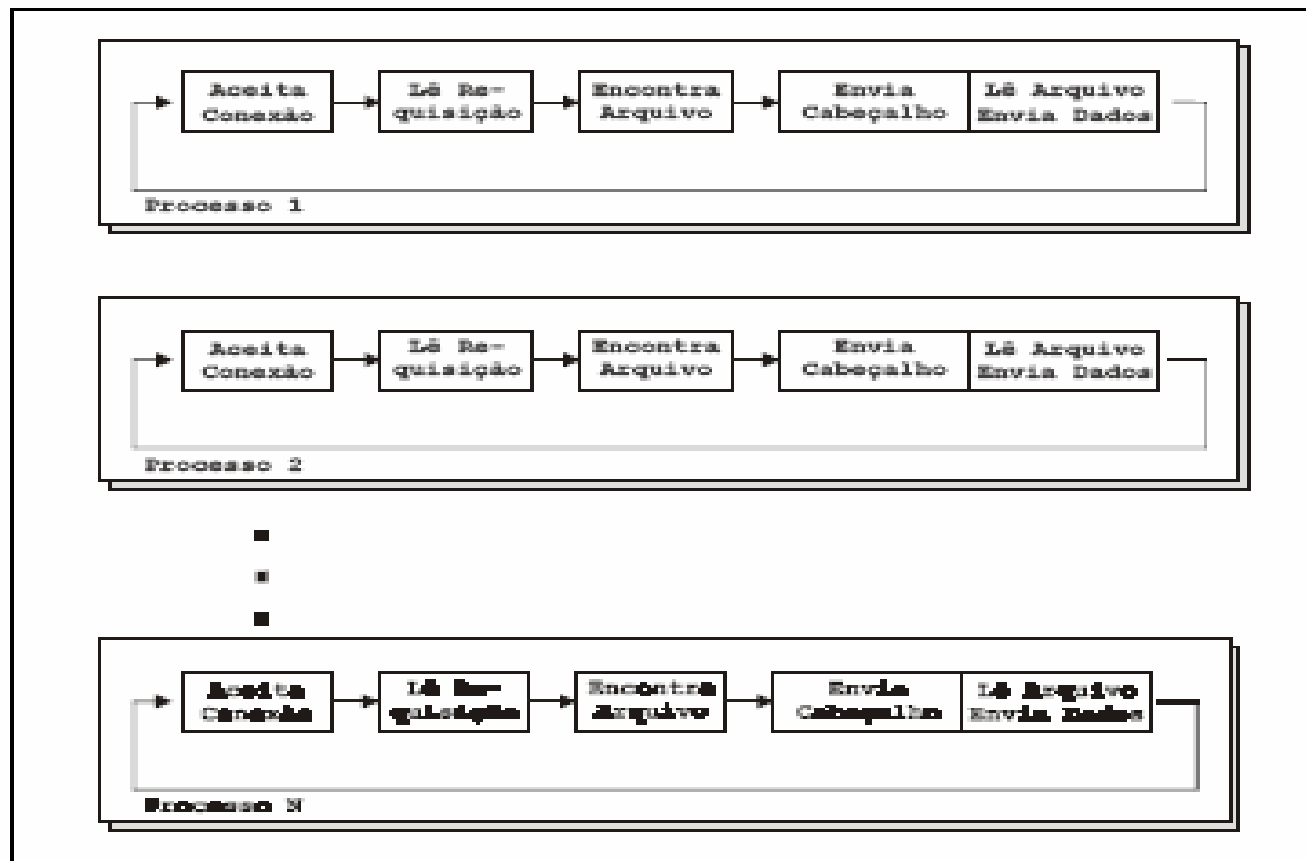


Figura 5.1: No modelo MP cada servidor manipula uma requisição por vez. Processos executam estágios sequencialmente.

servidores Web multithreads (MT)

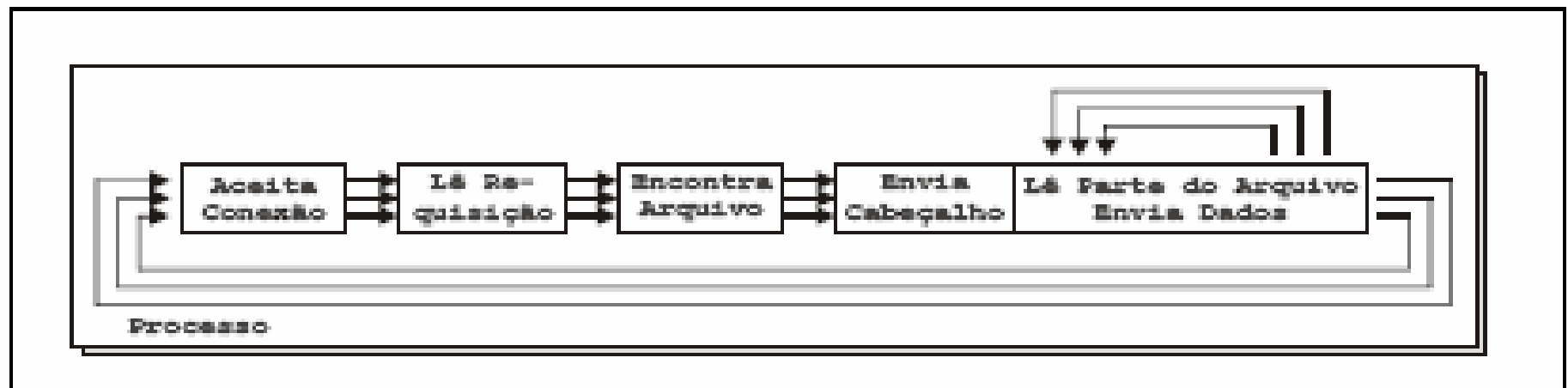


Figura 5.2: O modelo MT utiliza-se de um único espaço de endereçamento e múltiplas *threads* concorrentes de execução, cada uma manipulando uma requisição distinta.

servidores Web SPED

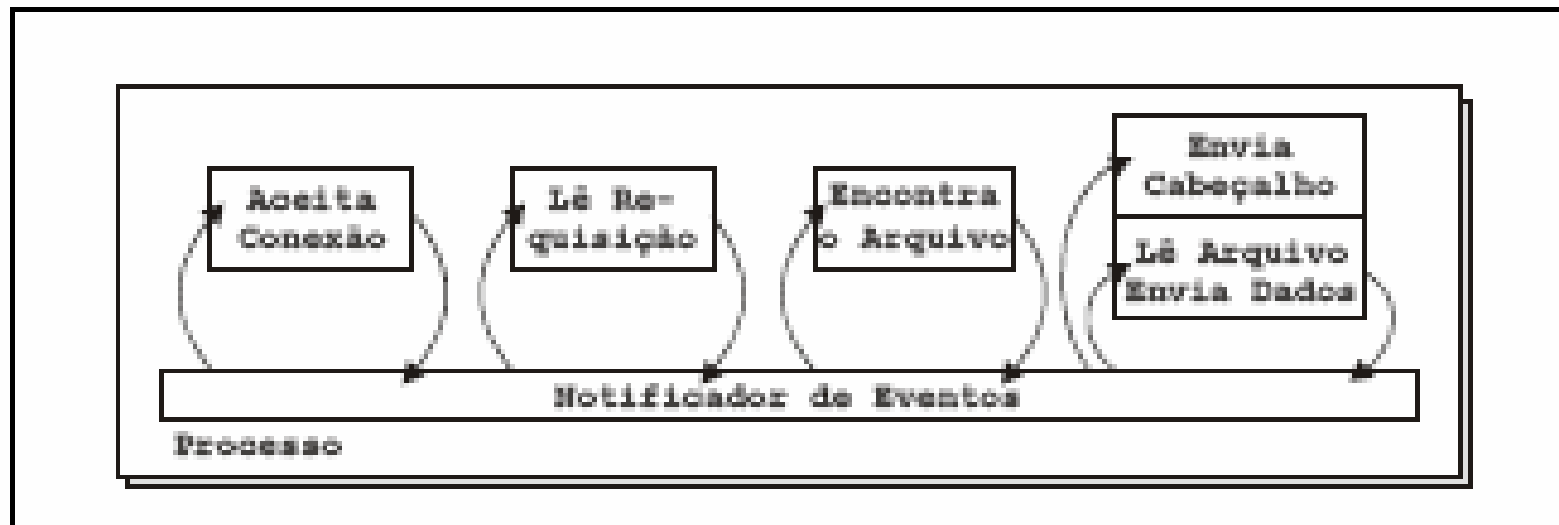


Figura 5.3: O modelo SPED se utiliza de um único processo para executar todo o processamento e atividade de disco baseado em eventos. Múltiplas requisições são manipuladas por um único processo.

Exercícios

1. Como são implementados servidores Web tradicional?
2. Como deve ser projeto um servidor Web tradicional?
3. Os servidores implementados no espaço do usuário dividem-se basicamente em três grupos, por evolução na implementação. Cite esses grupos e explique seu funcionamento.
4. Cite e explique, com suas palavras, as formas de organização em servidores Web.
5. Tomando como foco os objetivos, quais as técnicas implementadas para melhoria de desempenho na arquitetura de servidores Web?