



ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

PROJETO INTEGRADOR I-A

ANÁLISE DE DESEMPENHO DE CACHE

E

VIRTUALIZAÇÃO DE SISTEMAS OPERACIONAIS

BIATRICE DE LEON CANTALUPPI

LEONARDO ANGELO LOPES DA SILVA

PAULO CÉSAR RODRIGUES MASSON

VÍCTOR LIMA VIEIRA

Pelotas, 2025

SUMÁRIO

Introdução.....	2
Apresentação do Projeto.....	2
Objetivo do Relatório.....	2
Etapas Principais.....	2
Hierarquia de Memória.....	3
Registradores.....	4
Cache.....	4
Memória principal.....	4
Memória Secundária.....	4
Memória Cache.....	5
Características.....	5
Conceitos.....	5
Principais configurações.....	5
Tipos de mapeamento.....	6
Gerenciamento de cache.....	6
Virtualização.....	7
Definição.....	7
Conceitos.....	7
Características.....	7
Resultados preliminares.....	7

1. APRESENTAÇÃO DO PROJETO

Este trabalho visa analisar o desempenho de cache em um sistema operacional virtualizado, identificando as principais oportunidades para melhorar o desempenho e reduzir os custos. Além disso, este trabalho também visa contribuir para a compreensão dos efeitos da virtualização na análise de desempenho de cache.

1.1 OBJETIVO DO RELATÓRIO

- Avaliar o desempenho de sistemas computacionais com diferentes configurações de cache;
- Compreender o impacto da memória cache no desempenho de programas;
- Abordar conceitos de sistemas operacionais com o SimpleScalar.

1.2 ETAPAS PRINCIPAIS

1. Análise sobre virtualização e memória cache.
2. Principais configurações de cache.
3. Testes.
4. Resultados preliminares.

2. DESENVOLVIMENTO

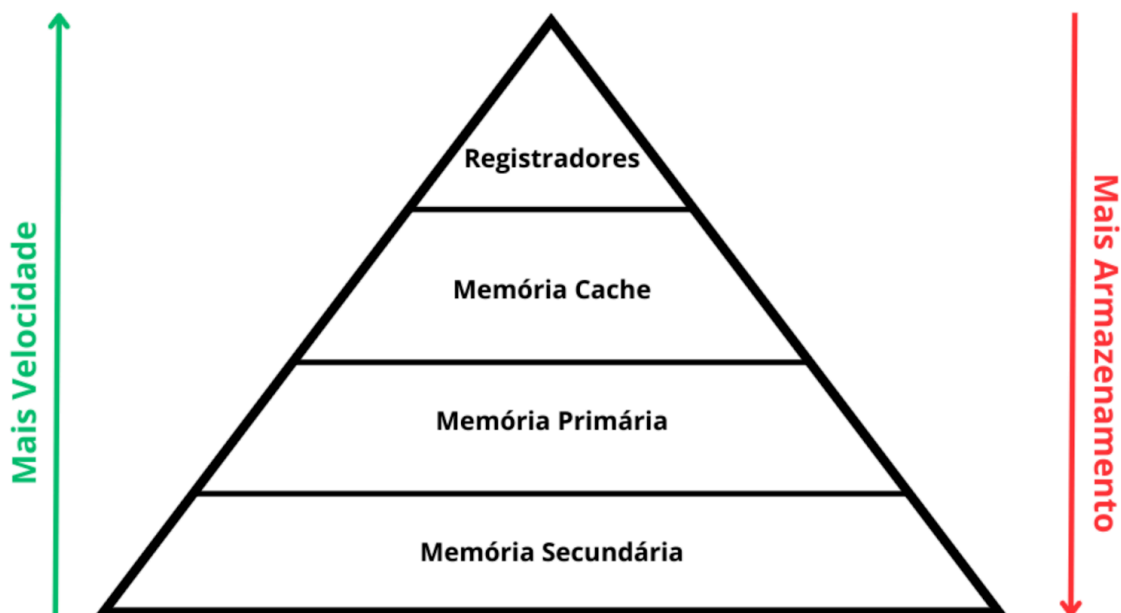
2.1 HIERARQUIA DE MEMÓRIA

A hierarquia de memória é um conceito que se refere à organização da memória em diferentes níveis, cada um com suas próprias características e funções.

É composta por:

- **Registradores:** São os componentes de memória mais rápidos e mais caros. Eles são usados para armazenar dados temporários durante a execução de instruções.
- **Cache:** É uma memória rápida e cara que armazena cópia de dados frequentemente acessados. Ela é usada para reduzir o tempo de acesso à memória principal.
- **Memória Principal:** É a memória que armazena os dados e programas que estão sendo executados pelo processador.
- **Memória Secundária:** É a memória que armazena dados e programas que não estão sendo executados pelo processador, mas que podem ser necessários em algum momento.

Figura 1: Hierarquia de Memória



Fonte: <https://blog.formacao.dev/o-que-e-hierarquia-de-memoria/>

2.2 MEMÓRIA CACHE

A memória cache é uma pequena quantidade de memória rápida que armazena dados frequentemente acessados por um sistema computacional. Ela é usada para melhorar o desempenho do sistema, reduzindo o tempo de acesso à memória principal.

Características:

- **Rápida:** Mais rápida do que a memória principal.
- **Pequena:** Menor do que a memória principal.
- **Volátil:** Os dados são perdidos quando o sistema é desligado.

Conceitos de Cache:

- **Associativa:** É um tipo de cache que armazena os dados em uma tabela de hashing, permitindo que os dados sejam acessados rapidamente.
- **Mapeamento:** É o processo de mapear os endereços de memória para os endereços de cache.
- **Substituição:** É o processo de substituir os dados armazenados na cache por novos dados.
- **Validação:** É o processo de verificar se os dados armazenados na cache são válidos.

As principais configurações de cache incluem:

- **Tamanho da Cache:** O tamanho da cache é a quantidade de memória disponível para armazenar dados. Um tamanho de cache maior pode melhorar o desempenho, mas também pode aumentar o custo e o consumo de energia.
- **Tamanho do Bloco:** O tamanho do bloco é a quantidade de dados que são armazenados em uma única linha de cache. Um tamanho de bloco maior pode melhorar o desempenho, mas também pode aumentar a quantidade de dados que precisam ser transferidos.
- **Tipo de Mapeamento:** O tipo de mapeamento é a forma como os dados são armazenados na cache.

Os principais tipos de mapeamento são:

- **Mapeamento Direto:** Os dados são armazenados em uma linha de cache específica, com base no endereço de memória.
- **Mapeamento Associativo:** Os dados são armazenados em uma linha de cache que é associada ao endereço de memória.

Mapeamento por Conjunto: Os dados são armazenados em um conjunto de linhas de cache.

Cache de Nível 1 (L1): É a cache mais próxima do processador, armazenando os dados mais frequentemente acessados.

Cache de Nível 2 (L2): É a cache que fica entre a cache L1 e a memória principal, armazenando os dados que não estão na cache L1.

Cache de Nível 3 (L3): É a cache que fica entre a cache L2 e a memória principal, armazenando os dados que não estão na cache L2.

Já as políticas de gerenciamento de cache são usadas para decidir quais dados devem ser armazenados na cache e quais devem ser removidos.

Algumas das políticas mais comuns incluem:

FIFO (First-In-First-Out): Os dados mais antigos são substituídos primeiro.

LRU (Least Recently Used): Os dados que não foram acessados recentemente são substituídos primeiro.

LFU (Least Frequently Used): Os dados que são menos frequentemente acessados são substituídos primeiro.

Essas políticas são importantes para garantir que a cache esteja sempre armazenando os dados mais importantes e frequentemente acessados, o que pode melhorar significativamente o desempenho do sistema.

2.3 VIRTUALIZAÇÃO

A virtualização é uma tecnologia que permite criar ambientes virtuais isolados, chamados de máquinas virtuais, que podem executar sistemas operacionais e aplicativos independentemente do sistema operacional hospedeiro.

A virtualização de Linux para Windows no VirtualBox é uma técnica que permite executar o sistema operacional Linux dentro de uma máquina virtual criada no sistema operacional Windows, utilizando o software VirtualBox.

O VirtualBox é um software de virtualização de código aberto que permite criar e executar máquinas virtuais em uma variedade de sistemas operacionais, incluindo Windows, Linux e macOS.

2.3.1 VANTAGENS DA VIRTUALIZAÇÃO DE LINUX PARA WINDOWS NO VIRTUALBOX:

Flexibilidade: permite executar o sistema operacional Linux em uma máquina virtual, sem a necessidade de reinstalar o sistema operacional ou criar uma partição separada.

Isolamento: a máquina virtual é isolada do sistema operacional hospedeiro, o que ajuda a prevenir a propagação de malware e a manter a segurança do sistema.

Portabilidade: a máquina virtual pode ser facilmente transferida para outro computador ou ambiente, sem a necessidade de reinstalar o sistema operacional ou aplicativos.

3. RESULTADO PRELIMINAR

Primeiro, baixamos o VirtualBox (Máquina Virtual), após isso baixamos o Mint porque era uma forma de virtualizar gastando menos memória, mas baixamos a versão errada de primeira, sendo a versão server (servidor) e tivemos que lidar apenas com comandos, o que tinha sido uma grande dificuldade. Ao trocar para a versão desktop do Mint, conseguimos rodar, porém achamos o sistema um pouco lerdo e por mais que o terminal estava rodando, a conclusão da instalação da máquina não foi concluída, pois não foi possível baixar aplicativos como Google e etc, era apenas possível mexer no terminal e coisas básicas. Nesse momento decidimos baixar o Ubuntu, colocamos 20gb, 2 CPUs, configuramos alguns detalhes a mais, e conseguimos iniciar o sistema, abrindo o terminal e rodando o sistema Linux de forma eficiente. Essa foi a experiência até então de Victor e Biatrice juntos, rodarem a máquina virtual. Leonardo através de seu computador optou por usar o docker e com sucesso conseguiu simulações da entrega parcial do Projeto.

Resumo da Execução da Simulação

Fizemos a virtualização de um sistema operacional Ubuntu na versão 24.04.1 LTS utilizando a tecnologia Docker ¹. O ambiente virtualizado foi configurado possuindo 16gb de memória RAM, dentro do ambiente, seguiu-se o guia² disponibilizado pela Prof^a. Dr^a Morgana M. A. da Rosa no Ambiente Virtual da universidade.

Já tendo o Docker instalado no subsistema WSL no Windows 11, criou-se o arquivo Dockerfile para gerar um ambiente Ubuntu virtualizado.



```
1 FROM ubuntu:latest
2
3 LABEL maintainer="1leonardo.lopes@gmail.com"
4
5 ENV DEBIAN_FRONTEND=noninteractive
6
7 RUN apt update && apt install -y \
8     curl \
9     nano \
10    vim \
11    build-essential \
12    && rm -rf /var/lib/apt/lists/*
13
14 CMD ["/bin/bash"]
15
```

Arquivo Dockerfile

Executou-se o comando

```
docker build -t ubuntu-container .
docker run -it ubuntu-container
```

Utilizou-se o comando docker cp para copiar os arquivos do SimpleScalar da máquina Windows para o container

Após, seguiu-se o guia da professora Morgana.

Foi utilizado o SimpleScalar com a ferramenta sim-cache para simular o comportamento do cache ao rodar o compilador gcc. A configuração incluiu caches L1 de instruções e dados, ambos com 64 conjuntos, blocos de 64 bytes, mapeamento direto e política de substituição LRU. Não foram utilizados caches L2 nem TLBs (Translation Lookaside Buffers).

Resultados Obtidos

1. Desempenho Geral

- Instruções executadas: 45.180.825
- Acessos à memória: 16.749.302
- Tempo total: 1 segundo
- Velocidade da simulação: 45.180.825 instruções por segundo

2. Cache de Instruções L1

- Total de acessos: 45.180.825
- Hits: 40.941.619 (~90,6%)
- Misses: 4.239.206 (~9,4%)
- Replacements: 4.239.142

3. Cache de Dados L1

- Total de acessos: 16.879.815
- Hits: 15.570.920 (~92,3%)
- Misses: 1.308.895 (~7,7%)
- Writebacks: 463.194

4. Uso de Memória e Paginação

- Total de páginas alocadas: 691
- Memória alocada: 2.764 KB
- Acessos à Tabela de Páginas: 227.874.906
- Page misses: 701 (taxa de ~0,0003%)

O cache de instruções teve um miss rate de 9,4%, já o cache de dados teve um desempenho um pouco melhor, com um miss rate de 7,7%.

A memória virtual teve um desempenho excelente, com quase nenhum page miss, o que indica que quase todas as páginas necessárias estavam presentes na RAM.

Referências

1. **ubuntu - Official Image | Docker Hub.** Docker Hub, 2025. Disponível em: https://hub.docker.com/_/ubuntu. Acesso em: 29 mar. 2025.
2. **Guia Prático para a Execução do Trabalho Individual: Análise de Desempenho de Cache e Virtualização de Sistemas Operacionais.** Plataforma A. Disponível em <https://ucpel.grupoa.education/plataforma/course/2309764/content/34852561>. Acesso em: 29 mar. 2025
3. **Hennessy, J. L., Patterson, D. A. (2017). Arquitetura de computadores: Uma abordagem quantitativa.** Elsevier.
4. **Stallings, W. (2018). Arquitetura e organização de computadores.** Cengage Learning.
5. **Silberschatz, A., Galvin, P. B., & Gagne, G. (2019). Sistemas operacionais.** Elsevier.
6. <https://blog.formacao.dev/o-que-e-hierarquia-de-memoria/>
7. https://wiki.betim.ifmg.edu.br/docs/organizacao_computadores/hierarquia_memorias

