

Pergunta Central

Como as threads virtuais funcionam em comparação as threads tradicionais em termos de escalabilidade, utilização de recursos e latência em cargas de trabalho de alto rendimento?

Objetivo

Identificar os cenários cada modelo se destaca e fornecer dados concretos para decisões sobre adoção de Virtual Threads em aplicações Java.

Implementação

Foram desenvolvidas quatro aplicações Java independentes:

- Threads Tradicionais – CPU-bound
- Threads Tradicionais – I/O-bound
- Threads Virtuais – CPU-bound
- Threads Virtuais – I/O-bound

Benchmark 1 – CPU-bound: Cálculo de números primos do limite inferior ao limite superior.

Benchmark 2 – I/O-bound: Simulação de operação de bloqueio criando duas aplicações Java. Um aplicativo faz um GET de requisição e recebe os resultados de outra aplicação Java, que leva um tempo específico de bloqueio e produz a resposta.

Experimentos

- Ambiente controlado no AWS EC2, configurações: t3.xlarge, 4 vCPUs, 16 GB RAM, Ubuntu 22.04
- Repetições: cada teste realizado três vezes
- Warm-up:
 - Benchmark 1 – carga de 10 threads por 5 minutos
 - Benchmark 2 – fase de aquecimento de 10 usuários simultâneos por 60 segundos
- Threads Virtuais: `Executors.newVirtualThreadPerTaskExecutor()`
- Threads tradicionais: `Executors.newFixedThreadPool()`

Métricas de Desempenho

- Taxa de transferência
- Latência
- Uso de memória
- Utilização da CPU

Ferramentas Utilizadas

- JDK 21
- Spring Boot 3.2
- IntelliJ IDEA
- Maven
- Apache JMeter: geração de carga e coleta de métricas
- VisualVM: monitoramento de CPU, heap e threads em tempo real

Resultados

Métrica	Tradicional - CPU	Virtual CPU	-	Tradicional - I/O	Virtual I/O	-
Taxa de transferência	8204,1 req/seg	8153,2 req/seg		5410,1 req/seg	8898,3 req/seg	
Latência	121 ms	118 ms		448 ms	319 ms	
Uso de memória	1,5 GB	1,3 GB		2,2 GB	1,4 GB	
Utilização da CPU	94%	93%		72%	84%	

Conclusão

Para cargas de trabalho com uso limitado de CPU: as threads virtuais e as threads de plataforma apresentam desempenho semelhantes. Tarefas com uso intensivo de CPU sofrem gargalos devido aos recursos computacionais, e o aumento da simultaneidade não terá muita importância.

Para cargas de trabalho vinculadas a I/O: As threads virtuais superaram as threads de plataforma em termos de taxa de transferência e latência. As Threads Virtuais permitiram maior simultaneidade, redução do consumo de memória e uma melhor utilização da CPU.

Trabalhos Futuros

- Cenários de Carga de Trabalho Mista
- Integração com aplicativos nativos da nuvem
- Modelos reativos e assíncronos