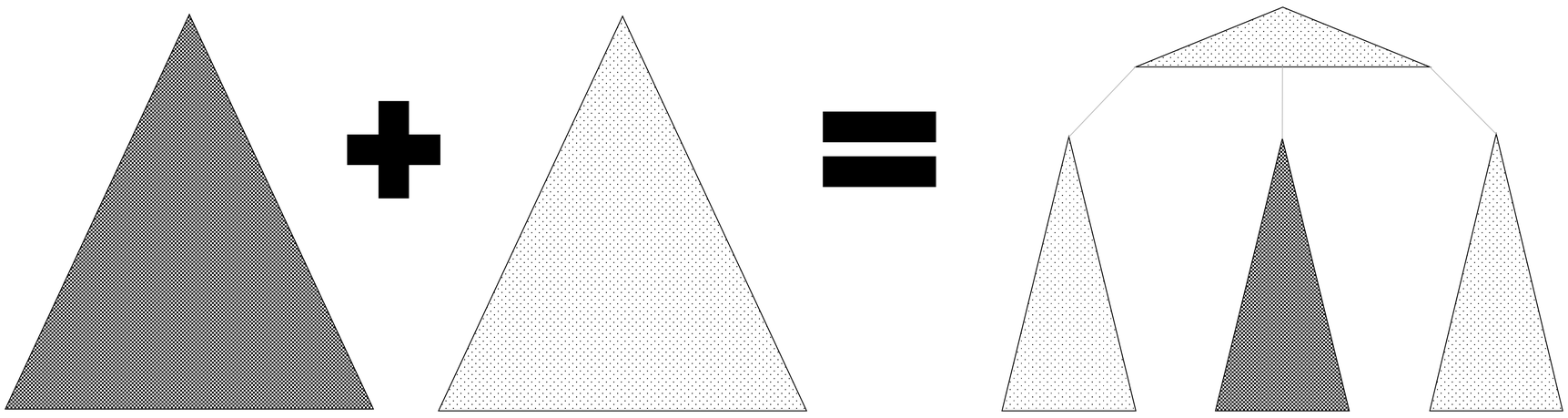
E-Tree: An Ever-Evolving Tree for Evolving Workloads

HARVARD John A. Paulson School of Engineering and Applied Sciences

Graham Lustiber **Advisor: Stratos Idreos**

Write Optimized + Read Optimized = Best of Both Worlds

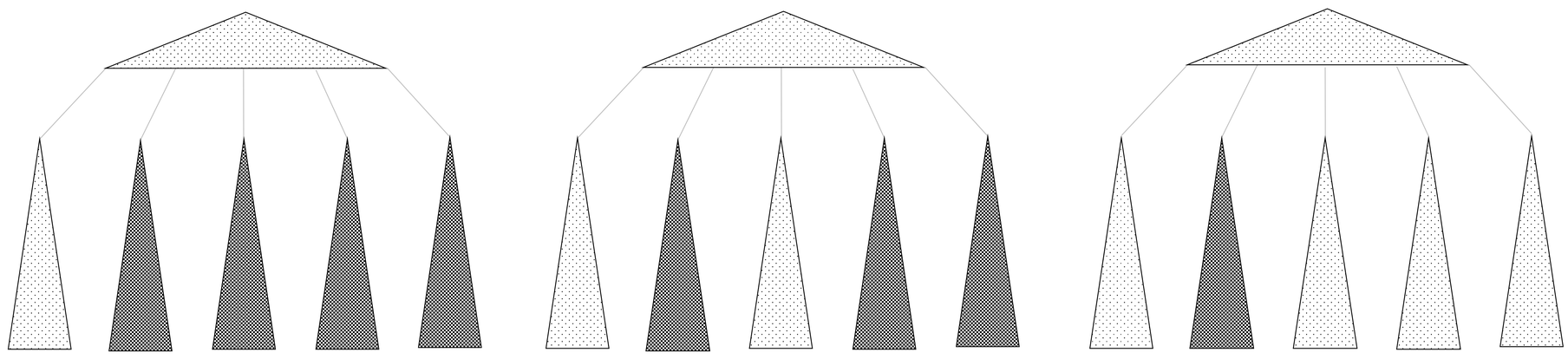


B+-Tree **Cache-Conscious Tree** **E-Tree**

Key Idea: Mixed memory layouts for mixed workloads across different key/value subranges

By tracking access patterns, subtrees dynamically reshape

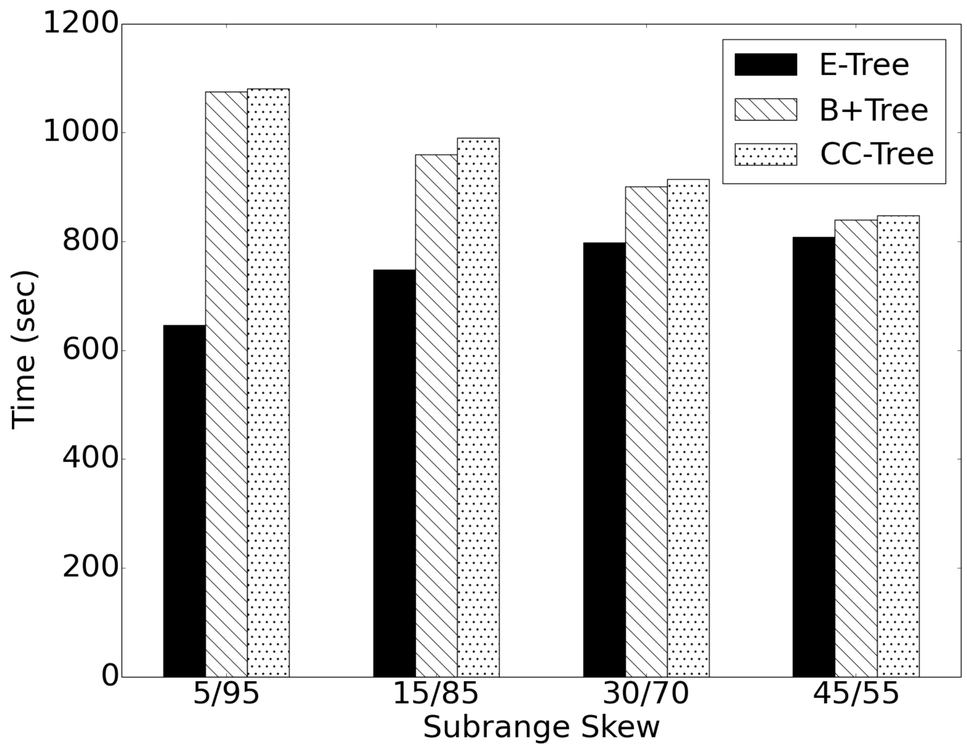
T_0: 95% Writes/5% Read → T_1: 70% Writes/30% Reads → T_2: 40% Writes/60% Reads



Low Overhead
Simple counters on each node with sampling

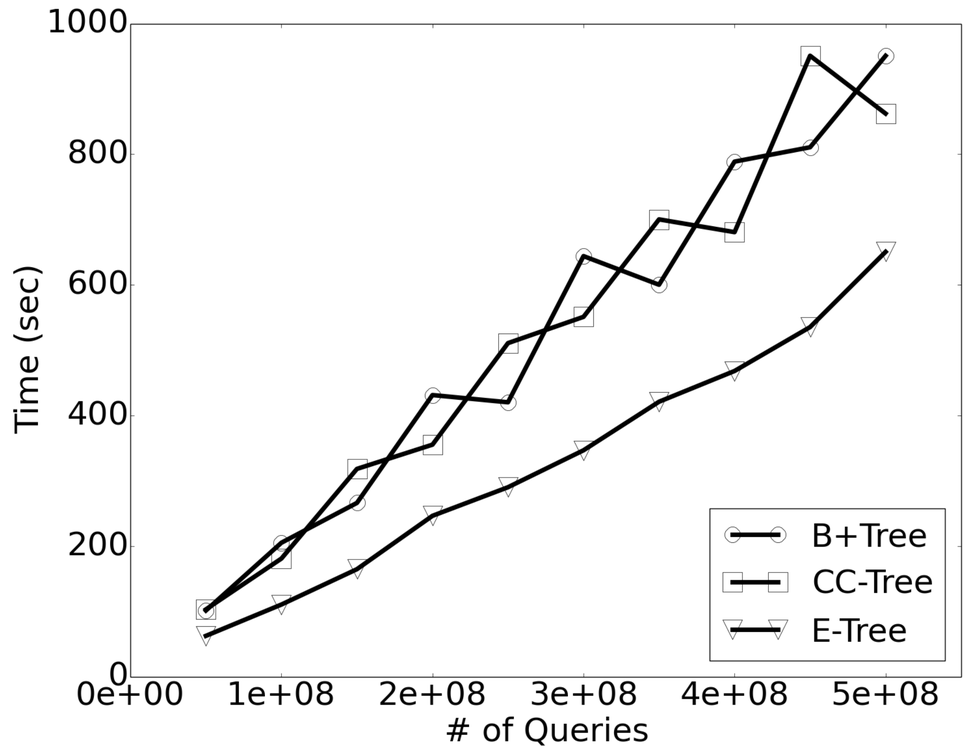
Automatic Optimality
Convergence to optimal memory layout

Flexible
No restriction on concurrency or operations



Setup: 2 billion key/value pairs, with queries to key subranges either read-skewed or write-skewed

Split workloads: the more skewed the subranges, the better E-Tree performs



Setup: same as before, but subrange skews randomly flip from read-skewed to write-skewed and vice versa

Shifting workloads: E-Tree adapts to new access patterns