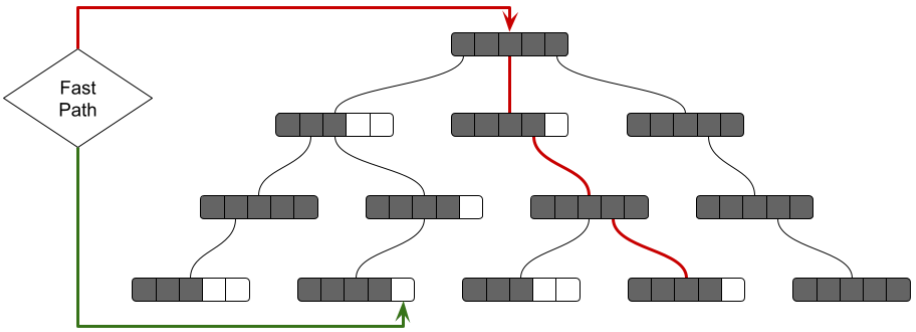
CAS CS 561: Data Systems Architectures
Data-intensive Systems and Computing Lab
Department of Computer Science
College of Arts and Sciences, Boston University
http://bu-disc.github.io/CS561/

CS561 Spring 2025 - Research Project

Title: *Cache-Awareness for Near-Sorted Indexing*

Background: B+trees are widely used for indexing in databases and key-value stores due to their efficiency in handling large, ordered datasets. However, traditional B+tree implementations can suffer from performance bottlenecks when ingesting near-sorted data, particularly due to frequent tree traversals and cache misses. Optimizing for cache awareness and leveraging the near-sorted nature of the data presents an opportunity to improve insertion performance without compromising search efficiency.



In a B+tree, nodes may be page-sized (e.g., 4 KB) to optimize for disk or memory paging, while cache lines are much smaller (e.g., 64 bytes). This disparity can lead to inefficiencies: while reducing the number of key comparisons a binary search within a node often accesses non-contiguous memory locations. These accesses may evict useful data from the CPU cache, increasing cache misses and degrading performance.

Objective: The objective of this project is to design and evaluate a B+tree optimized for near-sorted data ingestion by:

1. Implementing a B+tree that uses the last accessed leaf as a fast path for future inserts.
2. Append policy for the fast leaf instead of in-place inserts.
3. Investigating policies for handling unsorted leaves:
    - Sorting leaves on split.
    - Sorting leaves after moving the fast path to another leaf.
    - Keeping leaves always sorted.
4. Exploring cache-friendly optimizations, such as branchless binary-search algorithms.

Responsible Mentor: Konstantinos Karatsenidis