Aluno(a): Guilherme Ariza Gama Silva
Orientador(a): Anaximandro Anderson Pereira Melo De Souza

Curso: MBA em Engenharia de Software

Título do Projeto de Pesquisa

Análise comparativa de desempenho entre B-Trees e LSM-Trees em Hardware Moderno (SSDs)

Introdução

O armazenamento e a recuperação eficiente de dados são pilares fundamentais dos sistemas de gerenciamento de banco de dados (SGBDs). Historicamente, as estruturas de indexação, como as B-Trees, foram projetadas tendo como base o desempenho de discos rígidos (HDDs), onde o tempo de busca mecânico era o principal gargalo. Contudo, a transição para dispositivos de estado sólido (SSDs) mudou profundamente o paradigma de otimização de I/O, trazendo novas implicações sobre o desempenho dessas estruturas.

A B-Tree, proposta por Bayer e McCreight (1972), consolidou-se como padrão para indexação em disco, permitindo atualizações in-place e eficiência em leituras sequenciais. Entretanto, as características dos SSDs, como a ausência de partes móveis, latências reduzidas e restrições de escrita, reduziram a efetividade desse modelo. Já a LSM-Tree, introduzida por O'Neil et al. (1996), adota um modelo baseado em gravações sequenciais e compactações periódicas, tornando-se amplamente utilizada em sistemas modernos como RocksDB e Bigtable.

A escolha entre B-Trees e LSM-Trees tem impactos diretos na performance, custo operacional e vida útil do hardware de armazenamento. Com a ampla adoção de SSDs NVMe PCIe 4.0, que oferecem larguras de banda da ordem de 3-7 GB/s e latências significativamente reduzidas, torna-se essencial reavaliar as premissas estabelecidas em estudos anteriores focados em SSDs SATA ou interfaces mais antigas.

A relevância desta pesquisa se justifica por três aspectos principais: (1) Grande parte da literatura comparativa foi produzida quando SSDs NVMe ainda eram emergentes, exigindo uma reavaliação empírica dos trade-offs conhecidos; (2) A amplificação de escrita (WAF) influencia diretamente a longevidade dos SSDs, e uma compreensão precisa pode resultar em economias significativas em ambientes de produção; (3) Sistemas como Apache Cassandra, ScyllaDB e RocksDB utilizam LSM-Trees, enquanto PostgreSQL, MySQL e

Oracle dependem de B-Trees, tornando dados atualizados essenciais para decisões arquiteturais.

Apesar da ampla adoção de ambas as estruturas, existe uma lacuna na literatura sobre como elas se comportam especificamente em SSDs NVMe de última geração. As características únicas desses dispositivos como alto paralelismo interno, múltiplas filas de comando (multi-queue), e latências extremamente baixas - podem alterar significativamente os trade-offs tradicionais.

Desta forma, a questão central que orienta esta pesquisa é: como o desempenho relativo entre B-Trees e LSM-Trees é afetado pelas características específicas de SSDs NVMe modernos (Gen4), considerando diferentes padrões de carga de trabalho? Levanta-se como hipóteses que: (1) LSM-Trees continuarão apresentando throughput superior em cargas write-intensive, porém com margens menores devido às melhorias de latência do NVMe; (2) B-Trees demonstrarão vantagens mais pronunciadas em leituras pontuais, beneficiando-se da baixa latência do NVMe; (3) A amplificação de escrita das B-Trees será mais crítica em SSDs NVMe devido ao maior volume de operações possíveis por segundo.

Objetivo

O objetivo desta pesquisa é realizar uma análise comparativa de desempenho entre as estruturas B-Tree e LSM-Tree em ambientes de armazenamento com SSDs NVMe PCIe 4.0. Busca-se mensurar e comparar as métricas de throughput, latência e amplificação de escrita (WAF), identificando como as particularidades do hardware influenciam o desempenho de cada estrutura. Especificamente, pretende-se: (a) quantificar o WAF de cada estrutura sob diferentes cargas de trabalho; (b) comparar latências de leituras pontuais e range scans; (c) avaliar o throughput em cenários write-intensive, read-intensive e mistos.

Metodologia

A pesquisa será conduzida por meio de experimentação controlada, seguindo uma abordagem quantitativa comparativa. Serão implementadas duas configurações de sistema: uma baseada em B-Tree e outra em LSM-Tree, utilizando, respectivamente, um SGBD tradicional (PostgreSQL ou MySQL/InnoDB) e o RocksDB. O benchmark YCSB (Yahoo Cloud Serving Benchmark) será empregado para simular cargas de trabalho variadas, permitindo a análise quantitativa dos resultados.

O ambiente experimental será composto pelo SSD Kingston NV2 1TB M.2 NVMe PCIe 4.0 (3500 MB/s read, 2100 MB/s write, interface PCIe 4.0 x4, arquitetura DRAM-less com 3D NAND). Este hardware foi escolhido por ser representativo de SSDs NVMe Gen4 de custo acessível e amplamente disponíveis no mercado. O sistema operacional será Linux (kernel 5.15 ou superior) com suporte otimizado a NVMe, CPU com múltiplos threads para cargas concorrentes, e memória RAM mínima de 16GB para evitar interferências de swap.

Serão executados seis workloads padronizados do YCSB, cada um com dataset fixo de 10 milhões de registros: (A) Update Heavy - 50% leituras, 50% atualizações; (B) Read Mostly - 95% leituras, 5% atualizações; (C) Read Only - 100% leituras; (D) Read Latest - 95% leituras de registros recentes, 5% inserções; (E) Short Ranges - 95% scans em intervalos curtos, 5% inserções; (F) Read-Modify-Write - 50% leituras, 50% read-modify-write. Cada workload será executado com diferentes níveis de concorrência (1, 10, 50, 100 threads).

As métricas coletadas incluirão: (1) Throughput (operações/segundo); (2) Latência (média e percentis P50, P95, P99, P99.9); (3) Write Amplification Factor - razão entre bytes fisicamente escritos no SSD versus bytes logicamente escritos, monitorado via iostat, blktrace e métricas internas das engines; (4) Utilização de recursos (CPU, memória, IOPS, bandwidth).

O procedimento de execução para cada combinação consistirá em: warm-up do sistema, execução do workload por 10 minutos, coleta de métricas em tempo real, repetição 3 vezes para reprodutibilidade, e limpeza de cache entre experimentos. Os dados serão analisados com estatística descritiva, gráficos comparativos (box plots para latência, gráficos de linha para throughput), e testes de significância estatística (teste t de Student) para determinar relevância das diferenças observadas.

Resultados Esperados

Com base na literatura existente e nas características do hardware NVMe, espera-se observar padrões específicos de desempenho para cada estrutura.

Para LSM-Trees (RocksDB), antecipa-se throughput superior em workloads write-intensive (Workload A e F), possivelmente com margens reduzidas em comparação com resultados históricos em SSDs SATA, devido às melhorias de latência do NVMe que também beneficiam B-Trees. Espera-se WAF potencialmente reduzido em comparação com SSDs de gerações anteriores, dado o paralelismo interno e eficiência de garbage collection de SSDs NVMe modernos, além de latências de leitura pontual competitivas quando otimizadas com Bloom Filters adequados.

Para B-Trees, prevê-se vantagem significativa em leituras pontuais (point queries) nos workloads B e C, beneficiando-se da baixa latência de acesso aleatório do NVMe, desempenho superior em range scans (Workload E) devido à organização sequencial das páginas folha, porém WAF potencialmente mais crítico em workloads de alta escrita, podendo impactar a longevidade do SSD.

Espera-se identificar "pontos de inflexão" onde uma estrutura supera a outra dependendo da proporção de leituras/escritas, e possíveis cenários onde as diferenças são minimizadas pelas características do hardware NVMe.

Este trabalho pretende contribuir com: (1) dados empíricos atualizados sobre desempenho em hardware NVMe Gen4, preenchendo lacuna na literatura; (2) orientação prática para arquitetos de software na escolha de SGBDs; (3) análise de custo-benefício considerando performance e longevidade do hardware; (4) avaliação em hardware acessível, tornando resultados aplicáveis a pequenas e médias empresas; (5) protocolo experimental reprodutível para pesquisas futuras; (6) validação de hipóteses sobre como características do NVMe (baixa latência, alto paralelismo, multi-queue) afetam trade-offs clássicos entre estruturas write-optimized e read-optimized.

Cronograma de Atividades

Atividades planejadas	Nov/25	Dez/25	Jan/26	Fev/26	Mar/26	Abr/26	Mai/26	Jun/26
Planejamento,revisão bibliográfica e entrega do projeto de pesquisa	X							
Preparação, setup do ambiente (Hardware)	X	X						
Preparação, configuração das engines (B-Tree e LSM-Tree)		X						
Configuração do benchmark (YCSB)		X						
Execução dos testes e coleta de métricas			X					
Análise estatística dos dados e entrega dos resultados preliminares				X				
Redação TCC: Seções de metodologia e resultados				X	X			
Redação TCC: Seções de análise/discussão e conclusão					X			
Revisão final, formatação e entrega do TCC						X		
Preparação e entrega dos slides							X	X
Apresentação (Banca) e entrega da versão revisada do TCC								X

Referências

ARULRAJ, J.; PAVLO, A.; DULLOOR, S. R. A Case for Flash-Aware Data-Intensive Systems. CIDR, 2018.

BAYER, R.; MCCREIGHT, E. Organization and Maintenance of Large Ordered Indices. Acta Informatica, 1972.

CHANG, F. et al. Bigtable: A Distributed Storage System for Structured Data. OSDI, 2006.

COOPER, B. F. et al. Benchmarking Cloud Serving Systems with YCSB. SoCC, 2010.

META. RocksDB: A High Performance Key-value Store. Disponível em: https://rocksdb.org. Acesso em: 20 out. 2025.

O'NEIL, P. et al. The Log-Structured Merge-Tree (LSM-Tree). Acta Informatica, 1996.

RAMAKRISHNAN, R.; GEHRKE, J. Database Management Systems. McGraw-Hill, 2003.