

Análise de desempenho de Servidor Codificado em Python em Máquinas virtuais sob Diferentes Cargas de Trabalho

Ivalcleb Leônico Benigno de Souza (IFPB, Campus João Pessoa), Luciana Pereira Oliveira (IFPB, Campus João Pessoa)

E-mails: ivalcleb.leoncio@academico.ifpb.edu.br, luciana.oliveira@ifpb.edu.br.

Área de conhecimento:(Tabela CNPq): 1.03.03.04-9 Sistemas de Informação.

Palavras-Chave: python; virtualização; máquinas virtuais; desempenho de sistemas; carga de trabalho; gerenciamento de recursos.

1 Introdução

O avanço das tecnologias distribuídas e a crescente demanda por soluções eficientes em rede tornam essencial a compreensão do desempenho de aplicações desenvolvidas para ambientes cliente-servidor. Nesse cenário, a linguagem Python destaca-se, segundo (Pluralsight, 2024) e (OLIVEIRA et al., 2021), por ser atualmente a linguagem de programação mais utilizada, o que reforça sua relevância tanto no meio acadêmico quanto no mercado profissional.

Diante disso, este trabalho propõe a análise de desempenho de um servidor implementado em Python, executado em ambiente virtualizado e submetido a diferentes cargas de trabalho geradas por um cliente. Para simular um cenário de rede realista, adotou-se a arquitetura cliente-servidor com os dois componentes distribuídos em máquinas virtuais distintas, ambas configuradas no Oracle VirtualBox — atualmente a plataforma de virtualização mais utilizada para testes e desenvolvimento, conforme aponta (BrowserStack, 2024).

A separação entre cliente e servidor em ambientes virtualizados oferece maior controle sobre a infraestrutura e permite isolar variáveis que influenciam o desempenho. Além disso, o uso de máquinas virtuais proporciona uma plataforma flexível para simulação, sendo amplamente adotado em contextos de desenvolvimento, testes e produção devido à sua capacidade de replicar cenários diversos em um único equipamento físico.

O problema central investigado é compreender o impacto do aumento da carga de trabalho sobre o desempenho do servidor, identificando gargalos e limitações da infraestrutura virtualizada. Justifica-se esse estudo pela crescente adoção de ambientes virtualizados, onde a otimização do uso de recursos pode influenciar diretamente a eficiência dos processos computacionais.

Este estudo busca mensurar métricas como uso da CPU, número de trocas de contexto (voluntárias e involuntárias) e taxas de leitura e escrita em disco (R/W), com o intuito de fornecer uma análise técnica que auxilie na tomada de decisões quanto à alocação de recursos em ambientes virtuais.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta os materiais e métodos utilizados no experimento; a Seção 3 discute os resultados obtidos; e a Seção 4 traz as considerações finais e sugestões para trabalhos futuros.

2 Materiais e Métodos

Para realizar este experimento, foram utilizadas duas máquinas virtuais configuradas no software Virtual-Box (Versão 6.1). Ambas seguiram o modelo de arquitetura cliente-servidor, no qual uma VM executa o código cliente_threads.py e o serv_threads.py, juntamente com o script script_automatizar.sh que observa e registra o comportamento do servidor. Os códigos estão em <https://github.com/1valcl3b/experimento-sistemas-operacionais>. Cada máquina virtual foi configurada com 2 GB de RAM, 1 núcleo de processamento e 4 GB de armazenamento, utilizando o sistema operacional Debian 12.

A VM cliente foi responsável por gerar e enviar mensagens em três cenários distintos: Carga Baixa (duas threads com mensagens enviadas com intervalos regulares); Carga Média (10 threads com mensagens enviadas em intervalos regulares) e Carga Alta (50 threads com mensagens enviadas sem determinar um intervalo regular).

Para garantir a automação e confiabilidade da coleta de dados, foi desenvolvido um script em Bash que executava automaticamente 10 repetições para cada nível de carga de trabalho (baixa, média e alta), armazenando os resultados em arquivos CSV para posterior análise estatística. O script utiliza comandos do sistema operacional para monitorar métricas como uso da CPU, número de trocas de contexto e operações de entrada e saída. Durante a execução dos testes, o servidor foi mantido em background ().

As medições foram realizadas da seguinte forma: Consumo cpu com o comando `pidstat -u -p $(pgrep -f servidor.py)`; Trocas de contexto com `cat /proc/PID/status`; e Leitura/Escrita em Disco com `iostat -dx`

Os dados coletados foram organizados em tabelas e posteriormente analisados para identificar padrões de comportamento da máquina virtual sob diferentes cargas de trabalho. A seção a seguir apresentará os resultados obtidos e suas implicações para a otimização de ambientes virtualizados.

3 Resultados e Discussão

Para avaliar o comportamento do sistema sob diferentes cargas de trabalho, foram analisados três aspectos principais: uso da CPU, desempenho de R/W (leitura e escrita) e trocas de contexto. Os dados foram coletados a partir da execução repetida das cargas e registrados para posterior análise. A seguir, os resultados obtidos.

O uso da CPU foi medido em diferentes cargas de trabalho: baixa, média e alta. A Figura 1 apresenta a média do uso da CPU ao longo das repetições. Observa-se que, conforme esperado, o consumo da CPU cresce proporcionalmente ao aumento da carga. Em cargas baixas, o sistema mantém um uso reduzido do processador, enquanto para cargas médias e altas há um crescimento significativo no consumo de recursos.

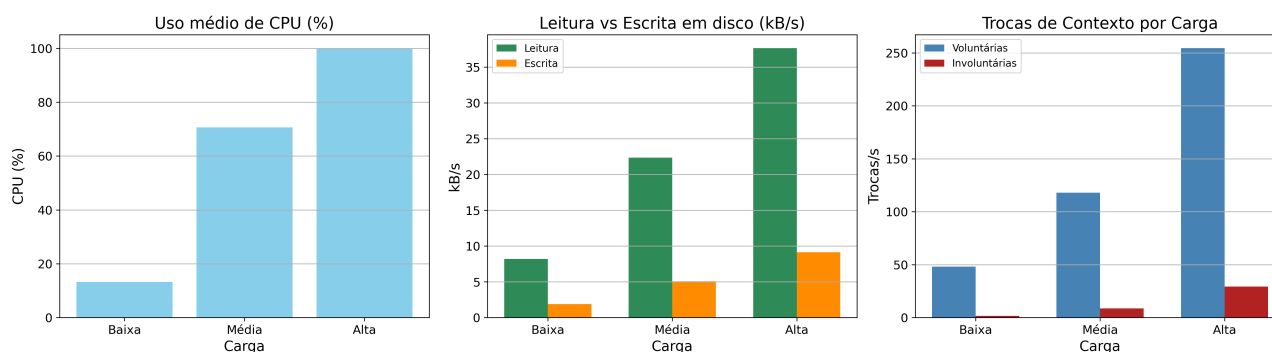


Figura 1: Uso da CPU %, Média da leitura e escrita KB/s em disco e Trocas de contextos por carga.

Nas repetições, ocorreram variações nos resultados, o que evidencia a influência de outros fatores no ambiente, como processos em background e políticas do escalonador.

A Figura 1 também apresenta a taxa média de leitura e escrita (em KB/s) para cada nível de carga de trabalho. Nota-se que, de forma geral, a leitura apresenta um volume maior do que a escrita, especialmente em cargas altas. Esse comportamento pode ser explicado pelo perfil das tarefas executadas, que exigem maior acesso a dados armazenados do que a gravação de novas informações.

Para cargas baixas, as operações de R/W são mínimas, indicando pouca movimentação de dados. Já em cargas médias e altas, a diferença entre os volumes de leitura e escrita torna-se mais evidente.

A quantidade de trocas de contexto voluntárias e involuntárias foi monitorada durante a execução dos experimentos. Observa-se que, à medida que a carga aumenta, há um crescimento significativo no número de

trocas de contexto voluntárias, indicando maior ocorrência de eventos de R/W e do que a efetiva necessidade de preempção por parte do escalonador.

Os valores observados demonstram que, para cargas altas, o número de trocas de contexto atinge picos elevados, refletindo o impacto do compartilhamento do processador entre múltiplos processos. Já para cargas baixas, a necessidade de preempção é reduzida, resultando em menor quantidade de trocas involuntárias.

Embora os testes tenham sido repetidos 10 vezes para cada carga de trabalho, os resultados foram apresentados por meio de médias. Como sugestão para estudos futuros, recomenda-se a utilização de boxplots para evidenciar variações, valores extremos e fornecer uma análise estatística mais robusta.

4 Considerações Finais

Este estudo teve como objetivo analisar o desempenho do servidor escrito em python em uma máquina virtual de baixa configuração sob diferentes níveis de carga de trabalho, considerando métricas como uso de CPU, trocas de contexto e taxa de R/W.

Os dados evidenciaram crescimento nas métricas analisadas conforme a carga aumentava. Esse comportamento reforça a importância de se compreender o gerenciamento de CPU em ambientes virtualizados, sobretudo em contextos de uso intensivo.

No que se refere às operações de entrada e saída (R/W), verificou-se um aumento significativo nas taxas de leitura do que escrita em cenários, o que aponta para possíveis gargalos na aplicação responsável pelo monitoramento, já que os resultados estavam sendo gravados na mesma VM do servidor.

As trocas de contexto, por sua vez, cresceram de forma notável em cargas elevadas, especialmente as involuntárias devido ao grande número de eventos R/W na placa de rede como também na leitura e escrita em disco.

Conclui-se que seria importante buscar uma forma de medir e observar o servidor isolando o script e o código do servidor, bem como a análise de outras métricas, como faltas de página e uso da memória RAM, bem como a replicação do experimento em outras distribuições do Linux e configurações da VM.

Apesar de o uso da memória RAM não ter sido monitorado neste experimento, sua inclusão pode enriquecer a análise, dado que o consumo de memória é uma métrica fundamental na avaliação de desempenho em ambientes virtualizados. Em trabalhos futuros, pretende-se incorporar esse monitoramento de forma integrada às demais métricas.

Outra perspectiva relevante para trabalhos futuros envolve a execução do experimento em múltiplas máquinas virtuais e em diferentes plataformas de virtualização, com configurações idênticas, a fim de comparar o desempenho entre ambientes.

Referências

BrowserStack. *Online Virtual Machine for Free*. 2024. Acesso em: 21 maio 2025. Disponível em: <<https://www.browserstack.com/guide/online-virtual-machine-free>>.

OLIVEIRA, L. P. et al. Título do capítulo – *exemplo: “análise ...”*. In: BAROLLI, L.; WOUNGANG, I.; ENOKIDO, T. (Ed.). *Advanced Information Networking and Applications, Proceedings of the 35th International Conference on Advanced Information Networking and Applications (AINA-2021), Volume 1*. [S.l.]: Springer, 2021, (Lecture Notes in Networks and Systems, v. 225). p. 383–392.

Pluralsight. *Top 10 Programming Languages in 2025*. 2024. Acesso em: 21 maio 2025. Disponível em: <<https://www.pluralsight.com/resources/blog/upskilling/top-programming-languages-2025>>.