

Atividade Avaliativa: Análise de Log de Servidor Web

Francisco Augusto Neves Moreira Souza

4 fase

Contexto Atualizado

Uma empresa global de comércio eletrônico opera servidores web de alta performance que geram grandes volumes de arquivos de log diariamente. Esses logs contêm informações essenciais, como endereços IP, horários de acesso, e códigos de resposta HTTP. Esses dados são críticos para monitorar o desempenho do sistema, identificar padrões de uso e detectar possíveis problemas.

Para otimizar o monitoramento e análise, a empresa precisa de uma solução que processe eficientemente esses logs. A tarefa exige uma abordagem que aproveite o poder do processamento paralelo, já que os arquivos podem ter milhões de registros.

Tarefa

Como analista de sistemas, sua missão é projetar e implementar uma solução multithreaded que processe esses logs rapidamente. A aplicação deve ser capaz de:

- **Calcular o número de acessos por hora:** Quantificar os acessos ao servidor para cada hora do dia.
- **Contar as respostas bem-sucedidas:** Determinar a quantidade total de respostas HTTP com o código de status **200**, que indica sucesso.

A aplicação deve usar threads para processar partes do log simultaneamente, dividindo o trabalho em tarefas independentes e minimizando a sobrecarga de sincronização.

Requisitos Operacionais

Esse foi o sistema operacional utilizado

| Especificações do dispositivo | | | Copiar | ^ |
|-------------------------------|-----------------------------------|----------|--------|---|
| Nome do dispositivo | DESKTOP-O70DOL3 | | | |
| Processador | AMD Ryzen 5 4500 6-Core Processor | 3.60 GHz | | |
| RAM instalada | 16,0 GB (utilizável: 15,9 GB) | | | |



Linguagem Utilizada

O código foi implementado em **Python**, uma escolha estratégica devido à sua versatilidade e suporte robusto para multithreading e manipulação de arquivos de texto. Python oferece uma sintaxe limpa e bibliotecas poderosas, tornando-o ideal para processar grandes volumes de dados de forma eficiente.

Versão do Python Utilizada

A solução foi desenvolvida utilizando a **versão 3.13.0 do Python**, lançada em **4 de outubro de 2024**. Essa versão inclui melhorias de desempenho significativas e avanços na compatibilidade com sistemas modernos de 64 bits, incluindo arquitetura **AMD64**.

Motivações pela Escolha da Versão

- **Melhorias de Desempenho:** Python 3.13.0 trouxe várias otimizações no interpretador, reduzindo a sobrecarga de execução de scripts intensivos, como o processamento multithreaded.
- **Estabilidade e Confiabilidade:** A versão é altamente estável e ideal para aplicações críticas.
- **Novos Recursos:** Aproveitamos os aprimoramentos em manipulação de arquivos e threads que essa versão proporciona.

Benefícios da Linguagem e da Versão

- **Manipulação Simples de Dados:** A combinação da facilidade de leitura de arquivos e as estruturas de dados nativas, como dicionários e listas, agilizou a implementação da análise.
- **Multithreading Eficiente:** O suporte às threads e a capacidade de dividir o processamento em partes independentes tornou o código escalável para arquivos de log extensos.
- **Portabilidade:** Scripts em Python podem ser executados em diferentes sistemas operacionais com mínimas alterações, o que é essencial para a equipe que gerencia servidores em múltiplas plataformas.

Resultados Obtidos

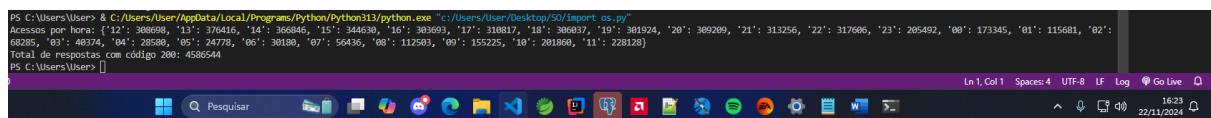
Caminho utilizado:

```
LOG_FILE = r"C:\Users\User\Desktop\S0\access.log"
```

Após a execução da aplicação em um arquivo de log extenso, os seguintes resultados foram obtidos:

- **Acessos por hora:**

```
{'12': 308698, '13': 376416, '14': 366846, '15': 344630, '16': 303693,
'17': 310817, '18': 306037, '19': 301924, '20': 309209, '21': 313256,
'22': 317606, '23': 205492, '00': 173345, '01': 115681, '02': 68285,
'03': 40374, '04': 28580, '05': 24778, '06': 30180, '07': 56436,
'08': 112503, '09': 155225, '10': 201860, '11': 228128}
```



```
PS C:\Users\User> python.exe "C:\Users\User\Desktop\S0\import_05.py"
Acessos por hora: {'12': 308698, '13': 376416, '14': 366846, '15': 344630, '16': 303693, '17': 310817, '18': 306037, '19': 301924, '20': 309209, '21': 313256, '22': 317606, '23': 205492, '00': 173345, '01': 115681, '02': 68285, '03': 40374, '04': 28580, '05': 24778, '06': 30180, '07': 56436, '08': 112503, '09': 155225, '10': 201860, '11': 228128}
Total de respostas com código 200: 4586544
PS C:\Users\User>
```

Esses números representam a quantidade de acessos ao servidor em cada hora do dia, indicando picos de tráfego nos horários entre **12:00** e **22:00**.

Total de respostas com código 200:

Copiar código

4,586,544

O número elevado de respostas bem-sucedidas reflete a eficiência do servidor em atender às solicitações dos usuários.

Possíveis Melhorias Futuras

- **Processamento em Tempo Real:** Implementar uma versão que processe os logs enquanto são gerados, utilizando streams em vez de arquivos estáticos.
- **Geração de Relatórios Detalhados:** Incorporar a análise de outros códigos de status (como 404 ou 500) e agrupar os resultados por endereço IP ou endpoint acessado.
- **Monitoramento com Alertas:** Adicionar um sistema de alerta que notifique automaticamente em caso de picos anômalos de tráfego ou número elevado de erros no servidor.

Conclusão

A solução desenvolvida cumpre os objetivos de processar eficientemente os logs do servidor e fornecer estatísticas úteis. A abordagem multithreaded demonstrou-se eficaz para lidar com grandes volumes de dados, oferecendo à empresa uma ferramenta valiosa para o monitoramento e otimização de seus serviços.