

- 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 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:

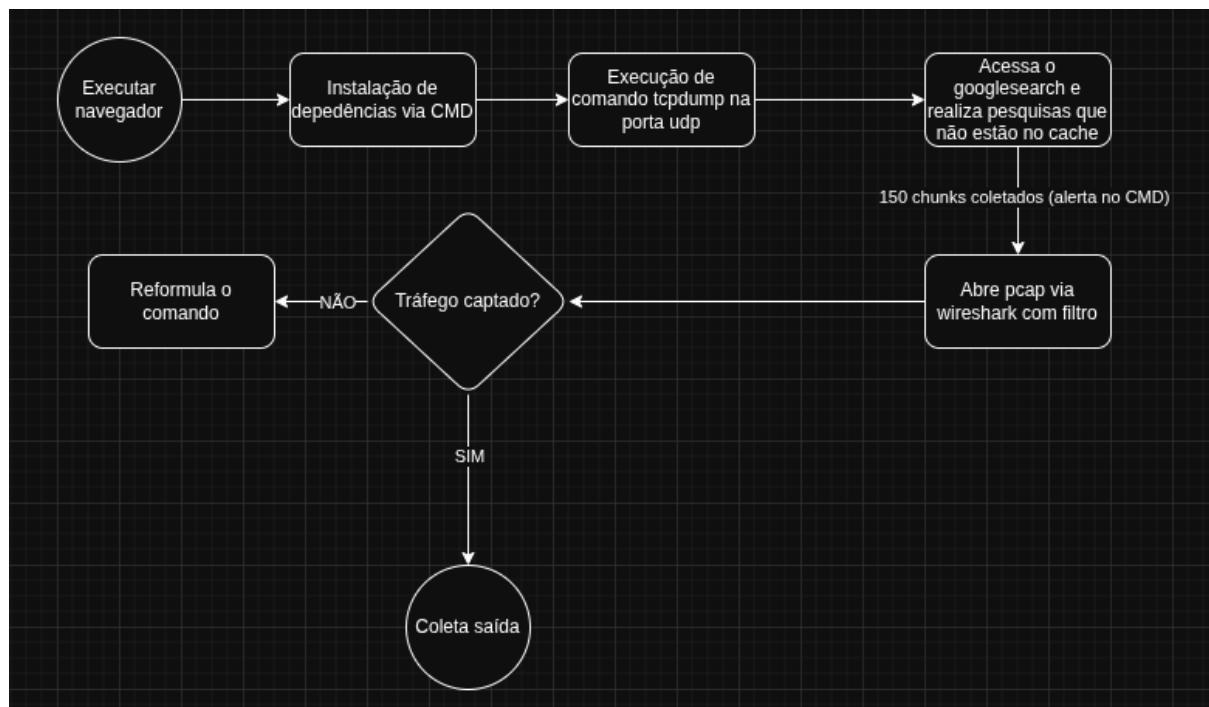
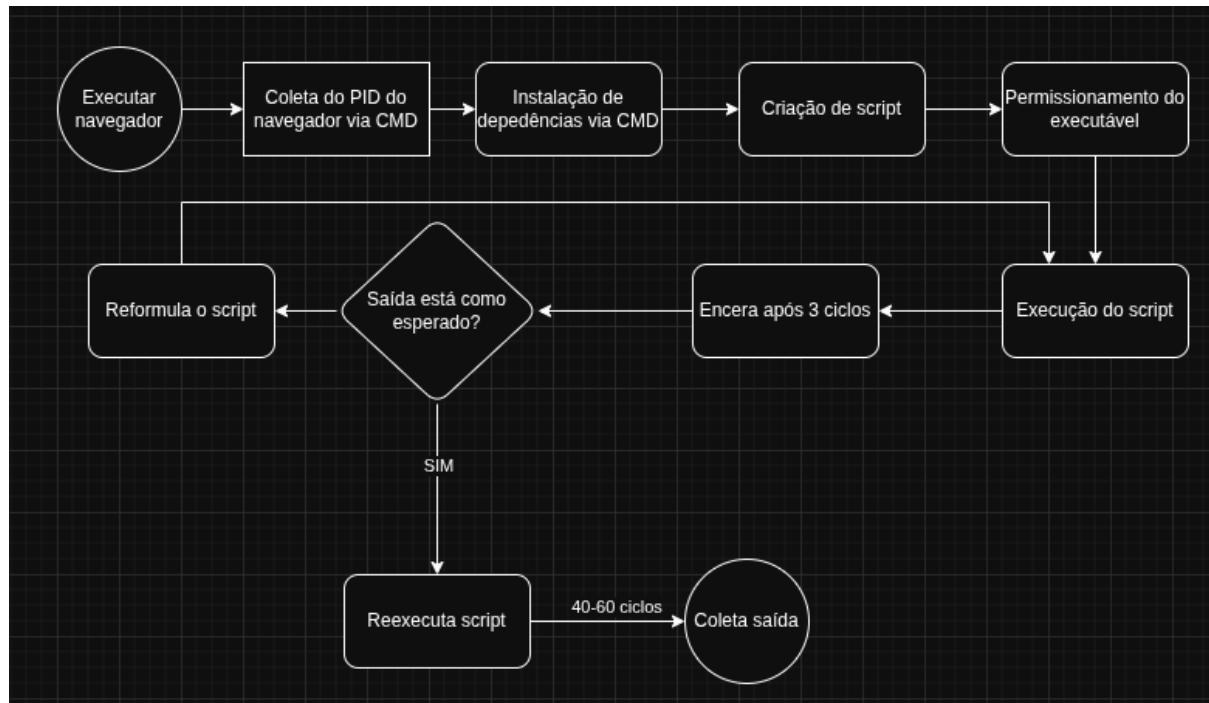
```
valcann@pop-os:~$ sudo tcpdump -i any udp port 53 -c 150 -w google_dns_capture.pcap
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
150 packets captured
304 packets received by filter
0 packets dropped by kernel
```

Tendo encerrado, basta utilizarmos o Wireshark (open source software para sniffing). Faremos uso do filtro 'dns.qry.name contains "google"' após abrirmos 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	15.696671	127.0.0.1	127.0.0.1	DNS	85 Standard query 0x89bc A clients2.google.com
56	15.696682	127.0.0.1	127.0.0.1	DNS	85 Standard query 0x8760 A clients1.google.com
57	15.697109	127.0.0.1	127.0.0.1	DNS	159 Standard query response 0x89bc A clients2.google.com CNAME clients.l.google.com A 172.217.39.78
58	15.697537	127.0.0.1	127.0.0.1	DNS	91 Standard query response 0x4e4d A tpc.googleadsindication.com
59	15.697542	127.0.0.1	127.0.0.1	DNS	91 Standard query response 0x4e4d A tpc.googleadsindication.com
60	15.697543	127.0.0.1	127.0.0.1	DNS	91 Standard query response 0x4e4d A tpc.googleadsindication.com
61	15.628961	127.0.0.1	127.0.0.1	DNS	187 Standard query response 0xca92 HTTPS tpc.googleadsindication.com A 0.0.0.0
62	15.627295	127.0.0.1	127.0.0.1	DNS	187 Standard query response 0x4e4d A tpc.googleadsindication.com A 0.0.0.0
63	15.627296	127.0.0.1	127.0.0.1	DNS	187 Standard query response 0x4e4d A tpc.googleadsindication.com A 0.0.0.0
64	15.861622	127.0.0.1	127.0.0.1	DNS	102 Standard query response 0x9f05 A Tunnel.googlezip.net A 216.239.34.157
65	15.968608	127.0.0.1	127.0.0.1	DNS	102 Standard query 0x318 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com
66	15.968609	127.0.0.1	127.0.0.1	DNS	102 Standard query 0x318 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com
67	15.968610	127.0.0.1	127.0.0.1	DNS	157 Standard query response 0xa168 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com CNAME rr6.sn-ccjuxj1cgcgxje.googlevideo.com A 168.99.226.17
68	16.320871	127.0.0.1	127.0.0.1	DNS	163 Standard query response 0x3277 HTTPS rr2.-sn-ccjuxj1cgcgxje.googlevideo.com CNAME rr6.sn-ccjuxj1cgcgxje.googlevideo.com HTTPS
69	16.320872	127.0.0.1	127.0.0.1	DNS	163 Standard query response 0x3277 HTTPS rr2.-sn-ccjuxj1cgcgxje.googlevideo.com CNAME rr6.sn-ccjuxj1cgcgxje.googlevideo.com HTTPS
70	17.781398	127.0.0.1	127.0.0.1	DNS	102 Standard query response 0x2765 A Tunnel.googlezip.net A 216.239.34.157
71	18.726519	127.0.0.1	127.0.0.1	DNS	91 Standard query 0x7057 A webapp.chatgpt4google.com
72	18.726520	127.0.0.1	127.0.0.1	DNS	102 Standard query 0x318 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com
73	18.728929	127.0.0.1	127.0.0.1	DNS	123 Standard query response 0xref57 A webapp.chatgpt4google.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 0xref57 HTTPS webapp.chatgpt4google.com SOA ns-1069.aesdns-02.net
75	18.779158	127.0.0.1	127.0.0.1	DNS	102 Standard query 0x318 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com
76	20.7698931	127.0.0.1	127.0.0.1	DNS	126 Standard query response 0x2d9e A www.google.com CNAME forcesasearch.google.com A 216.239.38.120
77	20.769139	127.0.0.1	127.0.0.1	DNS	100 Standard query 0x9ad4 HTTPS www.google.com
78	20.769140	127.0.0.1	127.0.0.1	DNS	100 Standard query 0x9ad4 HTTPS www.google.com CNAME forcesasearch.google.com SOA ns1.google.com
99	28.115764	127.0.0.1	127.0.0.1	DNS	108 Standard query response 0xb609 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com
100	28.115819	127.0.0.1	127.0.0.1	DNS	108 Standard query 0xb609 HTTPS rr2.-sn-ccjuxj1cgcgxje.googlevideo.com
101	28.115820	127.0.0.1	127.0.0.1	DNS	108 Standard query 0xb609 HTTPS rr2.-sn-ccjuxj1cgcgxje.googlevideo.com CNAME rr2.-sn-ccjuxj1cgcgxje.googlevideo.com HTTPS
102	28.168230	127.0.0.1	127.0.0.1	DNS	157 Standard query response 0x2765 A Tunnel.googlezip.net A 216.239.34.157
103	30.863504	127.0.0.1	127.0.0.1	DNS	157 Standard query 0x3f84 A tunnel.googlezip.net
104	30.863505	127.0.0.1	127.0.0.1	DNS	157 Standard query 0x3f84 A tunnel.googlezip.net
105	31.682432	127.0.0.1	127.0.0.1	DNS	84 Standard query 0x9ab3 A chatgpt4google.com
106	31.683086	127.0.0.1	127.0.0.1	DNS	84 Standard query 0x3c63 HTTPS chatgpt4google.com
107	31.683087	127.0.0.1	127.0.0.1	DNS	102 Standard query 0x318 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.com SOA ns-1069.aesdns-02.net
108	31.774523	127.0.0.1	127.0.0.1	DNS	116 Standard query response 0xa9d5 A chatgpt4google.com A 216.223.90.71 A 13.248.221.112
113	33.198450	127.0.0.1	127.0.0.1	DNS	85 Standard query 0x5567 A clients2.google.com
115	33.218976	127.0.0.1	127.0.0.1	DNS	159 Standard query response 0xb5b2 HTTPS clients2.google.com CNAME clients.l.google.com SOA ns1.google.com
126	33.648379	127.0.0.1	127.0.0.1	DNS	95 Standard query 0x9a65 A clients2.google.com
128	33.956181	127.0.0.1	127.0.0.1	DNS	125 Standard query response 0xa965 A clients2.google.com CNAME clients.l.google.com A 172.217.39.78
131	36.471937	127.0.0.1	127.0.0.1	DNS	86 Standard query 0x4969 A tunnel.googlezip.net
132	36.471938	127.0.0.1	127.0.0.1	DNS	102 Standard query 0x318 A rr2.-sn-ccjuxj1cgcgxje.googlevideo.net A 216.239.34.157
133	39.773985	127.0.0.1	127.0.0.1	DNS	86 Standard query 0x4640 A tunnel.googlezip.net
134	39.773437	127.0.0.1	127.0.0.1	DNS	102 Standard query response 0x4640 HTTPS tunnel.googlezip.net A 216.239.34.157
144	45.511512	127.0.0.1	127.0.0.1	DNS	93 Standard query 0xc8e6 HTTPS googleads.g.doubleclick.net
145	45.512048	127.0.0.1	127.0.0.1	DNS	93 Standard query response 0xc8e6 HTTPS googleads.g.doubleclick.net
147	47.099149	127.0.0.1	127.0.0.1	DNS	102 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.

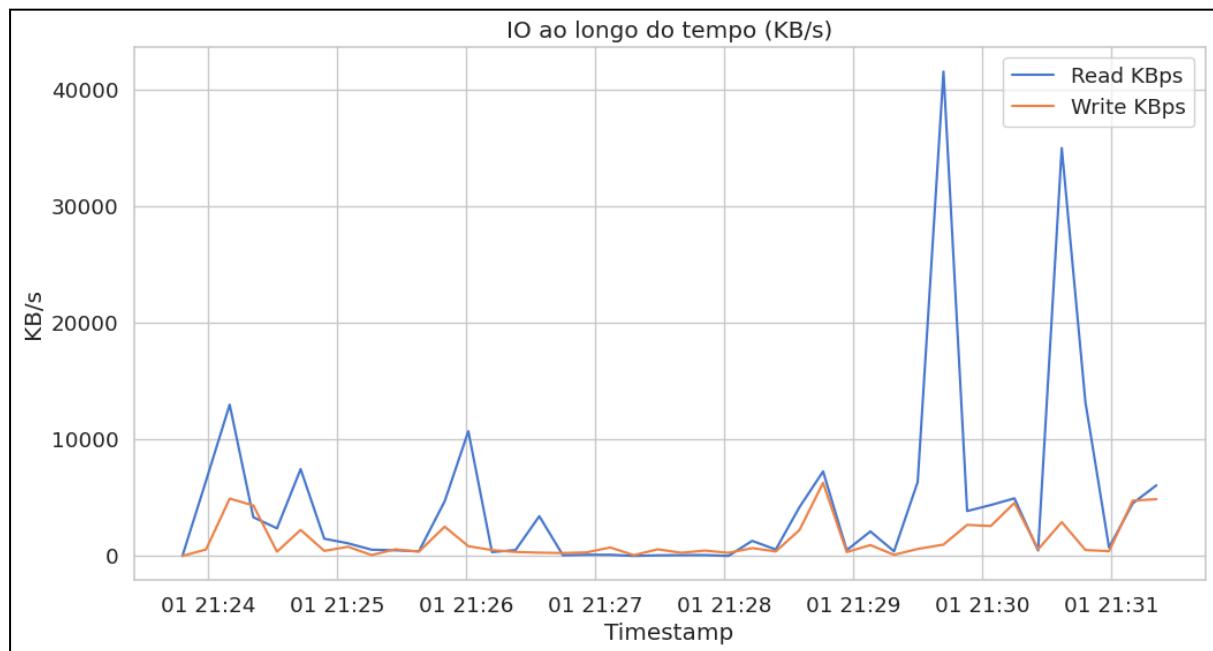


Uma pequena análise exploratória foi realizada trazendo uma estatística descritiva e sumária, tal é possível visualizar no seguinte notebook:

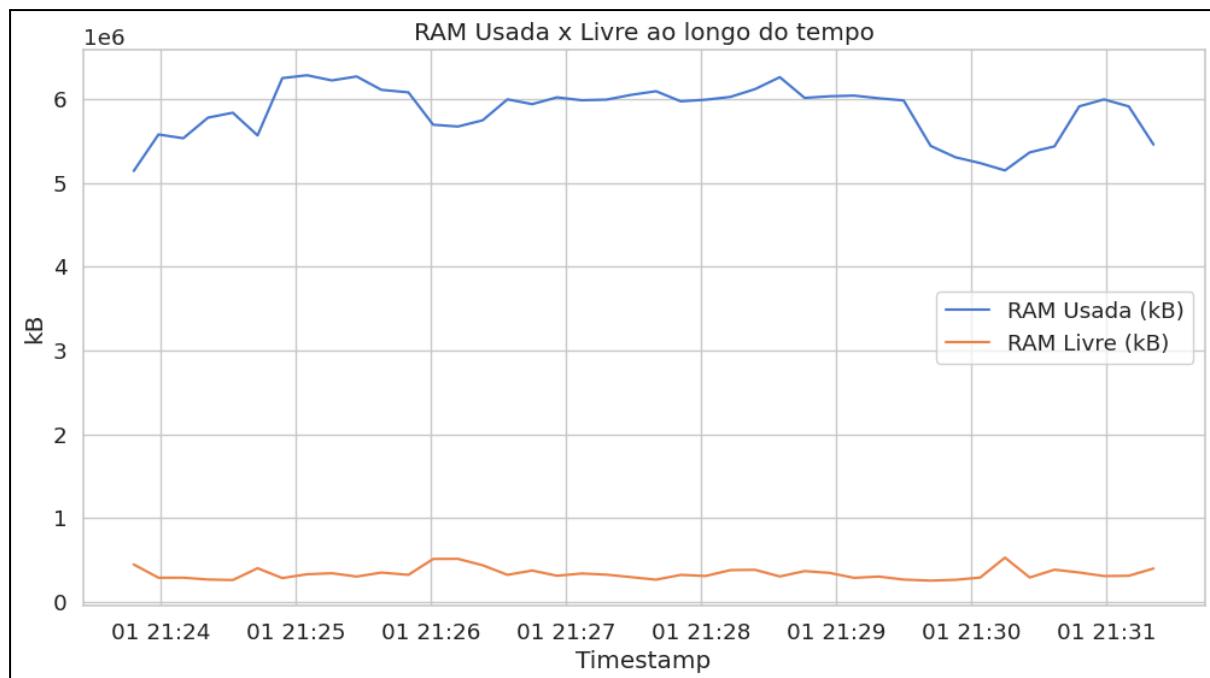
☞ PROFILING_EDA_PID3218_42C.ipynb (Notebook com gráficos e medidas). Abaixo os outputs apenas mais consideráveis das células:

	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

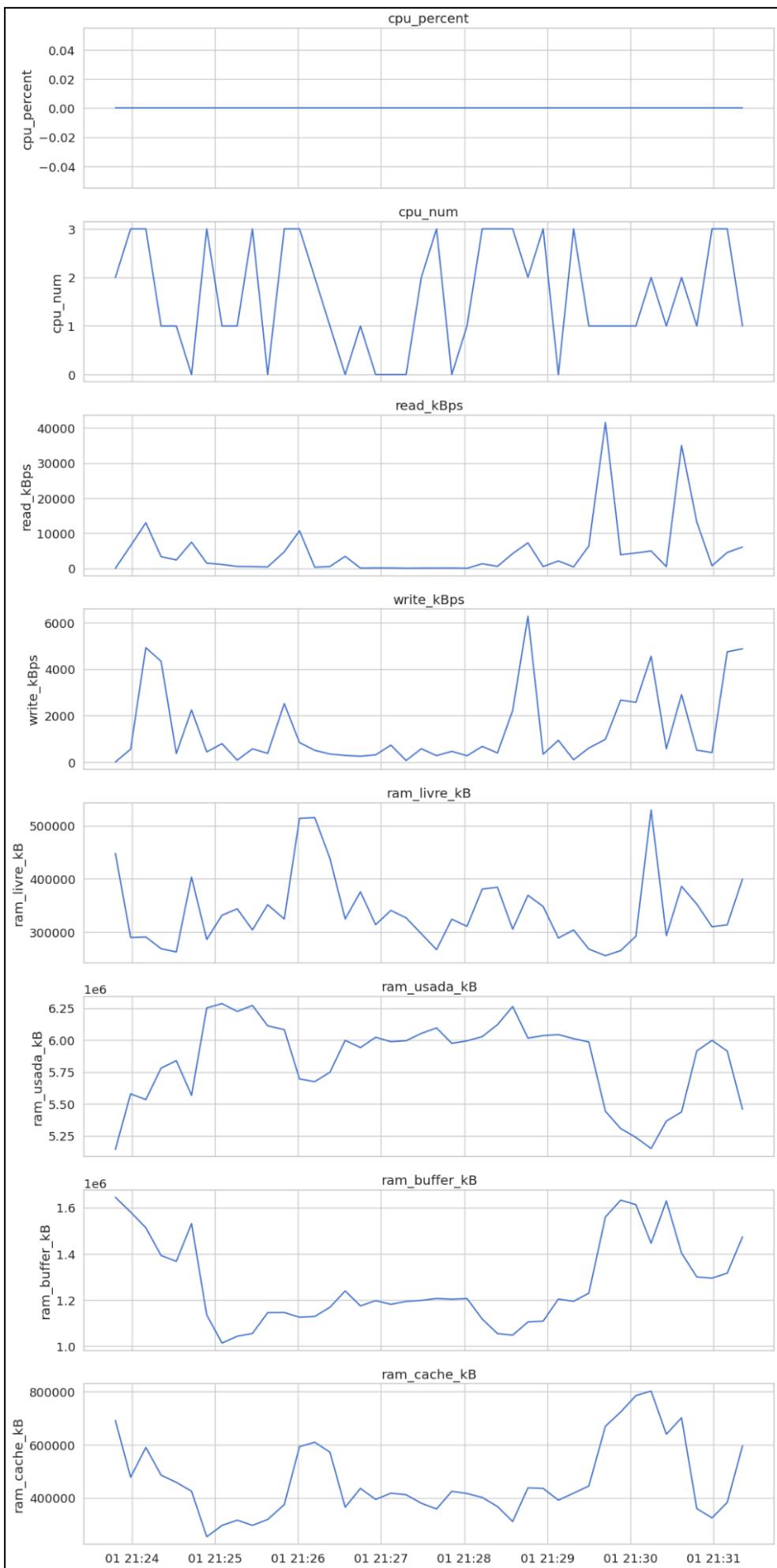
- A coluna PID é desprezível por tratar-se da tag do processo tratada. Observa-se que seu uso de CPU é desconsiderável, no processo tendo estado fixo em 0%, tal é um processo pai de outros módulos e normalmente possui valores próximos a 0 ou 0, não sendo interessante para o que foi solicitado analisar 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.