

Aluno: Felipe Biava Favarin

Turma: 4º Fase de Ciência da Computação

Link do Github (repositório público):

https://github.com/felipebfava/S_O

https://github.com/felipebfava/S_O/blob/main/README.md

https://github.com/felipebfava/S_O/blob/main/processo.py

Atividade Avaliativa

Análise de Log de Servidor Web

Contexto: Uma empresa de tecnologia gerencia vários servidores web que geram grandes arquivos de log diariamente. Esses logs contêm informações importantes, como endereços IP dos usuários, datas de acesso, e status de resposta. A empresa deseja processar esses logs para gerar estatísticas diárias, como o número de acessos por hora e a quantidade de respostas bem-sucedidas.

Tarefa: Como analista de sistemas, você deve implementar uma solução em multithreading para processar os arquivos de log de forma eficiente. Cada thread deve ser responsável por processar uma parte diferente do log, computando estatísticas independentes sem a necessidade de sincronização complexa.

Requisitos:

1. Baixe o arquivo de log disponível em <http://cti.videira.ifc.edu.br/~angelita/access.log>
2. Divida o arquivo de log em várias partes e atribua uma thread para processar cada parte.
3. Cada thread deve calcular o número de acessos por hora e o número de respostas com código de status 200 (OK) em sua parte do log.
4. As threads não devem compartilhar dados entre si, eliminando a necessidade de semáforos ou outras formas de sincronização.
5. Após todas as threads concluírem o processamento, os resultados devem ser combinados para gerar estatísticas finais para o dia.

Objetivo: O objetivo é usar threads para acelerar o processamento dos arquivos de log, permitindo a análise rápida de grandes volumes de dados sem riscos de deadlocks ou a complexidade de gerenciar dados compartilhados.

Requisitos Operacionais:

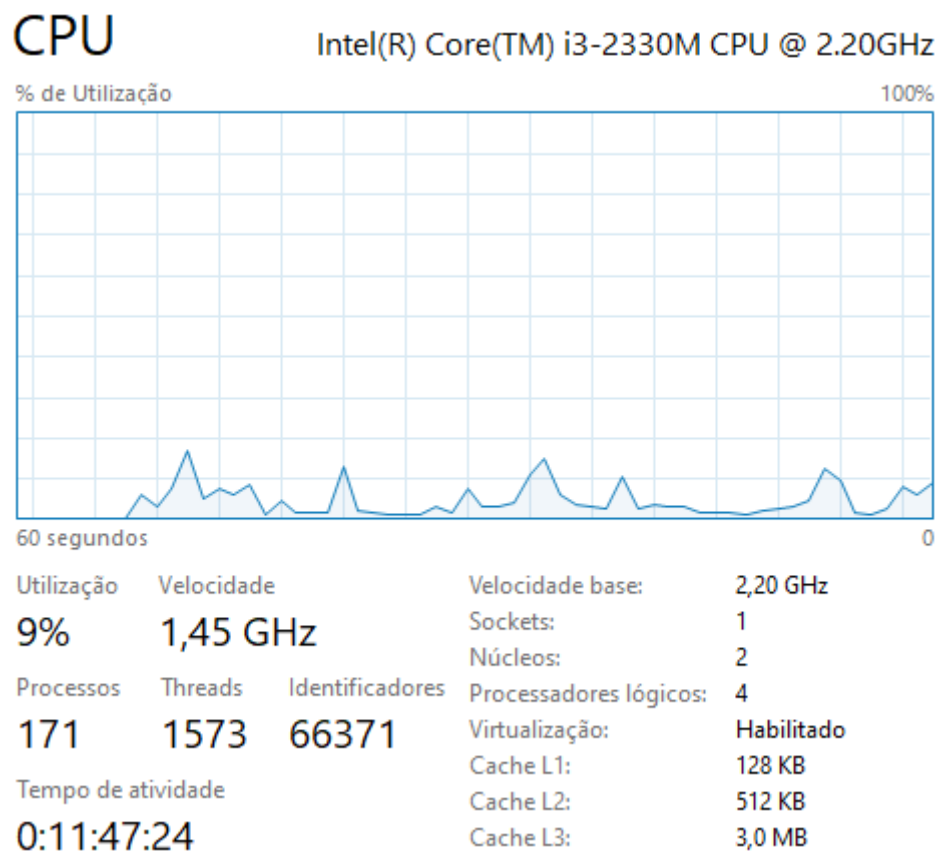
Notebook usado:

Intel(R) Core(TM) i3-2330M CPU @ 2.20GHz

Núcleos: 2

Processadores lógicos: 4

Imagem Referência:



Fonte: Próprio autor

Linguagem usada:

Código feito na linguagem Python, por fornecer uma melhor maneira de se trabalhar com threads.

Versão do Python utilizada:

Python 3.11.6 para 64 bit, arquitetura AMD64 no win32

Versão lançada em 2 de outubro de 2023.

Bibliotecas usadas:

import os: Para operações de interação com o sistema, como acessar o caminho do arquivo.

import threading: Para criar e gerenciar threads, em execução paralela, permitindo que o código processe os logs mais rapidamente.

Caminho do arquivo:

o arquivo foi pré-instalado em:

<http://cti.videira.ifc.edu.br/~angelita/access.log>

ATENÇÃO: O arquivo demora para ser instalado, talvez o seu navegador reclame do tamanho do arquivo (1,56 GB), o bloco de notas também não aguenta o tamanho do arquivo, utilize outro programa para isso, como o Visual Studio Code.

Versão do Visual Studio Code utilizada: 1.93.1

O caminho do arquivo pode ser visto na seguinte linha de código:

```
log_file = r"C:\Users\User\OneDrive\Área de Trabalho\threads\access.log"
```

Resultado Final:

Após cerca de 5 minutos, temos o seguinte resultado:

```
Acessos por hora: {'12': 308698, '13': 376416, '14': 366846, '15': 344630, '16': 303693, '17': 310817, '18': 306037, '19': 301924, '20': 309209, '21': 313256, '22': 316864, '10': 201860, '11': 228128, '23': 205492, '00': 173345, '01': 115681, '02': 68285, '03': 40374, '04': 28580, '05': 24778, '06': 30180, '07': 56436, '08': 112503, '09': 155225}
Total de respostas com código 200: 4585860
```

Perfeitamente funcional, feito da maneira certa, mostrando resultados com os dados desejados.