POCKETDATA

Pocket-scale data management research

Gourab Mitra, Carl Nussele, Gokhan Kul, Grant Wrazen, Lakshmi Ethiraj, Geoffery Challen, Lukasz Ziarek, Oliver Kennedy

University at Buffalo
The State University of New York

Introduction

Research on ”Small Data”. Dealing with data management challenges in personal and per-device interactions. Fuelled by smart devices and low-cost embedded computing platforms. At small scales, the law of large numbers is inapplicable and there is insufficient noise to absorb all of the outliers

Mobile Databases

- Most mobile apps use embedded databases.* Operate in heterogeneous environments (varying battery, RAM, CPU, storage).
- Focus more on efficiency and share resources with other apps on the phone.

Whereas Database Servers:

- Are tuned for continuous high-throughput query processing
- Have exclusive access to all resources of a machine
- Focus on performance (throughout, latency), not efficiency.

POCKETDATA Toolchain:

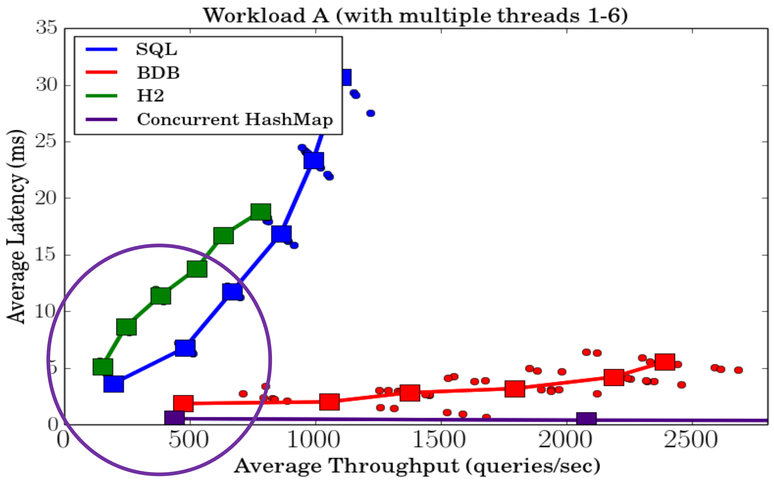
- Nexus 6 phones. Custom Android ROM with instrumentation in the SQLite native layer
- YCSB Benchmark as an app
- Log collected about different events - DB connections, schemas, statement compilations, and queries into Android buffer
- Experiments to emulate different kinds of workloads

Importance:

- Enabling research into cross-cutting communities working on data management, real-time and embedded devices, programming languages and operating systems.
- Helping app developers choose the best database
- Identifying performance bottlenecks in a database.

Our Focus

Unlike traditional database research, we focus on low-throughput, bursty workloads under resource constrained environments.



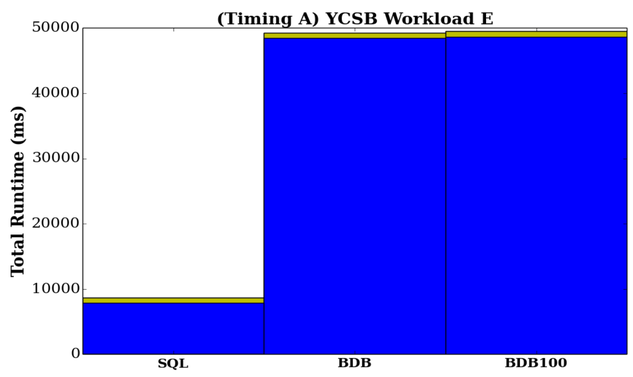
* *Curino Et. Al.* http://hdl.handle.net/1721.1/62241

Comparing mobile databases

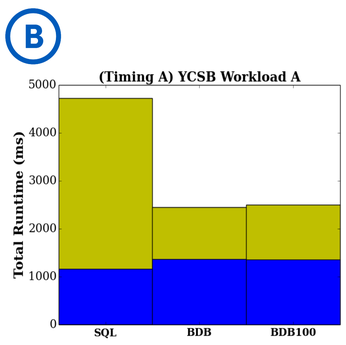
Comparisons among SQLite, BDB and BDB100 [See Figure below]:

A. SQLite is particularly tuned for scan-heavy workloads (YCSB E).
B. BDB wins for write-heavy workloads (YCSB A).
C. SQLite performs better for read-heavy workloads.(YCSB B)

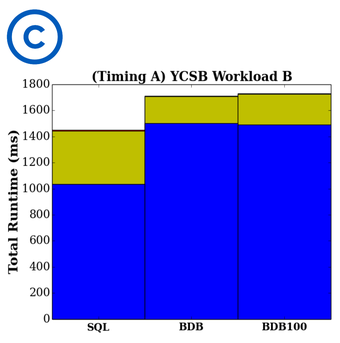
A



B



C



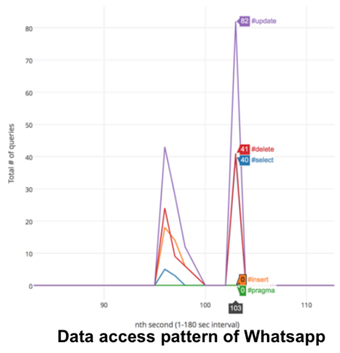
Studying query logs

Importance

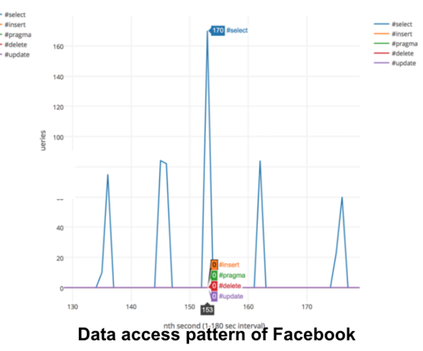
- Reveals the patterns in an application’s usage of database
- Identify most common usage patterns
- Identify outliers in usage patterns

Characteristics

- Most queries have inherent similarity in structure because they are machine-generated
- Bursty, variable, and hard to summarize as a simple distribution of queries
- Doesn’t contain cues about beginning and end of users tasks



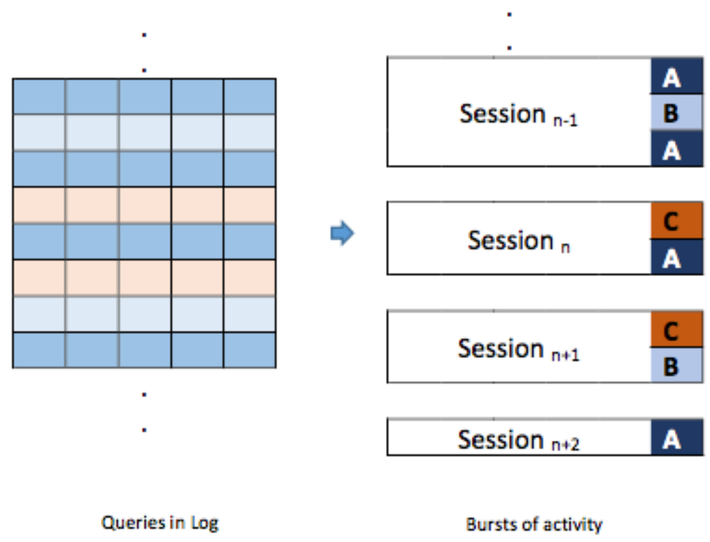
Data access pattern of Whatsapp



Data access pattern of Facebook

Session Identification

- Traditional approaches to session identification fail in our scenario
- Connection time, Timeout and Semantic segmentation
- User tasks keep switching between background and foreground ; Multi-tasking
- Database Sessions as a subset of repetitive logical user tasks
- Automatic session detection



Sessions and User Tasks

Session Similarity

- Modular approach for session similarity.
- Considers both query features and activity distribution within the sessions
- Reveals shared activities
- Helps identifying common and unusual behavior patterns
- Variety of application areas such as predicting incoming queries to improving database performance

Applications

- Methodology for automatic benchmark generation from query logs.
- Identification of representative samples from query logs
- Guidance for on-the-fly DB performance tuning.

The ODIn Lab @

odin.cse.buffalo.edu/research/pocketdata/index.html