Holistic Optimization of Database Applications
Karthyk Ramachandra, IIT Bombay
with R Guravannavar (IIT Hyderabad), M Chavan & S Sudarshan (IIT Bombay)

The Problem
- Applications invoke DB queries/Web Service requests repeatedly (with different parameters)
- Synchronously (blocking on every request)
- Naive iterative execution of such queries: inefficient
  - No sharing of work (e.g., Disk I/O)
  - Network round-trip delays

Soln 1: Batching
- Repeated invocation of a query automatically replaced by a single invocation of its batched form
- Enables use of efficient set-oriented query execution plans

Soln 2: Asynchronous Submission
- Repeated synchronous invocation of queries automatically replaced by asynchronous submission
- Application can perform other work while query executes
- Multiple queries could be issued concurrently

Approach
- Analyze application programs along with embedded queries
- Transform programs using equivalence rules and semantics preserving reordering
- Rewrite queries using decorrelation, APPLY operator, etc.
- DBridge: Implementation for Java/JDBC, using Soot framework (http://www.cse.iitb.ac.in/dbms/dbridge)

Performance gains (upto 7x)!

Program Transformation
- Identical API for Batching and Asynchronous approach

Input program
- while (category != null) {
  qt.setInt(1, category);
  int count = qt.executeQuery();
  sum = sum + count;
  category = getParent(category);
}

Output program after Loop Splitting
- while (category != null) {
  qt.setInt(1, category);
  int count = qt.executeQuery();
  sum = sum + count;
  category = getParent(category);
}

Parameter Batch (temp table)
- Build/Update DDG
- Split Loop
- Decompile
- Reordered program before Loop Splitting

Experiment 1: DB Query invocation
(Auction System Benchmark)

Experiment 2: Web Service invocation
(Batching not possible here as there is no set-oriented interface exposed by the web service)