Conhecendo e identificando os gargalos de desempenho no Linux Embarcado

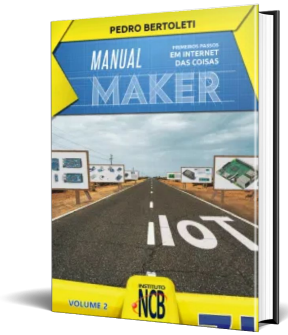
Autor:

Pedro Bertoleti

PALESTRANTE: PEDRO BERTOLETI

- Consultor Tecnológico - Instituto de Pesquisas ELDORADO
- Graduado em engenharia elétrica (UNESP)
- Entusiasta do movimento maker / DIY
- Áreas de interesse: sistemas embarcados (bare-metal, RTOS e Linux embarcado), conectividade e IoT

Autor de 3 livros:



INSTITUTO DE PESQUISAS ELDORADO

Há 24 anos, a ELDORADO desenvolve soluções de hardware e software customizadas para diversos setores do mercado, se posicionando como uma grande articuladora de inovação aberta, com projetos que colocam o Brasil no mapa mundial da tecnologia.

Presente em quatro cidades brasileiras (Campinas, Manaus, Porto Alegre e Brasília), o ELDORADO tem como destaques:

- Ser um provedor de P&D&I de classe mundial com histórico de 24 anos no desenvolvimento de soluções sob medida em software e hardware.
- Ser um dos protagonistas da Inovação aberta.
- Possuir conexão internacional: relacionamentos com RTOs mundiais e somos membros da EARTO (European Association of Research and Technology Organisations)
- Ter parcerias com hubs de inovação e atuamos com aceleração tecnológica de startups.

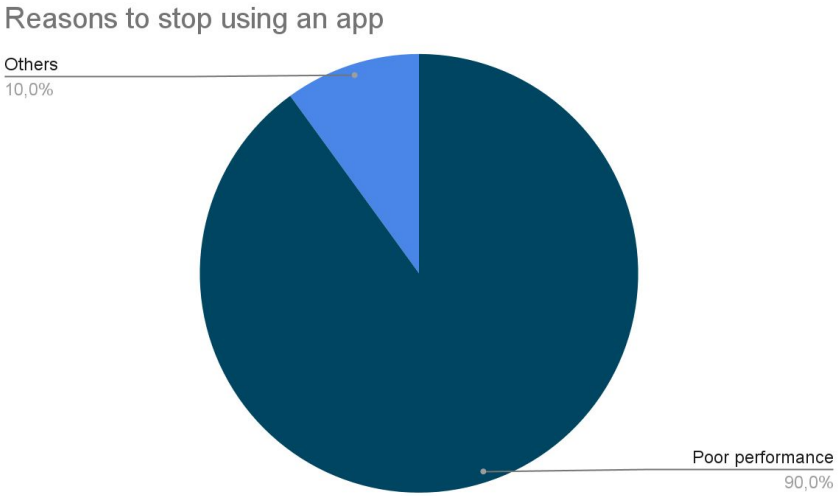
AGENDA

1. Introdução
2. Análise de uso de CPU
3. Análise de consumo de memória RAM
4. Iowait: o que é e como ele afeta o desempenho
5. Conclusão

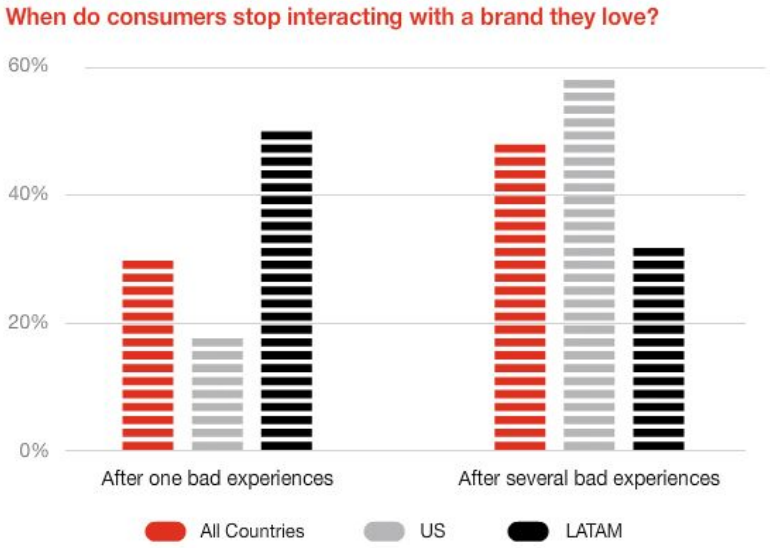
INTRODUÇÃO

Por que se preocupar com o desempenho de soluções em sistemas embarcados?

Efeitos do mau desempenho em sistemas computadorizados:

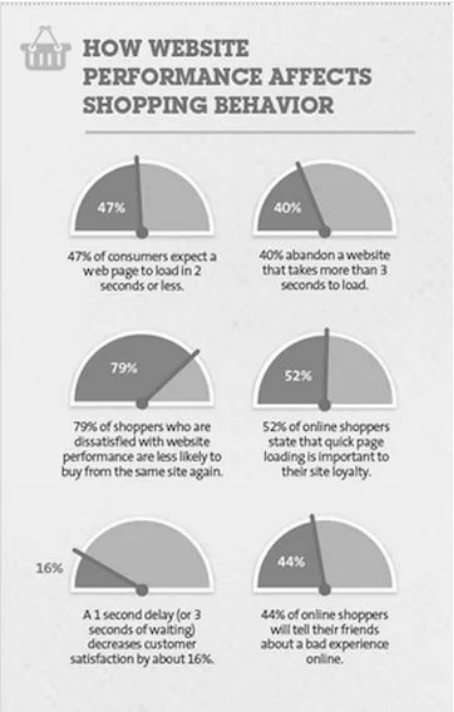


Fonte: [UX statistics (toptal)]



Fonte: [PwC]

Efeitos do mau desempenho em sistemas computadorizados:



Fonte: medium.com

- 48% of users say they feel frustrated and annoyed when they get to a site that's not mobile-friendly
- 36% said they felt like they've wasted their time by visiting those sites
- 52% of users said that a bad mobile experience made them less likely to engage with a company
- 48% said that if a site didn't work well on their smartphones, it made them feel like the company didn't care about their business

Fonte: [Think With Google]

Então, um mau desempenho significa:

Menor retenção de clientes e redução de receita

Reputação de empresas sendo afetadas

Perda de conversão de clientes

Obsolescência antecipada de sistemas

Problemas de desempenho afetam sistemas embarcados em:

1. Experiência de uso (UX)
2. Estabilidade e confiabilidade
3. Funcionamento de sensores e atuadores

Responsividade falha a cliques	Alto tempo de inicialização	Travamentos e congelamentos
Falhas intermitentes	Timeouts e deadlocks (Not Responding)	Corrompimento de dados
Leitura lenta de sensores	Ação tardia dos atuadores	Ações indesejadas e acidentes devido à lentidão

Os problemas de desempenho podem, portanto, ser abordados nos seguintes pontos de vista:

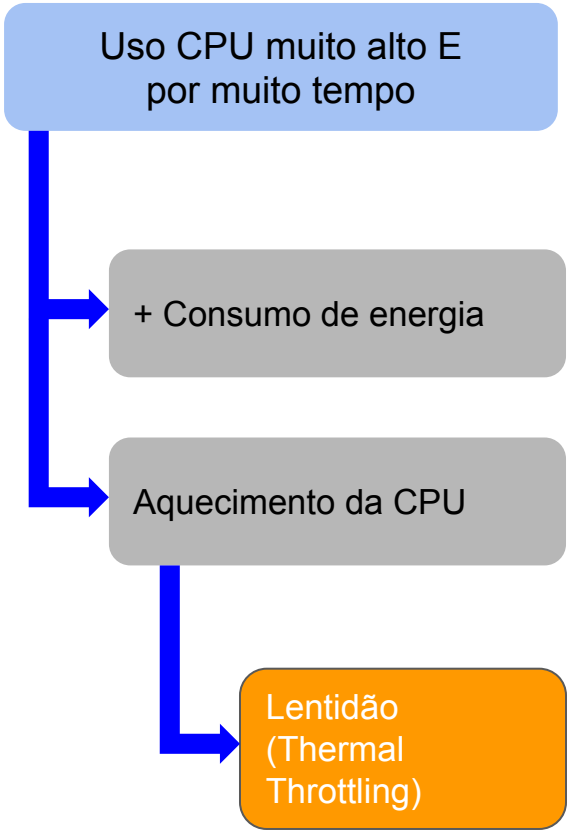
- Uso de CPU ao longo do tempo
- Uso e manipulação de memória RAM ao longo do tempo
- Uso de I/O ao longo do tempo

ANÁLISE DE USO DE CPU

PROBLEMA

O uso adequado de CPU em sistemas embarcados com Linux - e em qualquer outro sistema embarcado - é de suma importância para o bom desempenho geral do sistema em questão

Além disso, o uso adequado de CPu é fundamental para otimizar o consumo de energia elétrica da solução.



IDENTIFICAÇÃO DO PROBLEMA

Ao se observar lentidão e desempenho insatisfatório numa solução com Linux embarcado, uma das primeiras providências a se tomar é: verificar se o uso de CPU é alto ao longo de um período de tempo significativo.
A verificação deve ser feita em duas frentes, ambas durante um período de tempo expressivo:

- **Por processo específico**

- Processos que tendem a ter maior uso de CPU
- Processos previamente suspeitos ou com tendência a alto uso de CPU
- Processos inseridos recentemente (novos na solução)

- **Geral (somatória de todos os processos)**

- Monitoramento do uso de CPu geral ao longo de uma janela de tempo significativa para sua solução

IDENTIFICAÇÃO DO PROBLEMA

Ainda, ao ser observado uso excessivo de CPU, uma das duas hipóteses a seguir são válidas:

Hipótese 1: CPUs são subdimensionadas para a solução

Hipótese 2: processos e aplicações não suficientemente otimizados

IDENTIFICAÇÃO DO PROBLEMA

Ainda, ao ser observado uso excessivo de CPU, uma das duas hipóteses a seguir são válidas:

Hipótese 1: CPUs são subdimensionadas para a solução

Hipótese 2: processos e aplicações não suficientemente otimizados

IDENTIFICAÇÃO DO PROBLEMA

Ainda, ao ser observado uso excessivo de CPU, uma das duas hipóteses a seguir são válidas:

Hipótese 1: CPUs são subdimensionadas para a solução

Neste caso, infelizmente não há muito a se fazer além de se substituir o SoC ou SIP da solução para um que suporte-a.

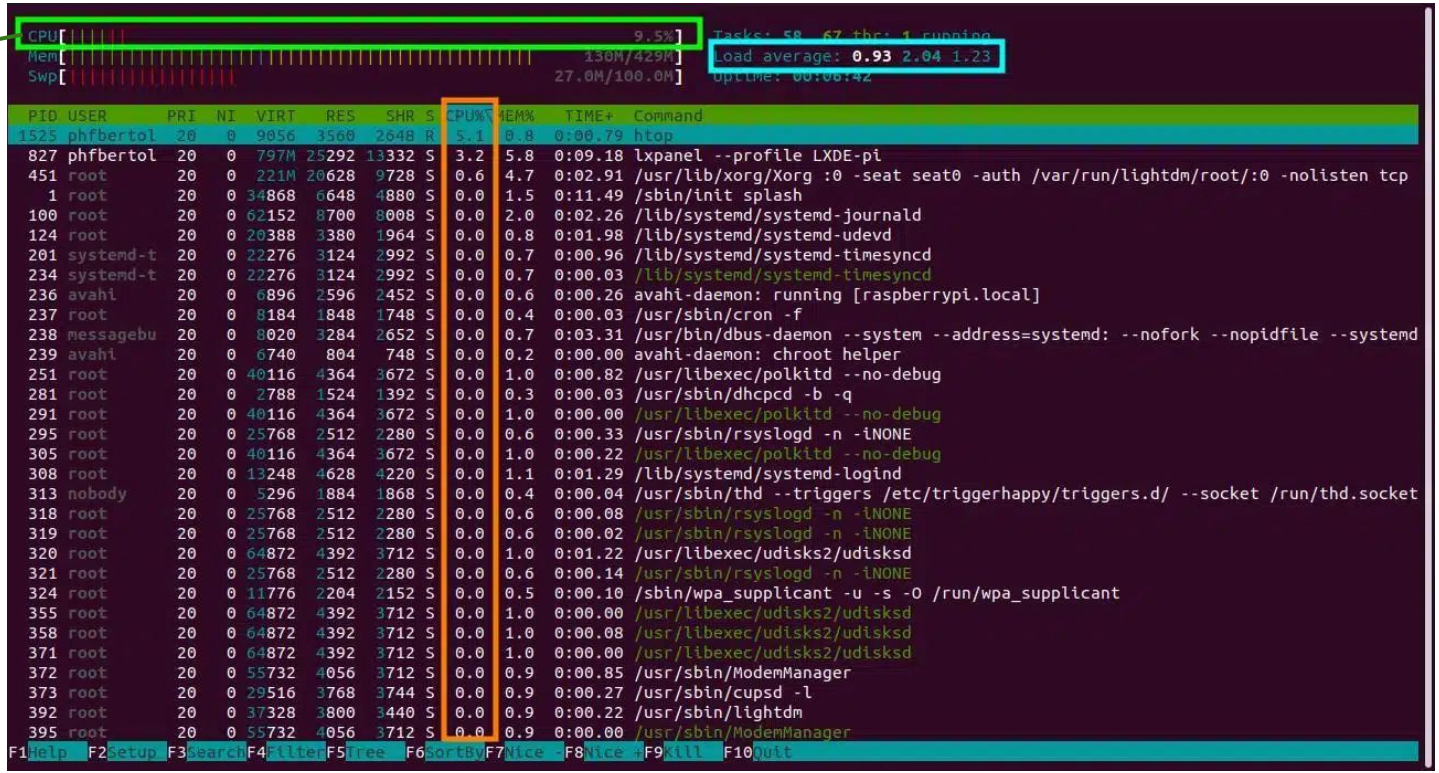
Hipótese 2: processos e aplicações não suficientemente otimizados

Melhorias na implementação de software das aplicações / processos da solução podem otimizar significativamente o uso de CPU.

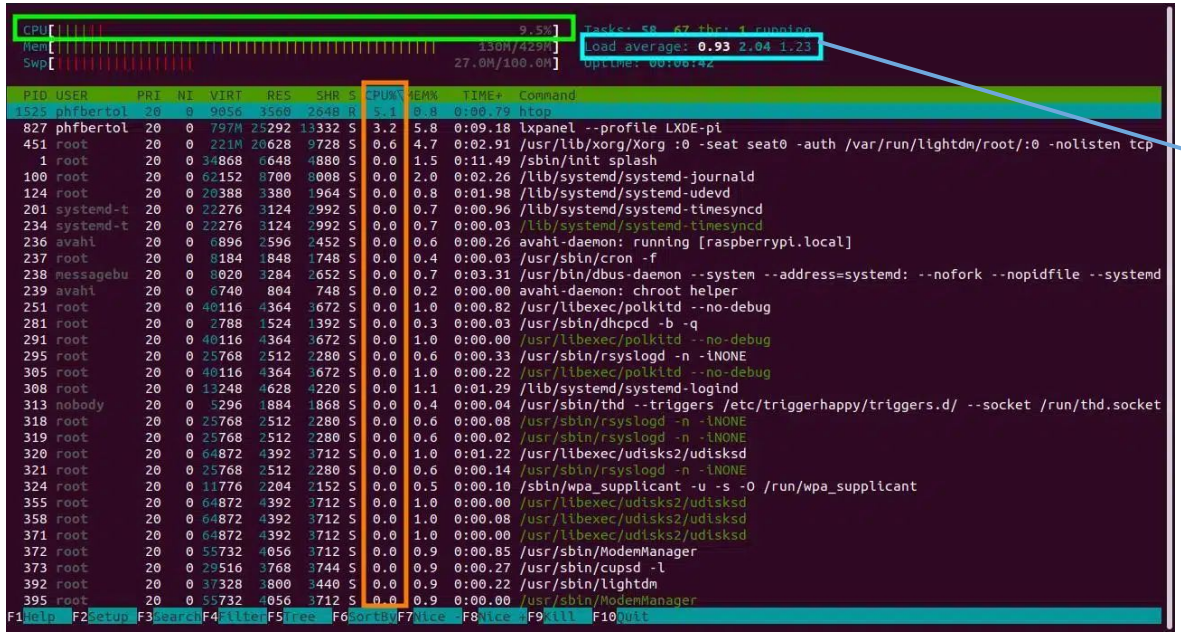
Logo, otimizar tais aplicações e processos tem como efeito a melhora geral de desempenho do Linux embarcado.

E como monitorar o uso de CPU?
Ferramenta: htop

% instantâneo de uso de CPU



E como monitorar o uso de CPU?
Ferramenta: htop



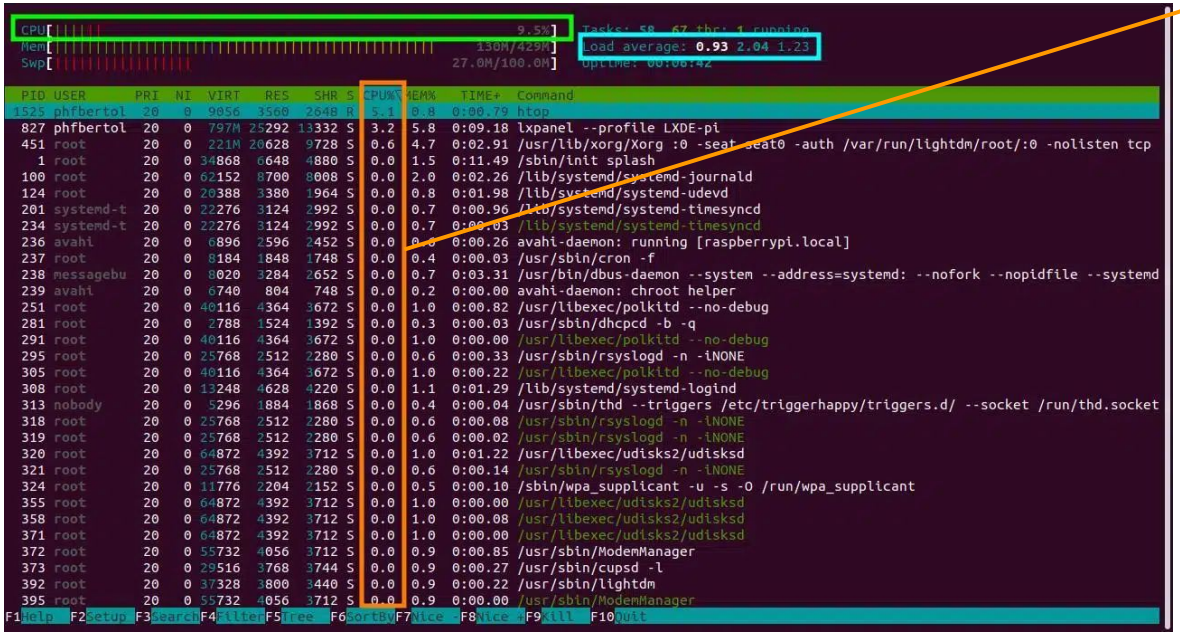
Load average:

- Uso no último minuto
- Uso nos últimos 5 minutos
- Uso nos últimos 15 minutos

Assumindo o número de CPUs como N, se o load average for:

- Inferior a N: as CPUs estão superdimensionadas
- Igual a N: quantidade de CPUs corretamente dimensionada
- Superior a N: as CPUs estão subdimensionadas

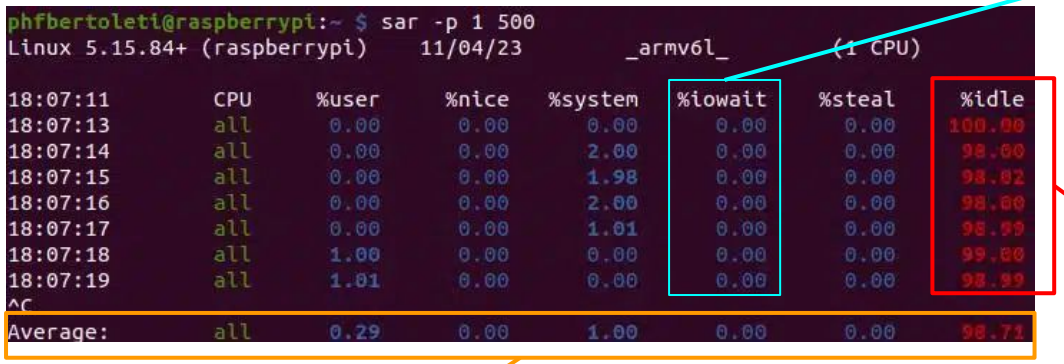
E como monitorar o uso de CPU?
Ferramenta: htop



CPU % por processo

Muito útil para identificar processos que, isoladamente, demandam muito processamento das CPUs disponíveis

E como monitorar o uso de CPU?
Ferramenta: sar (System Activity Reporter)



iowait: tempo de espera por alguma resposta de I/O (disco, por exemplo)

Tempo de ociosidade de CPU

% de tempo de uso de CPU para cada tipo de atividade

Estratégias para investigação de uso de CPU no Linux embarcado

- **Passo 1:** Pelo Load Average (nos últimos 1, 5 e 15 minutos), verificar se as CPUs estão subdimensionadas. Se sim, seguir ao passo 2.

- **Passo 2:** Verificar, via % de uso de CPU individual, se algum processo ou aplicação está utilizando mais CPU que o esperado (ou usando algo muito alto constantemente).

Caso não existir, todo o conjunto de softwares utilizados na solução pode ser além do suportado pelo hardware. Nesse caso, para minimizar os problemas de desempenho, verifique se há otimizações possíveis de serem feitas em termos de iowait.

Caso existir, partir para passo 3.

- **Passo 3:** se o processo ou aplicação que está utilizando muito processamento é de sua autoria, verifique possíveis otimizações. Caso for da autoria de terceiros, procure por atualizações ou até mesmo abra um Bug Report para o responsável (pois seu problema pode ser comum a muitas outras pessoas).

ANÁLISE DE USO DE MEMÓRIA RAM

O bom uso de memória RAM é vital para um bom desempenho de um sistema embarcado.

É na memória RAM que são armazenados, em tempo de execução:

- Processos em execução
- Aplicações em execução
- Processos e aplicações do sistema operacional (Linux embarcado)
- Arquivos frequentemente acessados (pinned files)
- Variáveis
- …e demais dados necessários para o sistema operacional e aplicações funcionarem conforme o esperado

Em alguns casos de uso, a quantidade de memória RAM demandada pode ser:

- Próximo da quantidade máxima disponível
- **Acima da quantidade máxima disponível**

Quando é demandada uma quantidade maior de memória RAM do que a disponível, o sistema operacional pode encerrar processos pouco prioritários para liberação de RAM ou, de forma mais comum, realizar a operação de SWAP.

No SWAP, o sistema operacional envia, para um espaço dedicado no disco (chamado **swap memory**), as páginas de memória menos utilizadas, liberando espaço na memória RAM para páginas com dados mais utilizados.

Da mesma forma, se forem necessários os dados armazenados na swap memory, estes são lidos do disco.

Embora resolva o problema de memória RAM insuficiente, o swap tem como consequência a **lentidão**.

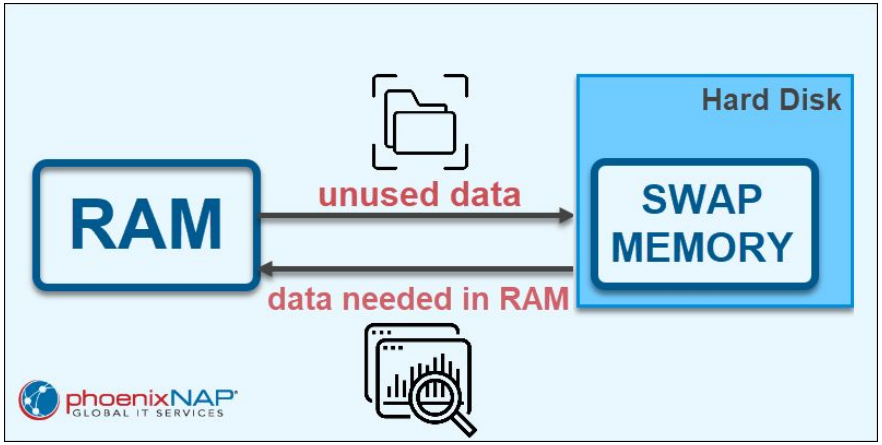
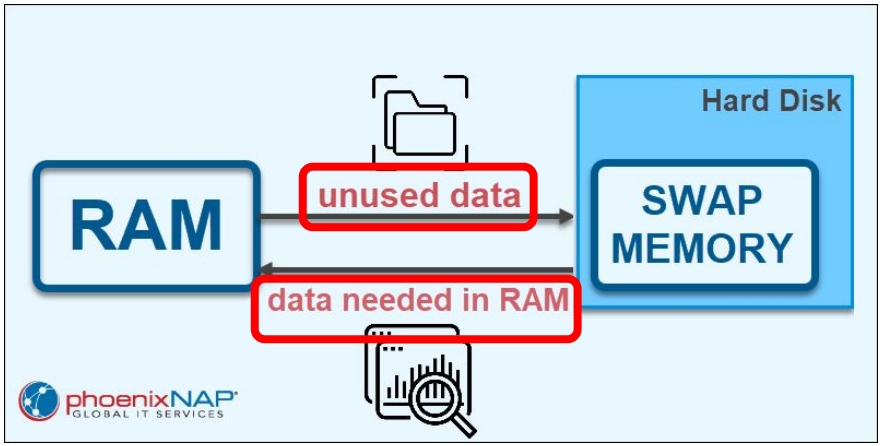


Imagem obtida de: https://phoenixnap.com/kb/swap-memory

Swap in
acesso a disco é lento, logo swap in é lento

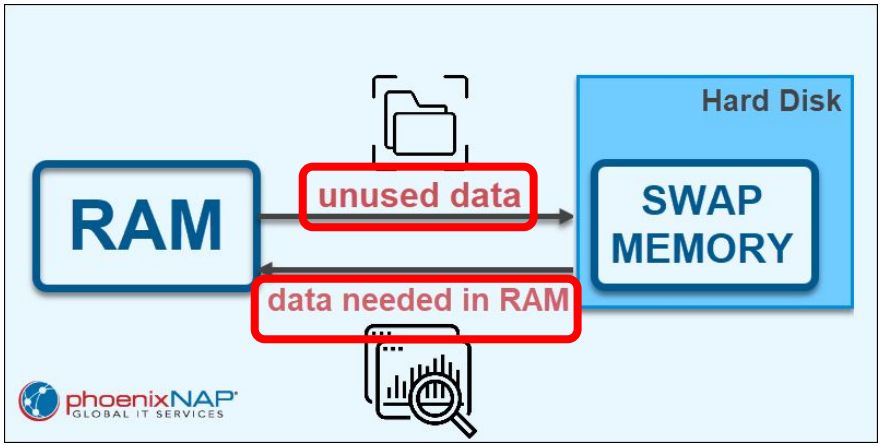


Swap out
acesso a disco é lento, logo swap in é lento

Cada dado que faz parte da operação de swap implica, obrigatoriamente, em **atraso para escrita em disco e um atraso para leitura em disco**.

Isso aumenta o iowait, tempo que os processos e aplicações esperam por uma resposta de I/O, causando lentidão.

Swap in
acesso a disco é lento, logo swap in é lento



Swap out
acesso a disco é lento, logo swap in é lento

Quando a quantidade de RAM disponível é muito pequena para as necessidades da solução, muito swap é feito. Logo, muitos atrasos (em virtude das leituras e escritas em disco) ocorre.

Em casos extremos, o sistema operacional gasta mais tempo fazendo escrita e leitura da swap memory do que usando os dados lidos. Nesse caso, ocorre o **memory thrashing**.

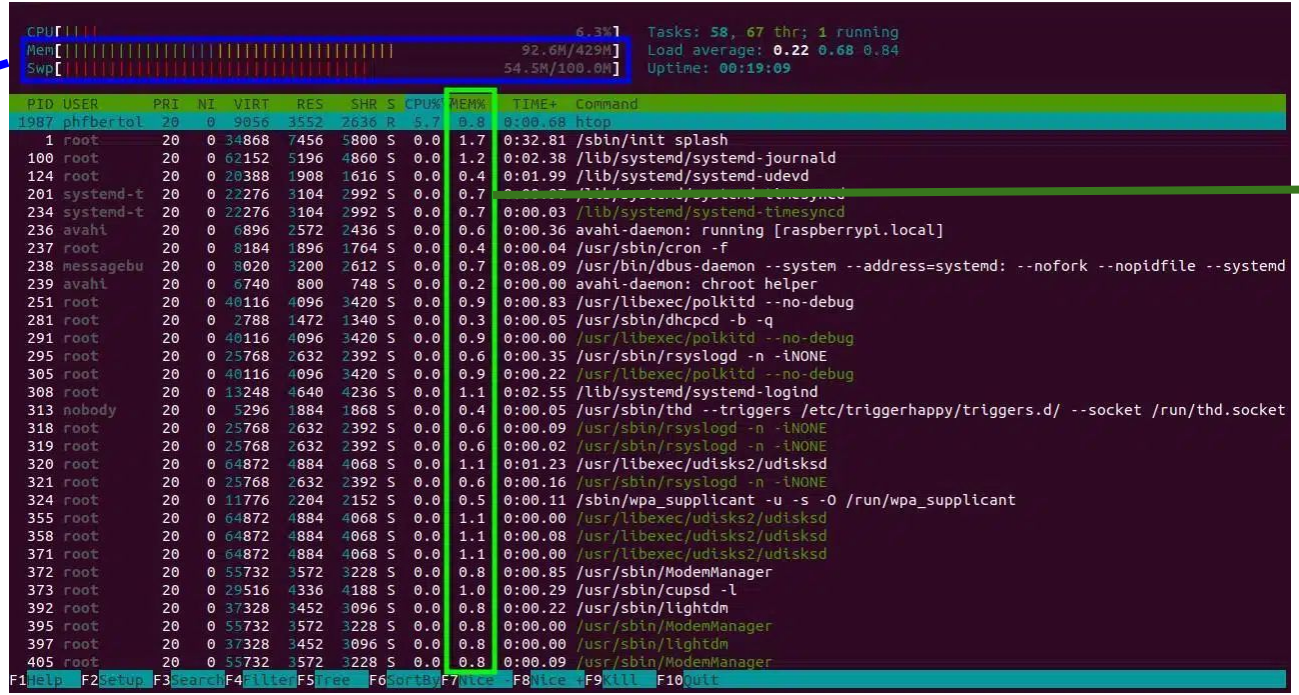
O **memory thrashing** é, como visto, uma consequência do uso exagerado de memory swap.

Em uma analogia livre, o **memory thrashing é como cavar um poço profundo usando uma colher**.

A colher (memória RAM) comporta pouco material (dados, buffers, aplicações, etc.), logo quem está cavando o buraco (sistema operacional) deve fazer várias viagens até um reservatório (memória de massa) para esvaziar a colher nele (memory swap) e continuar cavando. Nesse caso, dada a baixa capacidade da colher, o processo de esvaziá-la leva um tempo significativo em relação à operação principal, que é cavar. Isso faz com que o sistema operacional gaste muito tempo esvaziando a colher, causando o memory thrashing).

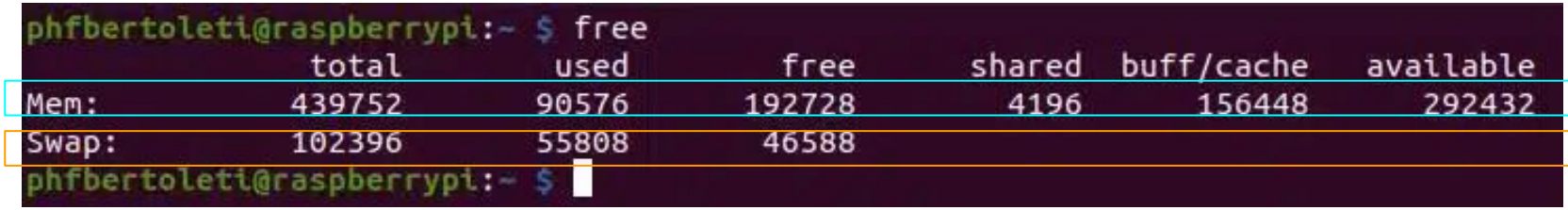
Como diagnosticar problemas de performance relacionados a consumo de memória RAM?
Ferramenta: htop

Uso % de RAM e Swap



Uso % de RAM por processo

Como diagnosticar problemas de performance relacionados a consumo de memória RAM?
Ferramenta: free



Estratégias para evitar o swap e suas consequências:

- **Utilize ZRAM:** o ZRAM é, em poucas palavras, um RAM drive (partição montada em RAM), em que tudo é gravado de forma compactada e automaticamente descompactado ao ser lido. Dessa forma, como o tempo de acesso em RAM é muito menor do que tempo de acesso em disco, é possível se usar o ZRAM como memory swap e minimizar a lentidão nas operações de swap.

- **Configuração do swappiness:** parametrize corretamente o swappiness em /proc/sys/vm/swappiness . Dessa forma, é possível configurar o quão agressivamente o Linux fará o swap, o que ajuda a minimizar a quantidade de operações de swap em uma dada janela de tempo.

- **Identifique vazamentos de memória:** memory leaks causam grandes consumos de memória RAM e, logo, muito swap e memory thrashing. Se um processo ou aplicação é de sua autoria e está com consumo crescente de RAM ao longo do tempo, utilize ferramentas como o valgrind para detectar onde há memory leaks e, assim, permitir a correção dos mesmos.

IOWAIT

Em qualquer sistema computadorizado, é sabido que periféricos mais lentos (discos, por exemplo) são MUITAS VEZES mais lentos do que processadores.
Tomando como exemplo um disco (seja HD ou memória Flash), o tempo que este requer para escrever um dado ou ler um dado é **MUITO** superior ao tempo que o processador leva para processar este mesmo dado.

Quando é necessária a escrita ou leitura de um dado em um periférico mais lento, a consequência é um grande tempo de ociosidade do processador, o qual tem que esperar tal escrita ou leitura para prosseguir com suas tarefas.

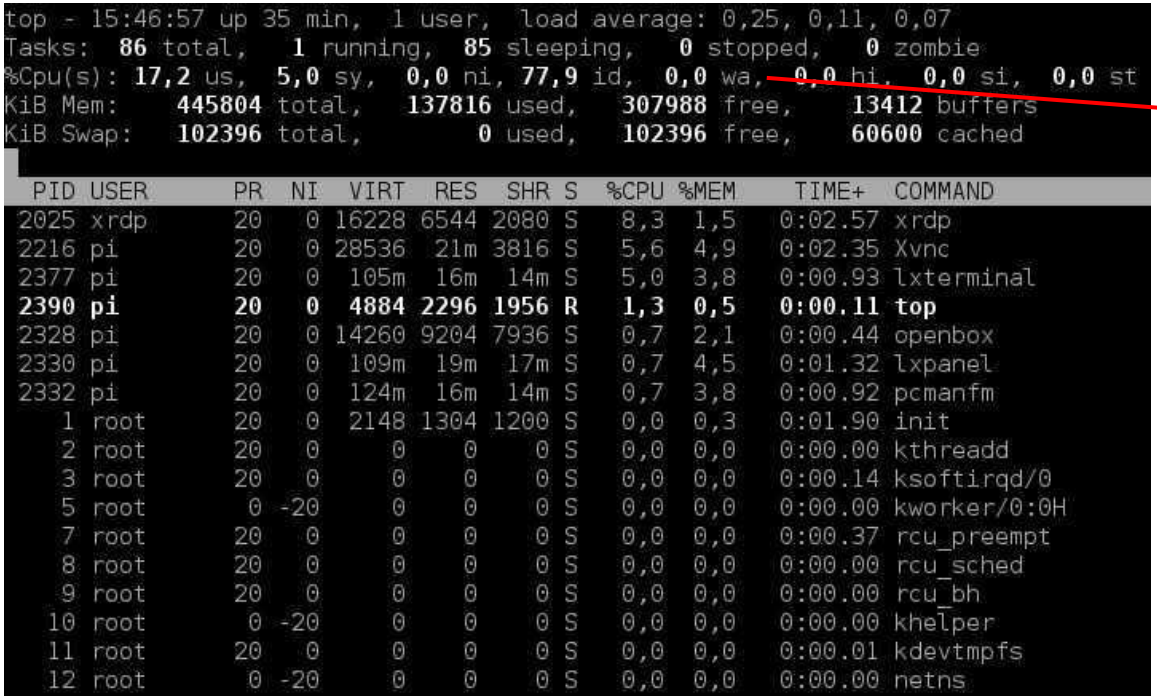
No Linux, este tempo de ociosidade de uma ou mais CPUs enquanto espera-se por uma resposta de um periférico lento é chamado de **iowait**, e é representado por um valor percentual, **significando a porcentagem de tempo de uso que o processador gastou esperando pela resposta de um periférico.**

Por isso percebemos menos lentidão geral de um sistema Linux quando:

- Os dados a serem manipulados já estão carregados em memória RAM
- É utilizado uma memória de armazenamento em massa que apresenta maior velocidade de escrita e leitura (NVME, por exemplo)

Dessa forma, **quanto mais lentos forem os periféricos, maior iowait é esperado.**

Como é feita a análise de iowait?
Ferramenta: top



O valor wa corresponde a porcentagem de iowait

Estratégias para diminuição do iowait:

1. Sempre que possível, **substitua memórias de armazenamento lentas** (HD, por exemplo) por outras mais rápidas (NVME, por exemplo)

2. Procure utilizar o flag de **discard** no fstab.

O **discard** é usado para habilitar o recurso de TRIM em dispositivos de armazenamento (memória Flash). O TRIM é uma operação que permite ao sistema operacional ter conhecimento de quais blocos de dados do sistema de armazenamento não estão mais em uso e, logo, podem ser apagados previamente, melhorando a eficiência da escrita de novas páginas futuras na memória Flash.

Dúvidas?

Contato:

E-mail: contato@pedrobertoleti.com.br
LinkedIn: https://www.linkedin.com/in/pedro-bertoleti-28703a25/
Instagram: @pedro_bertoleti