

Análise de Desempenho entre Threads Tradicionais e Threads Virtuais em Java

Stephanye Cristine Antunes De Cunto

2025

Proposta do TCC

O Java possui diversas formas de criar uma thread, como herdando a classe `Thread`, implementando a interface funcional `Runnable` ou utilizando as interfaces `Callable` e `Future`. Com o lançamento do Java 19, foi introduzida uma nova abordagem: as threads virtuais. Diferentemente das threads tradicionais, que são gerenciadas pelo sistema operacional (SO), as threads virtuais são gerenciadas pela JVM (Java Virtual Machine).

O objetivo deste trabalho é analisar se existem diferenças de desempenho entre as técnicas de paralelização quando há uma alta carga. Para isso, serão comparadas threads tradicionais, gerenciadas pelo SO, e threads virtuais, gerenciadas pela JVM, com o intuito de avaliar o desempenho e o comportamento de cada abordagem.

Serão realizados dois benchmarks em ambiente controlado. O primeiro será CPU-bound, utilizando algoritmos como cálculo de números primos ou multiplicação de matrizes, onde se espera que a implementação com threads virtuais não apresente ganhos significativos em relação às threads tradicionais. O segundo será I/O-bound, com a implementação de um servidor HTTP que atenderá requisições simulando espera por meio de `Thread.sleep()`. Nesse caso, espera-se que as threads virtuais apresentem melhor desempenho, devido à sua menor sobrecarga e maior escalabilidade em operações de I/O.

As métricas a serem utilizadas incluem: tempo total de execução, throughput (requisições ou tarefas por segundo), uso médio de CPU e memória, e número de threads criadas. Para o benchmark CPU-bound será analisado o speedup, enquanto para o benchmark I/O-bound será avaliada a latência média.

Para a realização deste trabalho, serão utilizadas as seguintes ferramentas:

- **JDK (Java Development Kit)**: fornecerá o ambiente necessário para compilação e execução dos programas em Java.
- **VisualVM**: será utilizado para monitoramento e análise de desempenho das aplicações.
- **Spring Boot**: será empregado na implementação do servidor HTTP para o benchmark I/O-bound.

Com isso, espera-se obter um relatório comparativo entre as técnicas, identificar os cenários em que cada modelo apresenta melhor desempenho e realizar uma discussão sobre as vantagens e desvantagens de cada abordagem.