Alunos: Caio Domingues, Vinicius Coelho.

Matriculas: 201700550, 201700121

**1° Questão**

Os servidores web são implantados no espaço do usuário, mas com o surgimento de uma nova tendência começou visar-se um melhor desempenho, então surgiu o interesse de buscar novas formas de se organizar um servidor web.

**2° Questão**

O servidor deve ser implantado de forma que atenda o maior número de requisições possível, tendo várias formas de ser implantada, com isso cabe-se escolher a arquitetura que mais se adéqua.

**3° Questão**

**Servidores Web multi-processos (MP):** Um único processo consegue executar todos os passos sequencialmente, sendo uma requisição por vez. Quando se utilizar múltiplos processos consequentemente múltiplas requisições podem ser atendidas.

**Servidores Web Multi-Threads (MT):** Esse servidor emprega múltiplas threads independentes dentro de um único processo, num único espaço de memória, cada threads manipula uma só requisição e são executadas sequencialmente, bastante semelhante ao MP, tendo a única diferença o espaço de memória.

**Single Process Event Driven (SPED):** O servidor SPED utiliza apenas um processo para executar todo o processamento, ou seja, um único processo é capaz de executar múltiplas requisições.

**4ª Questão**

• **Servidor Interativo:** O servidor mais simples, trata as requisições uma a uma e ordem de chegada, sem que haja concorrência. Na prática, é um servidor ineficiente, sendo utilizado apenas para fins didáticos.

• **Processo por Requisição:** Um processo principal fica esperando requisições em uma porta pré-definida, ao ter uma requisição, ele cria uma cópia e transfere a outro processo -“secundário” - a responsabilidade de tratar este. A arquitetura é limitada pela capacidade da máquina no qual o servidor se encontrar, visto que existirão inúmeras cópias executando de forma concorrente. Tendo como limite o número de processos concorrentes, com a possibilidade de bloqueio de uma execução, já que cada processo “secundário” ocupa memória e CPU, além da conexão em rede com um cliente.

**• Pool de Processos:** Baseada nas abordagens simplistas do anterior, procurando eliminar suas desvantagens. O diferencial é que ao seu início, o servidor cria um número mínimo de processos, a chamada **Process Pool**. Estes que após atenderem uma requisição, não encerram sua execução, eles ficam ativos esperando outra requisição. Utilizada em servidores web como o Apache.

• **Thread por Requisição:** Para cada requisição, se cria uma nova linha de execução (thread). A vantagem é a redução do custo de recursos de uma máquina em relação a de um processo, já que varias threads podem utilizar o mesmo endereço, códigos e dados; como estes estão agrupados num mesmo processo, a mudança de contexto é mais ágil. Recomendado para tarefas com alto uso de dispositivos de E/S, já que é necessário aguardar leitura de dados no disco e pela recepção/envio de pacotes por rede. 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.

• **Pool de Threads:** Funcionamento similar ao Pool de Processos, difere somente a troca de Processos por Threads, eliminando a sobrecarga da criação de uma nova thread p/ cada requisição nova no servidor. Seu principal diferencial é o controle aprimorado no nível de multiprogramação utilizado.

**5ª Questão**

Técnicas para melhoria de desempenho de servidores Web:

* 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.