Arquitetura Distribuída para Cálculo de Primos

Resumo

Este artigo apresenta uma aplicação cliente-servidor escrita em Python que explora conceitos de sistemas distribuídos, concorrência e paralelismo. A solução responde a requisições de múltiplos clientes via TCP, garantindo sincronização das métricas com o uso de locks e delegando cálculos CPU-bound para um pool de processos.

Introdução

Aplicações distribuídas demandam coordenação cuidadosa entre componentes concorrentes. Ao combinar múltiplas threads de atendimento com um pool de processos dedicado ao cálculo de números primos, garantimos escalabilidade horizontal para o servidor e isolamento de computações intensivas. O cliente utiliza comunicação síncrona sobre TCP com mensagens JSON, mantendo um thread dedicado para recebimento das respostas.

Metodologia

A arquitetura utiliza um servidor TCP multi-thread que aceita conexões e instancia uma thread por cliente. As requisições são decodificadas em mensagens estruturadas e encaminhadas para um componente Dispatcher que mantém estatísticas protegidas por um lock. Cada cálculo é executado por um ProcessPoolExecutor, permitindo verdadeiro paralelismo em múltiplos núcleos. A Figura 1 resume o fluxo de comunicação e cooperação entre threads e processos.

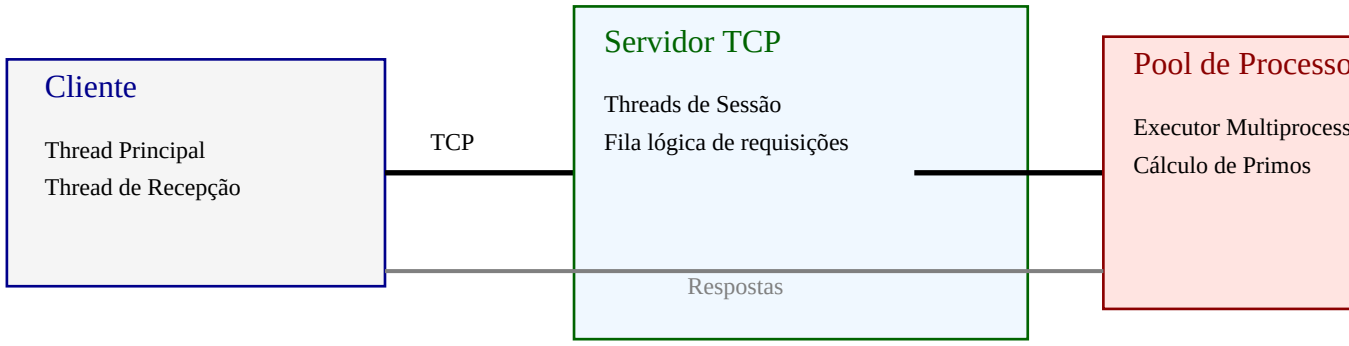

Figura 1 – Fluxo geral da solução distribuída.

Resultados

Foram conduzidos testes funcionais por meio de múltiplos clientes simultâneos. A Tabela 1 ilustra comandos executados e respostas recebidas do servidor, comprovando o funcionamento concorrente e a manutenção de métricas em tempo real.

Comando	Descrição	Resposta (resumida)

prime 104729	Verificação de primo grande	{"is_prime": true}
range 1 30	Listagem de primos	{"count": 10, ...}
count 1 100000	Contagem intensiva	{"count": 9592}
stats	Métricas do servidor	{"total_requests": 42, ...}

Conclusão

O projeto demonstra como a combinação de threads, processos e mecanismos de sincronização pode ser aplicada para construir serviços distribuídos robustos. Além de atender a múltiplos clientes simultaneamente, a solução garante paralelismo efetivo para tarefas CPU-bound, apresentando métricas claras do comportamento do sistema.

Referências

Python Software Foundation. *Python 3 Documentation: multiprocessing — Process-based parallelism*.