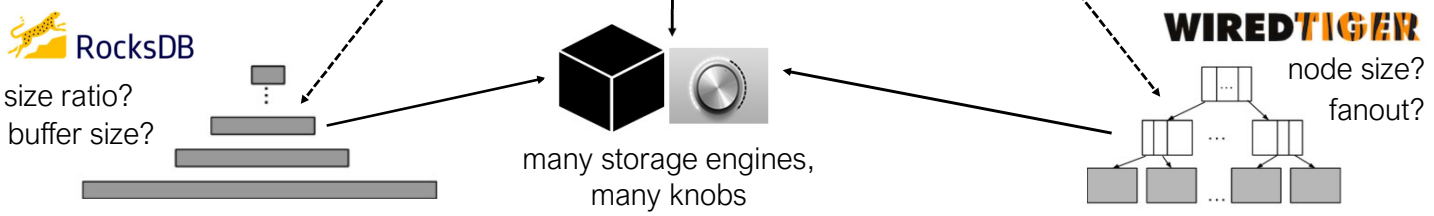
HARVARD John A. Paulson School of Engineering and Applied Sciences

From Worst-Case to Average-Case Analysis: Accurate Latency Predictions for Key-Value Storage Engines

Meena Jagadeesan and Garrett Tanzer | Advised by Subarna Chatterjee, Stratos Idreos

THE GOAL: Which storage engine and tuning is best for an application?

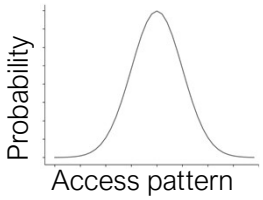
Put((5, -10)), Get(5), Put((7, -17)), Get(10), Get(5), Get(7), Get(10), Put((5, -17))



AVERAGE-CASE ANALYSIS: A Distribution-Aware Framework

Workload Feature Vector

Distributions



Operations

Storage Engine Configuration

engine = RocksDB

size ratio = 10

buffer size = 1GB

filter memory = 2GB

Unified

Distribution-Aware Cost Model

Workload I/Os

EXAMPLE: Get Cost on LSM

"Early stopping"

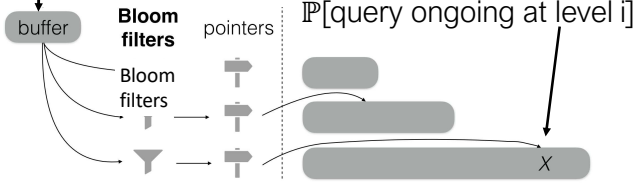
$$C_{\text{get}} = (1 - C_{\text{buffer}}) + \sum_{i=1}^{L} p_i * C_i$$

p_i: ℙ["false" access at level i]

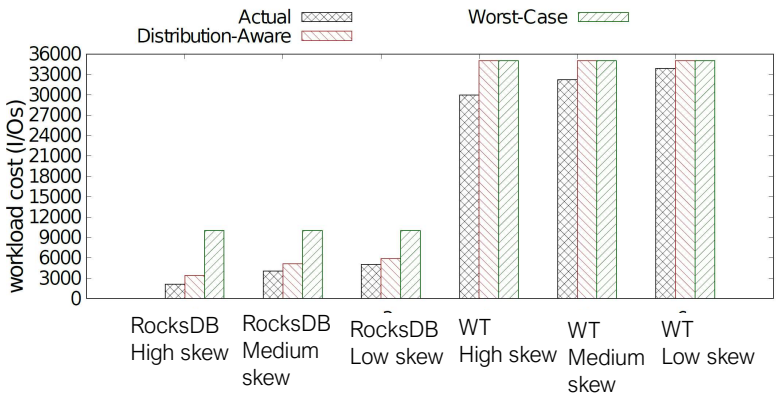
C_{buffer}: ℙ[key in buffer]

C_i: ℙ[query ongoing at level i]

False positive rate



VERIFICATION: Key-Value Stores



Distribution-aware model captures workload I/Os!

10 mil keys; workload (20,000 ops): 50% reads/50% writes

USE CASES: How can these models be utilized?

Compare designs

Automate selection

Search design space

Create self-designing storage engine