

FITting Tree: A Data-Aware Index

Goals Determine if the FITting tree provides improvement in memory efficiency over a B+ tree

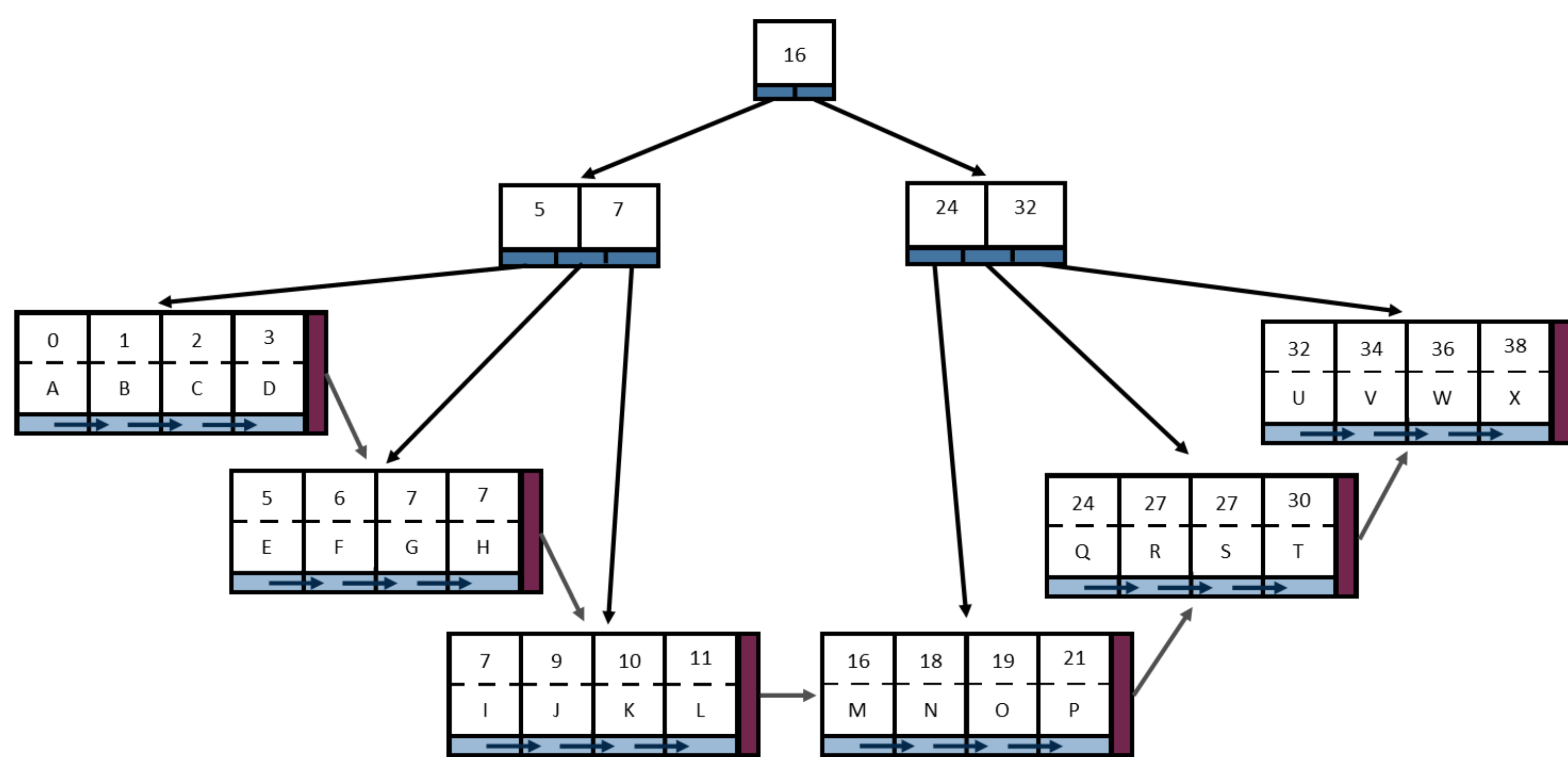
Determine if FITting tree performance is comparable to that of a B+ tree

Structures Created functional B+ and FITting trees from scratch, with both structures having...

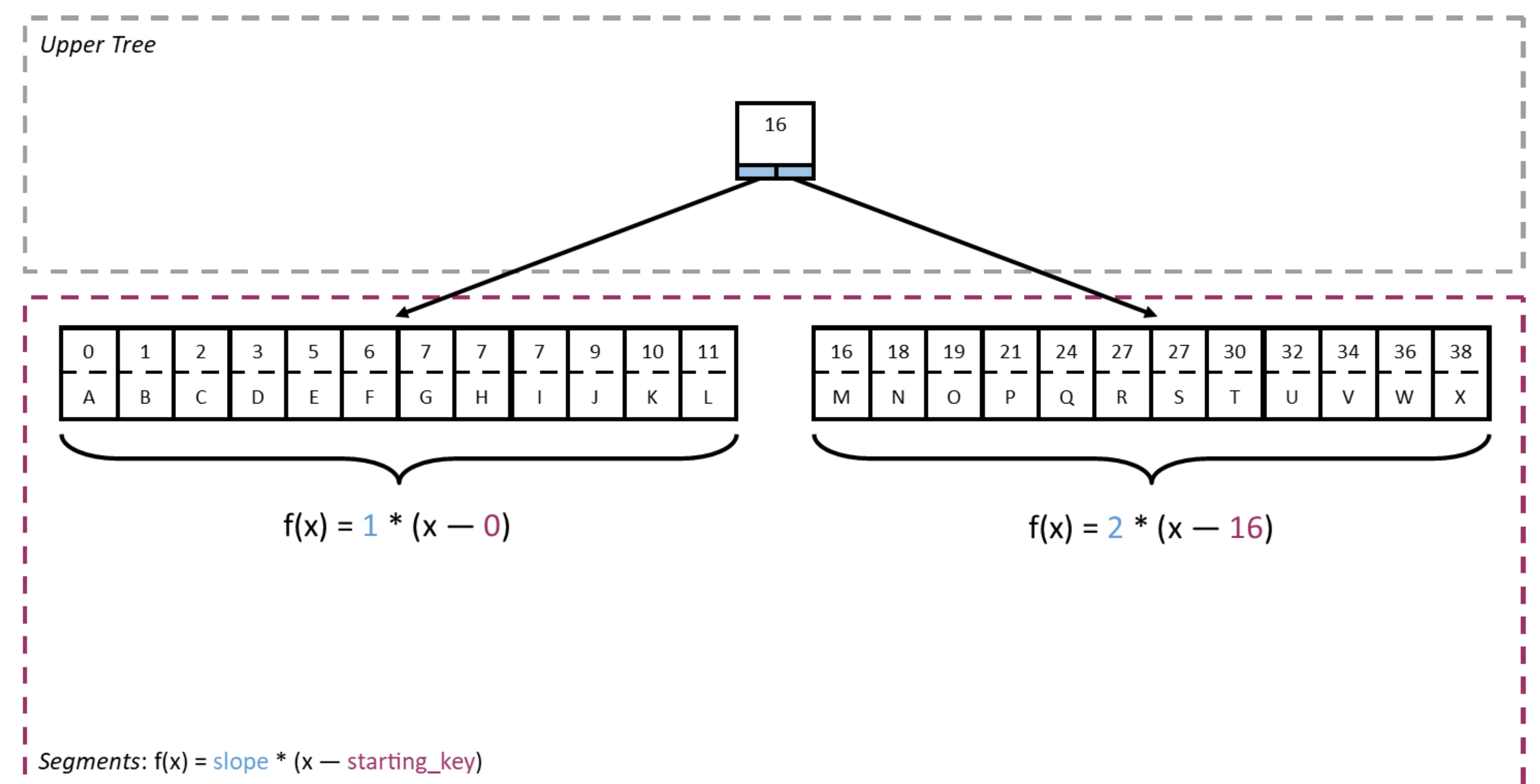
Caching

Disk-Residence

B+ Tree



FITting Tree



Inner Nodes

- Key/pointer pairs indicate pathing for finding values under a given key
- Key ranges are naturally expressed through sorted nature of stored pairs

Leaf Nodes

- Formed by groups of sorted key/value pairs according to key order
- Each leaf node is linked to its neighbors, allowing for the collection of values in multiple nodes

Upper Tree

- Uses B+ tree as an upper management layer to find segments quickly
- Keys in the B+ tree naturally map over the ranges of stored segments.

Segments

- Map over a region of sorted key/value pairs using a segment function
- Segment functions are *linear approximations* given a calculated slope and starting key

Test Design Inspired by TPC-C, a common DBMS benchmark

Ramp up to steady state with percentage-based mix of transaction types

Hold steady state for 8+ hours and record

Metrics Total Memory Efficiency (data memory / total structure memory)

Throughput of response times by transaction type

Tools C++

CMake/Make

g++