

ADS – Prof. Caio Souza  
2a. Lista de Exercícios

---

- 1) Escreva um shell script que colete informações a cada 10 segundos sobre o seu navegador, coletando quantos % da CPU, ele está consumindo, qual CPU ele está executando, e quantos kbytes por segundo (kB wr/s) seu navegador está escrevendo no disco (dica: pidstat). Nesse mesmo script colete informações do estado geral da sua máquina (total da RAM que está livre, total da RAM usada, total da RAM reservada para buffer e para cache). Salve toda a coleta em um arquivo do tipo .csv.

O SO utilizado é um Pop Os! (Base Ubuntu)

1. Instalação de dependências

```
sudo apt update
sudo apt install sysstat
```

2. Criação e execução do .sh

Editor:

```
nano coleta_chrome.sh
```

Script:

```
#!/bin/bash

if [ -z "$1" ]; then
    echo "Uso: $0 <PID_DO_CHROME>"
    exit 1
fi

PID=$1
CSV_FILE="chrome_metrics.csv"

# Cabeçalho do CSV
echo
"timestamp,pid,cpu_percent,cpu_num,read_kBps,write_kBps,ram_livre_kB,ram_usada_kB,ram_buffer_kB,ram_cache_kB" > "$CSV_FILE"

# Captura inicial para calcular deltas depois
prev_read_io=$(grep "^read_bytes:" /proc/$PID/io | awk '{print $2}')
prev_write_io=$(grep "^write_bytes:" /proc/$PID/io | awk '{print $2}')

while true; do
    timestamp=$(date "+%Y-%m-%d %H:%M:%S")

    # CPU (usando pidstat)
```

```

pidstat_out=$(pidstat -h -p $PID 1 1 | grep -v "Linux" | grep -E
"^[:space:]*$PID")
cpu_percent=$(echo "pidstat_out" | awk '{print $7}')
cpu_num=$(ps -o psr= -p $PID | tr -d ' ')

# IO atual
read_io=$(grep "^read_bytes:" /proc/$PID/io | awk '{print $2}')
write_io=$(grep "^write_bytes:" /proc/$PID/io | awk '{print $2}')

# Calcular diferenças com segurança usando bc
delta_read=$(echo "$read_io - $prev_read_io" | bc)
delta_write=$(echo "$write_io - $prev_write_io" | bc)

if [[ $(echo "$delta_read >= 0" | bc) -eq 1 ]]; then
    read_kBps=$(echo "scale=2; $delta_read / 1024 / 10" | bc)
else
    read_kBps=0
fi

if [[ $(echo "$delta_write >= 0" | bc) -eq 1 ]]; then
    write_kBps=$(echo "scale=2; $delta_write / 1024 / 10" | bc)
else
    write_kBps=0
fi

# Memória
mem_out=$(free -k | grep Mem)
ram_total_kB=$(echo $mem_out | awk '{print $2}')
ram_livre_kB=$(echo $mem_out | awk '{print $4}')
ram_buffer_kB=$(echo $mem_out | awk '{print $6}')
ram_cache_kB=$(echo $mem_out | awk '{print $7}')
ram_usada_kB=$(echo "$ram_total_kB - $ram_livre_kB - $ram_buffer_kB -
$ram_cache_kB" | bc)

# Escreve linha no CSV
echo
"$timestamp,$PID,$cpu_percent,$cpu_num,$read_kBps,$write_kBps,$ram_livre
_kB,$ram_usada_kB,$ram_buffer_kB,$ram_cache_kB" >> "$CSV_FILE"

# Atualiza valores anteriores
prev_read_io=$read_io
prev_write_io=$write_io

sleep 10
done

```

Permissões:

```
chmod +x coleta_chrome.sh
```

A execução será apenas sobre o processo pai:

```
./coleta_chrome.sh 3218
```

Métricas resultantes após 42 ciclos (4,33min): [CSV com resultados](#)

- 2) Escreva um shell script que utiliza o tcpdump para capturar 150 pacotes do tipo UDP indicando a porta 53 e o endereço IP da página de pesquisa da Google (Abra o navegador execute algumas pesquisas durante a coletas). Em seguida grave todas as informações em um arquivo do tipo pcap.

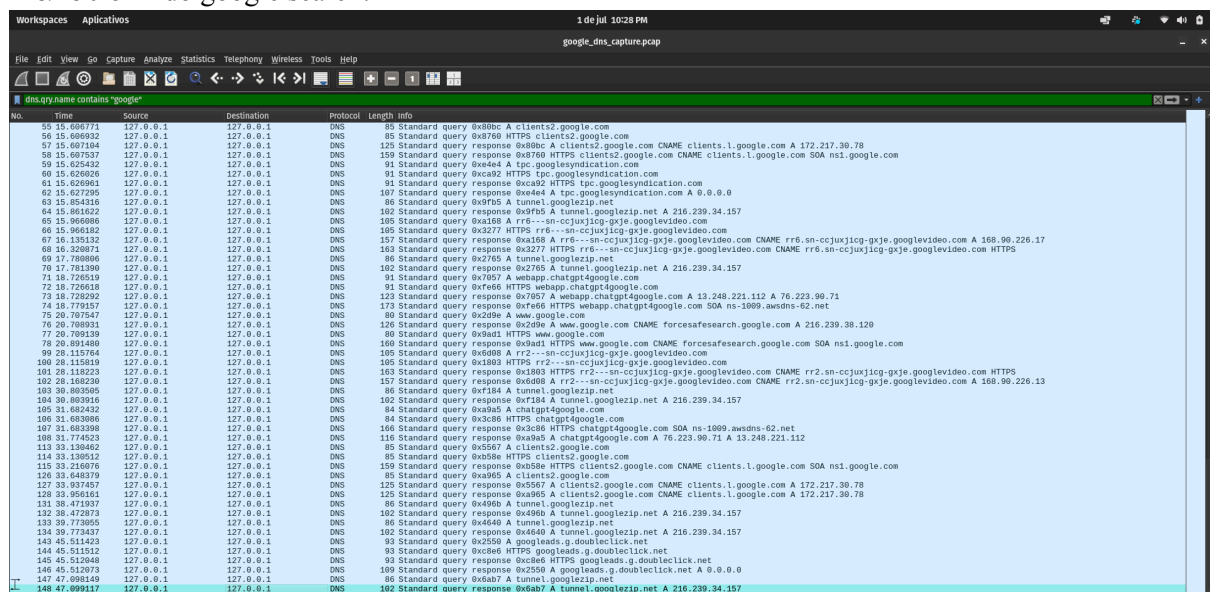
Primeiramente, acionamos o sniffing para captura de de 150 pacotes referenciando a porta UDP:

```
sudo tcpdump -i any udp port 53 -c 150 -w google_dns_capture.pcap
```

A partir daí a captura irá iniciar até alcançar no tráfego 150 pacotes:

```
valcann@pop-os:~$ sudo tcpdump -i any udp port 53 -c 150 -w google_dns_capture.pcap
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
150 packets captured
304 packets received by filter
0 packets dropped by kernel
```

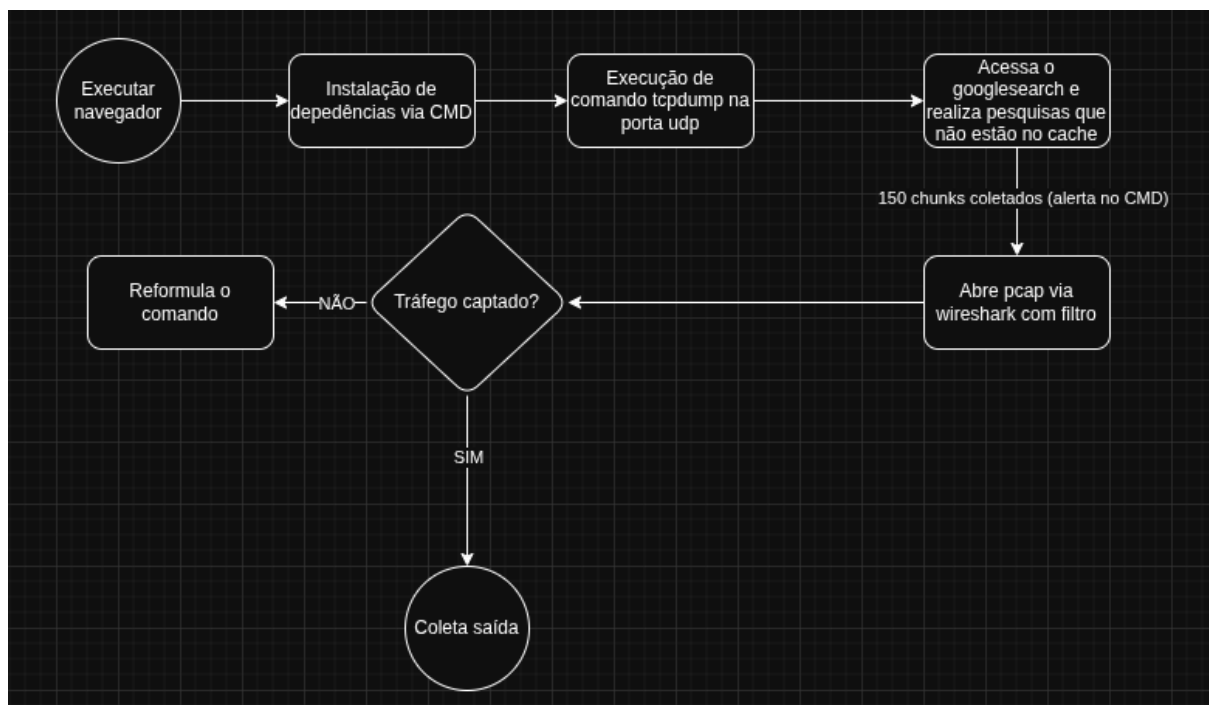
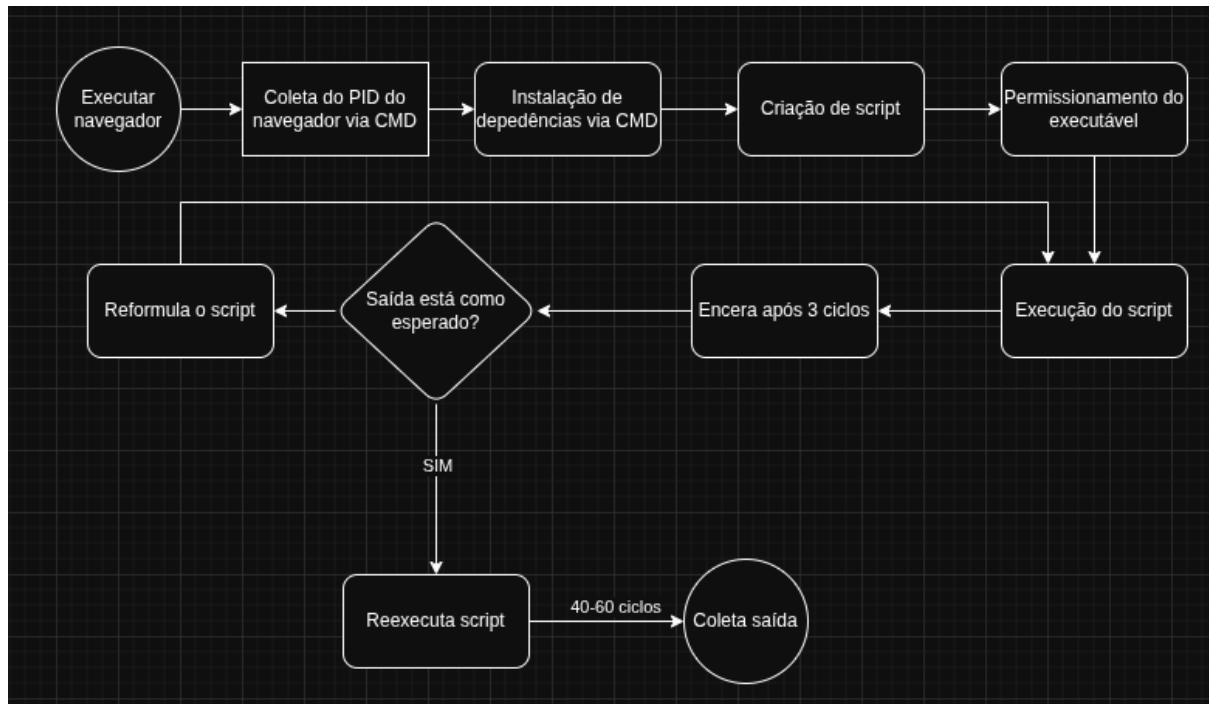
Tendo encerrado, basta utilizarmos o Wireshark (open source software para sniffing). Faremos uso do filtro 'dns.qry.name contains "google"' após abrimos o .pcap salvo. Tendo em vista que durante esse curto período realizei muitas pesquisas (tais fora do cache), a captura de pacotes envolve 148/150 o IP do google search:



No.	Time	Source	Destination	Protocol	Length	Info
55	15.606771	127.0.0.1	127.0.0.1	DNS	85	Standard query 0x80bc A clients2.google.com
56	15.609332	127.0.0.1	127.0.0.1	DNS	85	Standard query 0x8760 HTTPS clients2.google.com
57	15.607184	127.0.0.1	127.0.0.1	DNS	125	Standard query response 0x80bc A clients2.google.com CNAME clients.l.google.com A 172.217.39.78
58	15.607537	127.0.0.1	127.0.0.1	DNS	159	Standard query response 0x7f60 HTTPS clients2.google.com CNAME clients.l.google.com SOA ns1.google.com
59	15.625432	127.0.0.1	127.0.0.1	DNS	91	Standard query 0xe4e4 A tpc.googleadsyndication.com
60	15.626026	127.0.0.1	127.0.0.1	DNS	91	Standard query 0xca92 HTTPS tpc.googleadsyndication.com
61	15.629061	127.0.0.1	127.0.0.1	DNS	91	Standard query response 0xca92 HTTPS tpc.googleadsyndication.com
62	15.627295	127.0.0.1	127.0.0.1	DNS	107	Standard query response 0xe4e4 A tpc.googleadsyndication.com A 0.0.0.0
63	15.654316	127.0.0.1	127.0.0.1	DNS	86	Standard query 0x9fb5 A tunnel.googlezip.net
64	15.861622	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0x9fb5 A tunnel.googlezip.net A 216.239.34.157
65	15.966988	127.0.0.1	127.0.0.1	DNS	105	Standard query 0xa188 A rr6--sn-cjuxjicg-gxje.googlevideo.com
66	15.966182	127.0.0.1	127.0.0.1	DNS	105	Standard query 0x3277 HTTPS rr6--sn-cjuxjicg-gxje.googlevideo.com
67	16.135132	127.0.0.1	127.0.0.1	DNS	107	Standard query response 0xa188 A rr6--sn-cjuxjicg-gxje.googlevideo.com CNAME rr6.sn-cjuxjicg-gxje.googlevideo.com A 168.90.226.17
68	16.320871	127.0.0.1	127.0.0.1	DNS	163	Standard query response 0x3277 HTTPS rr6--sn-cjuxjicg-gxje.googlevideo.com CNAME rr6.sn-cjuxjicg-gxje.googlevideo.com HTTPS
69	17.788986	127.0.0.1	127.0.0.1	DNS	86	Standard query 0x2765 A tunnel.googlezip.net
70	17.781396	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0x2765 A tunnel.googlezip.net A 216.239.34.157
71	16.726519	127.0.0.1	127.0.0.1	DNS	91	Standard query 0x7057 A webapp.chatgpt4.google.com
72	16.726618	127.0.0.1	127.0.0.1	DNS	91	Standard query 0xf6e6 HTTPS webapp.chatgpt4.google.com
73	18.728292	127.0.0.1	127.0.0.1	DNS	123	Standard query response 0xf6e7 A webapp.chatgpt4.google.com A 13.248.221.112 A 76.223.90.71
74	18.779157	127.0.0.1	127.0.0.1	DNS	173	Standard query response 0xf6e6 HTTPS webapp.chatgpt4.google.com SOA ns-1009.awsdns-62.net
75	20.707547	127.0.0.1	127.0.0.1	DNS	80	Standard query 0x2d9e A www.google.com
76	20.708931	127.0.0.1	127.0.0.1	DNS	126	Standard query response 0x2d9e A www.google.com CNAME forcesafesearch.google.com A 216.239.38.120
77	20.709139	127.0.0.1	127.0.0.1	DNS	80	Standard query 0x9ad1 HTTPS www.google.com
78	20.891486	127.0.0.1	127.0.0.1	DNS	160	Standard query response 0xad1 HTTPS www.google.com CNAME forcesafesearch.google.com SOA ns1.google.com
99	28.115764	127.0.0.1	127.0.0.1	DNS	105	Standard query response 0x6088 A rr2--sn-cjuxjicg-gxje.googlevideo.com
100	28.115819	127.0.0.1	127.0.0.1	DNS	105	Standard query 0x1803 HTTPS rr2--sn-cjuxjicg-gxje.googlevideo.com
101	28.116223	127.0.0.1	127.0.0.1	DNS	163	Standard query response 0x1803 HTTPS rr2--sn-cjuxjicg-gxje.googlevideo.com CNAME rr2.sn-cjuxjicg-gxje.googlevideo.com HTTPS
102	28.168230	127.0.0.1	127.0.0.1	DNS	157	Standard query response 0x6088 A rr2--sn-cjuxjicg-gxje.googlevideo.com CNAME rr2.sn-cjuxjicg-gxje.googlevideo.com A 168.90.226.13
103	30.802560	127.0.0.1	127.0.0.1	DNS	86	Standard query 0xf184 A tunnel.googlezip.net
104	30.803915	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0xf184 A tunnel.googlezip.net A 216.239.34.157
105	31.682432	127.0.0.1	127.0.0.1	DNS	84	Standard query 0xab5a A chatgpt4.google.com
106	31.683988	127.0.0.1	127.0.0.1	DNS	84	Standard query 0x3c86 HTTPS chatgpt4.google.com
107	31.682398	127.0.0.1	127.0.0.1	DNS	166	Standard query response 0x3c86 HTTPS chatgpt4.google.com SOA ns-1009.awsdns-62.net
108	31.774523	127.0.0.1	127.0.0.1	DNS	116	Standard query response 0xa9a5 A chatgpt4.google.com A 76.223.90.71 A 13.248.221.112
113	33.136462	127.0.0.1	127.0.0.1	DNS	85	Standard query 0x5587 A clients2.google.com
131	38.471937	127.0.0.1	127.0.0.1	DNS	89	Standard query 0xb08e HTTPS clients2.google.com
132	38.472873	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0xb08e HTTPS clients2.google.com CNAME clients.l.google.com A 172.217.39.78
133	39.773955	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0x4640 A tunnel.googlezip.net A 216.239.34.157
134	39.773437	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0x4640 A tunnel.googlezip.net A 216.239.34.157
143	45.511428	127.0.0.1	127.0.0.1	DNS	82	Standard query 0x2550 A googleads.g.doubleclick.net
144	45.511512	127.0.0.1	127.0.0.1	DNS	93	Standard query 0xc8e8 HTTPS googleads.g.doubleclick.net
145	45.512848	127.0.0.1	127.0.0.1	DNS	93	Standard query response 0xc8e8 HTTPS googleads.g.doubleclick.net
146	45.513073	127.0.0.1	127.0.0.1	DNS	109	Standard query response 0x2550 A googleads.g.doubleclick.net A 0.0.0.0
147	47.098149	127.0.0.1	127.0.0.1	DNS	86	Standard query 0x6ab7 A tunnel.googlezip.net
148	47.099117	127.0.0.1	127.0.0.1	DNS	102	Standard query response 0x6ab7 A tunnel.googlezip.net A 216.239.34.157

Arquivo .pcap: [Tráfego da segunda questão](#)

- 3) Apresente uma metodologia para o experimento realizado (fluxograma ou BPMN). Com base nas informações obtidas pela experimentação realize uma análise estatística, esta deverá incluir gráficos, e tabela com as principais medidas de tendência central e de dispersão. Se você fosse o administrador desse sistema, que tipo de sugestão daria aos tomadores de decisão? Apoie sua sugestão através da apresentação dos resultados obtidos.

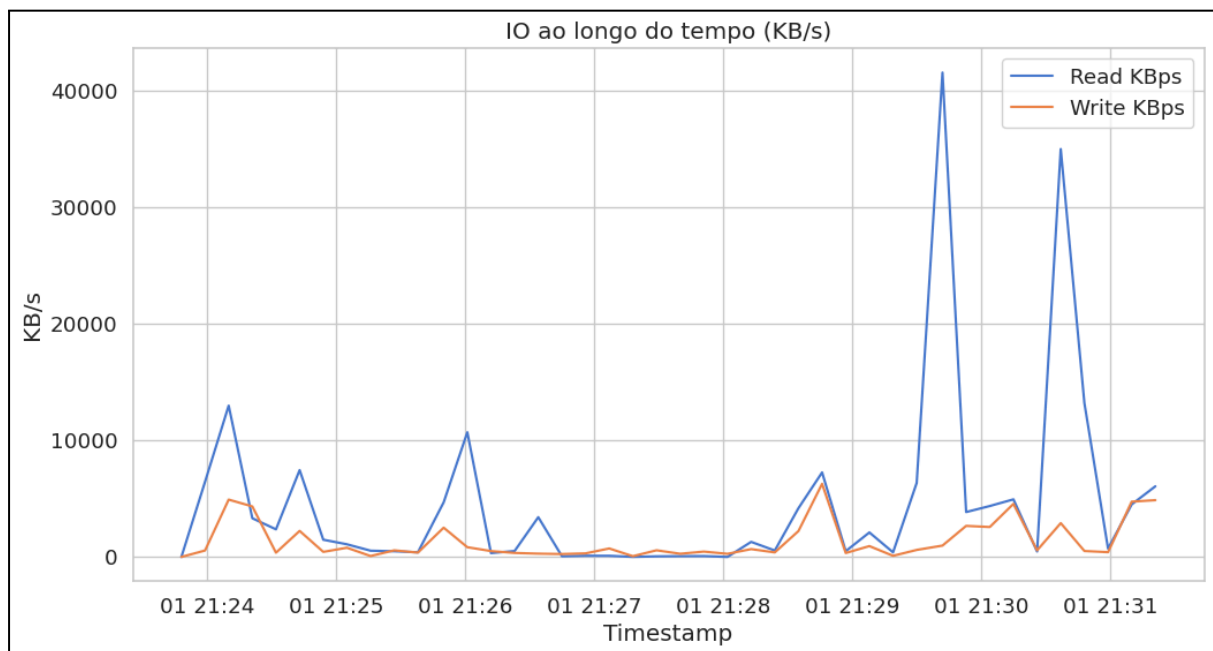


Uma pequena análise exploratória foi realizada trazendo uma estatística descritiva e sumária, tal é possível visualizar no seguinte notebook:

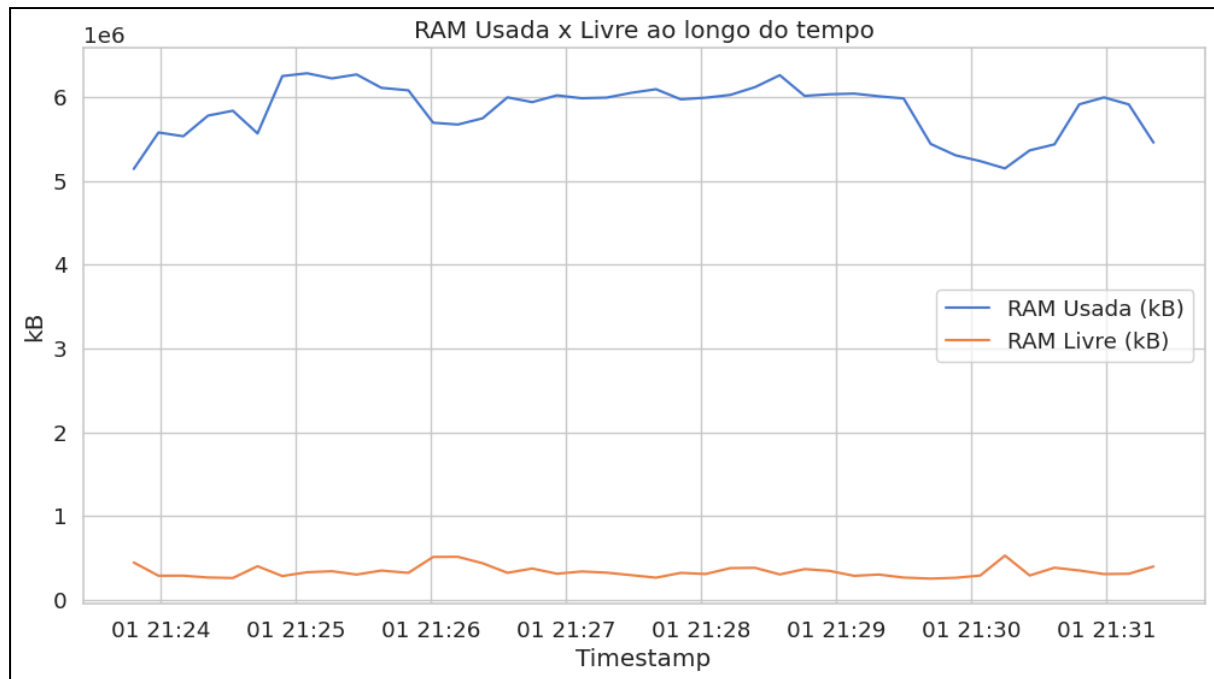
🔗 [PROFILING\\_EDA\\_PID3218\\_42C.ipynb](#) (Notebook com gráficos e medidas). Abaixo os outputs apenas mais consideráveis das células:

	mean	median	std	min	max
pid	32178.00	32178.0	0.00	32178.0	32178.0
cpu_percent	0.00	0.0	0.00	0.0	0.0
cpu_num	1.62	1.0	1.15	0.0	3.0
read_kBps	4617.91	1385.4	8408.07	0.0	41557.6
write_kBps	1388.67	566.6	1671.41	0.8	6273.2
ram_livre_kB	340300.00	324242.0	68682.37	255404.0	529724.0
ram_usada_kB	5845978.48	5985154.0	319276.50	5143160.0	6284556.0
ram_buffer_kB	1273908.19	1203396.0	187343.99	1013056.0	1644544.0
ram_cache_kB	465381.33	421278.0	141646.60	254100.0	801240.0

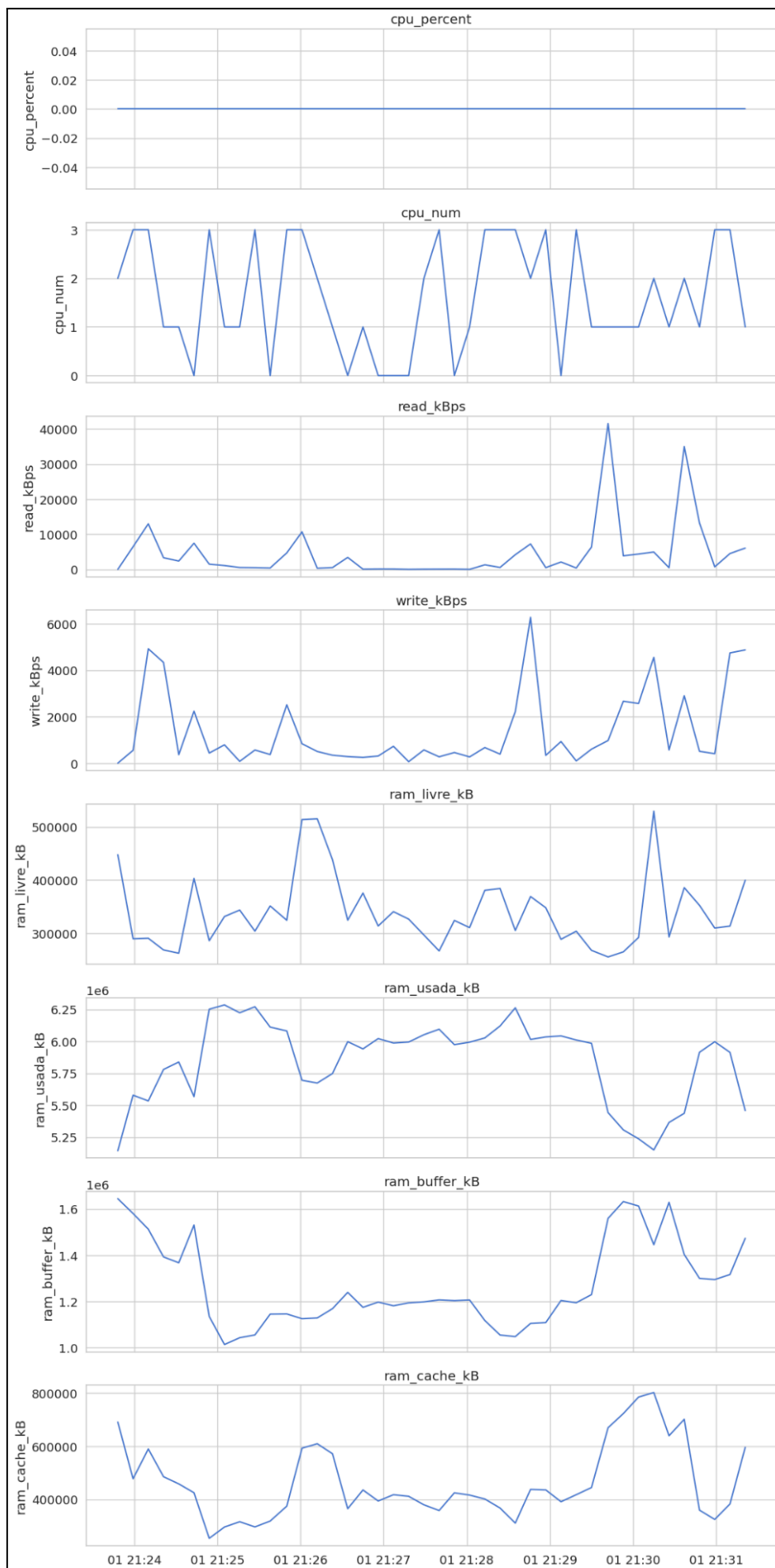
- A coluna PID é desprezível por tratar-se da tag do processo tratada. Observa-se que seu uso de CPU é desprezível, no processo tendo estado fixo em 0%, tal é um processo pai de outros módulos e normalmente possui valores próximos a 0 ou 0, não sendo interessante para o que foi solicitado analisar a soma dos filhos. O cpu\_num também é descartável por ser categórica (representando qual CPU está sendo utilizada).



- A taxa de leitura é muito maior justamente por não estar tratando de uploads apenas de pesquisas simples



- A RAM do PC está consideravelmente ocupada,, pois estava sendo consumida por várias outras aplicações, incluindo outro navegador, um SGBD, slack, file explorer e outros vários processo do chrome (filhos) em abas variadas;
- Abaixo a variação linear pelo timestamp em X de todas as medidas.



## Recomendações:

### 1. Uso de memória RAM alto:

A RAM apresentou alta ocupação durante a coleta, devido à execução simultânea de múltiplas aplicações como já dito, estando operando próximo ao limite em alguns momentos. Diante disso:

- a. Deve-se encerrar aplicações em segundo plano que não estejam sendo ativamente utilizadas;
- b. Recomenda-se considerar um upgrade de memória RAM física, caso a multitarefa seja uma necessidade constante;
- c. Prosseguir monitorando para avaliar necessidades e gargalos.

### 2. Analisar a escrita no disco:

Foram registrados picos de escrita no disco de até 72,28 KB/s, o que, embora não seja alto, pode indicar atividades frequentes de cache, logs ou uso de swap. Diante disso:

- a. Deve-se utilizar ferramentas como iotop ou iostat para identificar quais processos estão gerando maior volume de escrita;
- b. Recomenda-se configurar o Chrome para armazenar cache em disco secundário, quando disponível, utilizando parâmetros como `--disk-cache-dir="/media/HD/cache"`. Caso não possua é bom avaliar a necessidade;
- c. Deve-se verificar o uso de swap com comandos como `vmstat` ou `free -h` e, se necessário, ajustar o parâmetro `swappiness` com `sysctl vm.swappiness=10`.

### 3. Investigar o uso de CPU no processo pai do Chrome

O processo pai do Chrome apresentou uso constante de 0% da CPU. Esse comportamento é esperado, pois o processo pai do Chrome atua apenas como orquestrador. O consumo de CPU está nos processos filhos responsáveis por abas, extensões e renderização. Diante disso:

- a. Deve-se utilizar `pidstat -G chrome` ou identificar os processos filhos com `ps --ppid <PID>` para monitoramento real;
- b. Recomenda-se repetir o experimento com páginas mais exigentes, como vídeos em alta resolução ou aplicações web complexas.
- c. Sugere-se acessar `chrome://system` para examinar detalhadamente os processos ativos no navegador;
- d. Ou apenas fechar as abas em desuso. Tais apesar de esteticamente bem cismadas por grupos, ainda consomem RAM e são muitas:

### 4. Acompanhar o uso de buffer e cache

O uso de buffer cache variou entre 348 MB e 924 MB, indicando uso ativo de cache de disco para otimização de desempenho. Esse comportamento é considerado positivo e mostra que o sistema utiliza RAM disponível para acelerar operações de leitura.