Data Synthesis for automatically generating Smartphone Database Benchmarks

Gourab Mitra, Oliver Kennedy

University at Buffalo
The State University of New York

Introduction

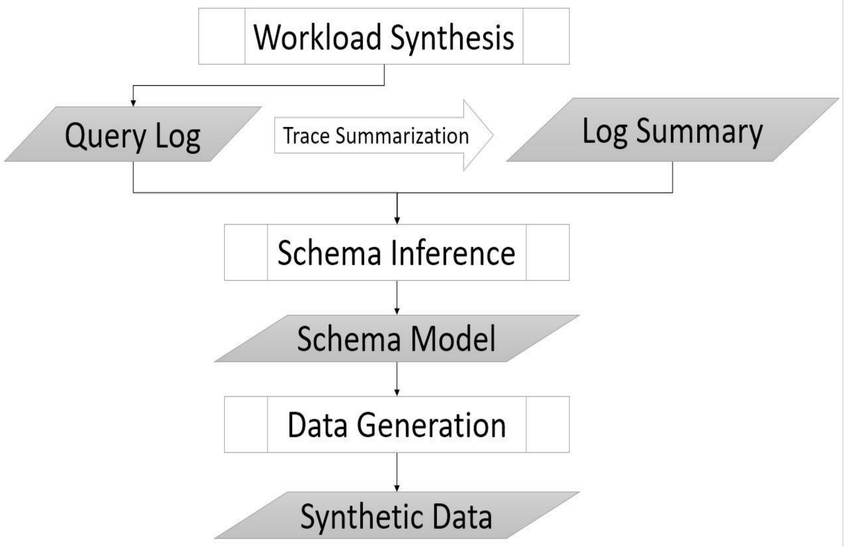
- Database management system - heart of most data services. Unlike server-side databases, smartphone databases not well studied.
- To build applications which evolve with changing requirements- correct choice of database
- Benchmarks help decide *correct* choice of databases.

Databases Benchmarks

- Domain specific. No *one-size-fits-all* benchmark.
- Enable comparison. Help with performance tuning.
- Need to *characterize* an application to prepare a relevant benchmark.
- Most benchmarking suites employ data generators - rely on formal schema of schema.
- Generation of a benchmark from a query log is challenging.
- Database schema needs to be inferred from just the query log.

Our goal is to build an automatic generator for smartphone database benchmarks. In order to do that, we must synthesize data to be used in the benchmark.

This poster discusses ongoing research on design of a methodology to infer database schema from a query log.

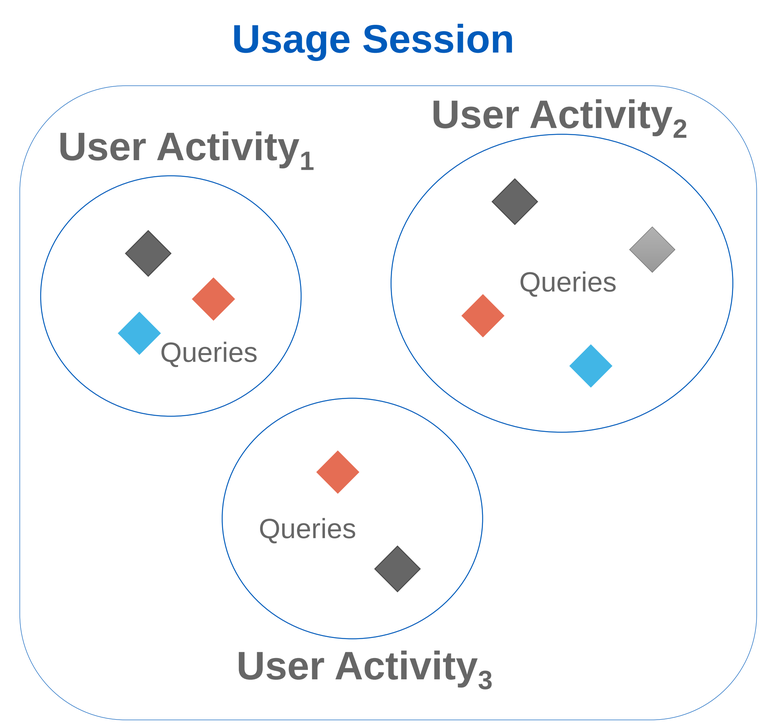


System Design

- Input to the system - a query log of an application generated on a smartphone database.
- Output - synthetic data in multiple csv files, each corresponding to a table in the inferred schema.

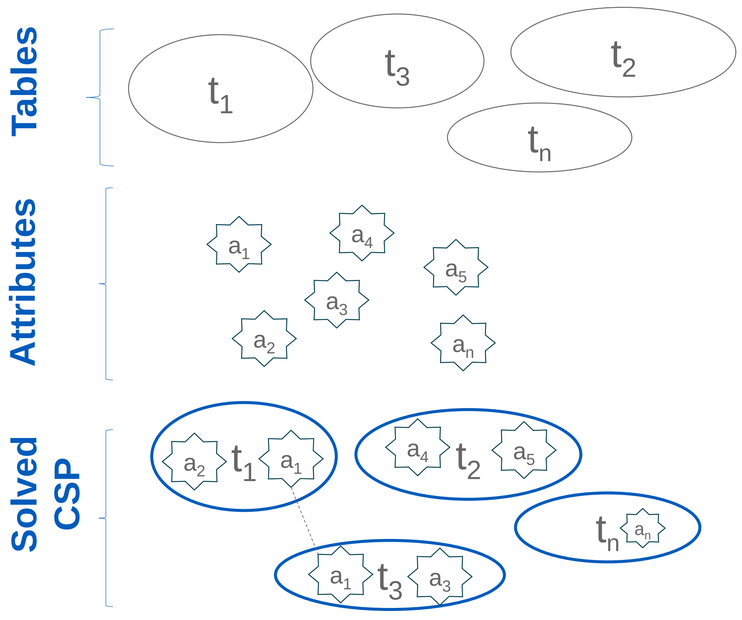
Trace Summarization

- Summarizing workloads - identifying a *representative* subset that captures essence of a larger workload.
- Automatic characterization of workload – identify user activity and usage sessions.
- Flexible – ability to modify trace characterizations based on parameters
- Summarization is optional for data generation but necessary for building a benchmark

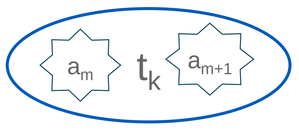


Schema Inference

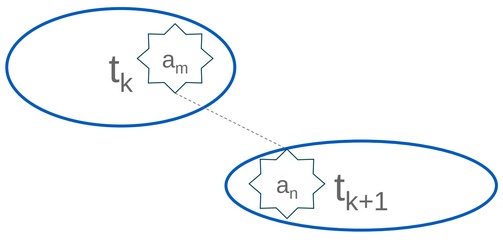
- Features extracted from the queries contain *hints* about the underlying schema. Formula a **Constraint Satisfaction Problem** (CSP)



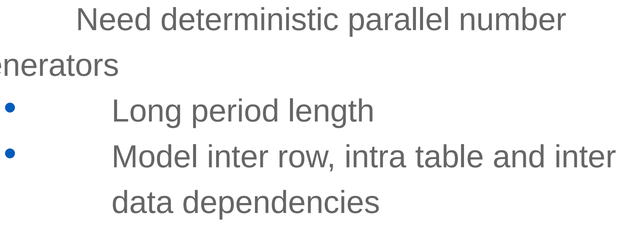
- INSERT, DELETE and UPDATE statements - parts of the solution directly



- JOIN conditions - identify relationships



- SELECT statements would help us identify more constraints for our CSP.



- Efficiency of schema inference process depends on the efficiency of the workload summarization process

- One way to deal with this issue is to output a set of candidate schemas
- Another solution could be the use of a likelihood function as an optimization goal.

Data Generation

- Inferred schema translated to a formal schema specification.
- Deterministic, works in parallel threads , able to generate complex data sets
- Need deterministic parallel number generators
  - Long period length
  - Model inter row, intra table and inter table data dependencies
- Tools like PDGF (Parallel Data Generation Framework) work with XML schema definitions.

Related Work

- Seltzer *et al.* have demonstrated the need for application specific benchmarking. One such way is trace based. Application traces are preprocessed to produce user and activity profiles. Profiles are used to generate representative workload for the system.
- Zhang et al have modelled database schema inference from a query log as a Constraint Satisfaction Problem (CSP).
- Rabl et al. present a large scale data generation framework for benchmarking. It uses XML files for supplying a schema model.

Conclusion

We introduce the problem of data synthesis for automatically generating mobile database benchmarks. Such a system can be built by :-

- **Schema Inference:** from an application's query log.
- **Data Generation:** by feeding schema model to standard data generators.

The ODIn Lab @

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