

Static Typing Meets Adaptive Optimization: A Unified Approach to Recursive Queries

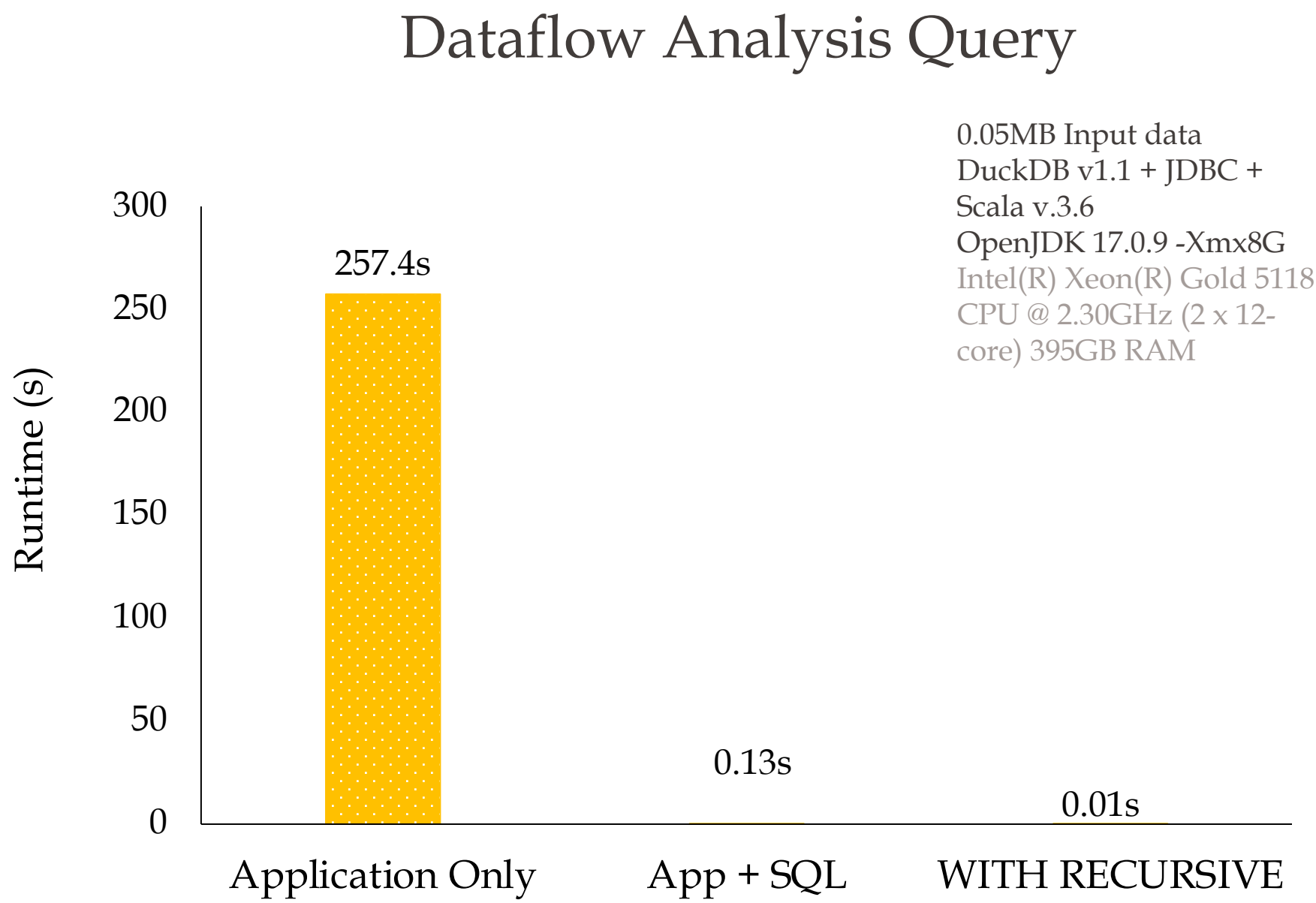
Anna Herlihy, Anastasia Ailamaki, Martin Odersky

Key data analyses are fixed-point problems

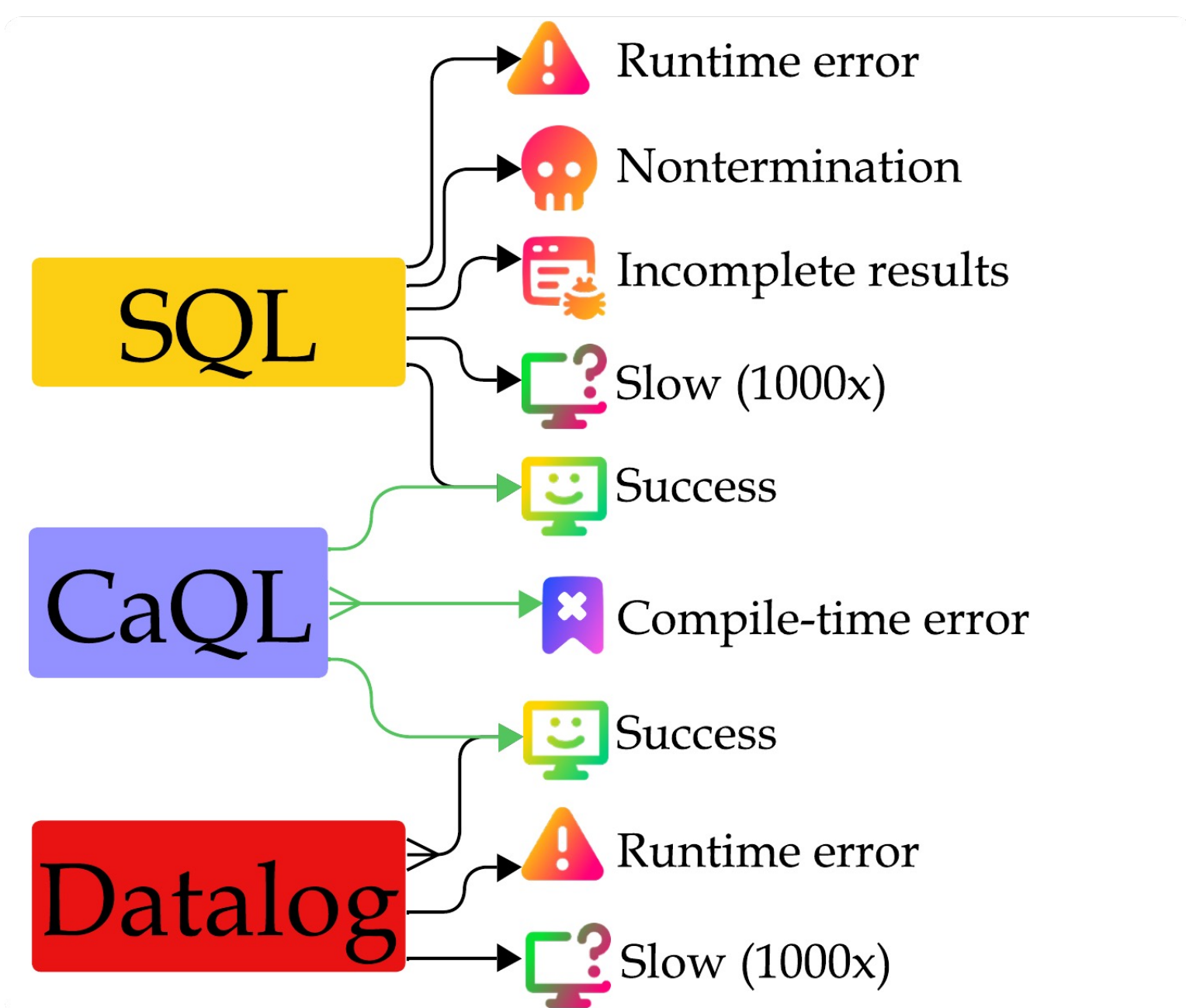
Fixpoint of f is $f(x) = x$
Use cases: reachability queries, program analysis
(Java, tensors, smart contracts), network analysis, etc.
hierarchical data, social media analysis, bill-of-materials, ...

SQL'99 added WITH RECURSIVE
SQL becomes Turing-Complete: new classes of problems
Fixed-points can now run in relational databases

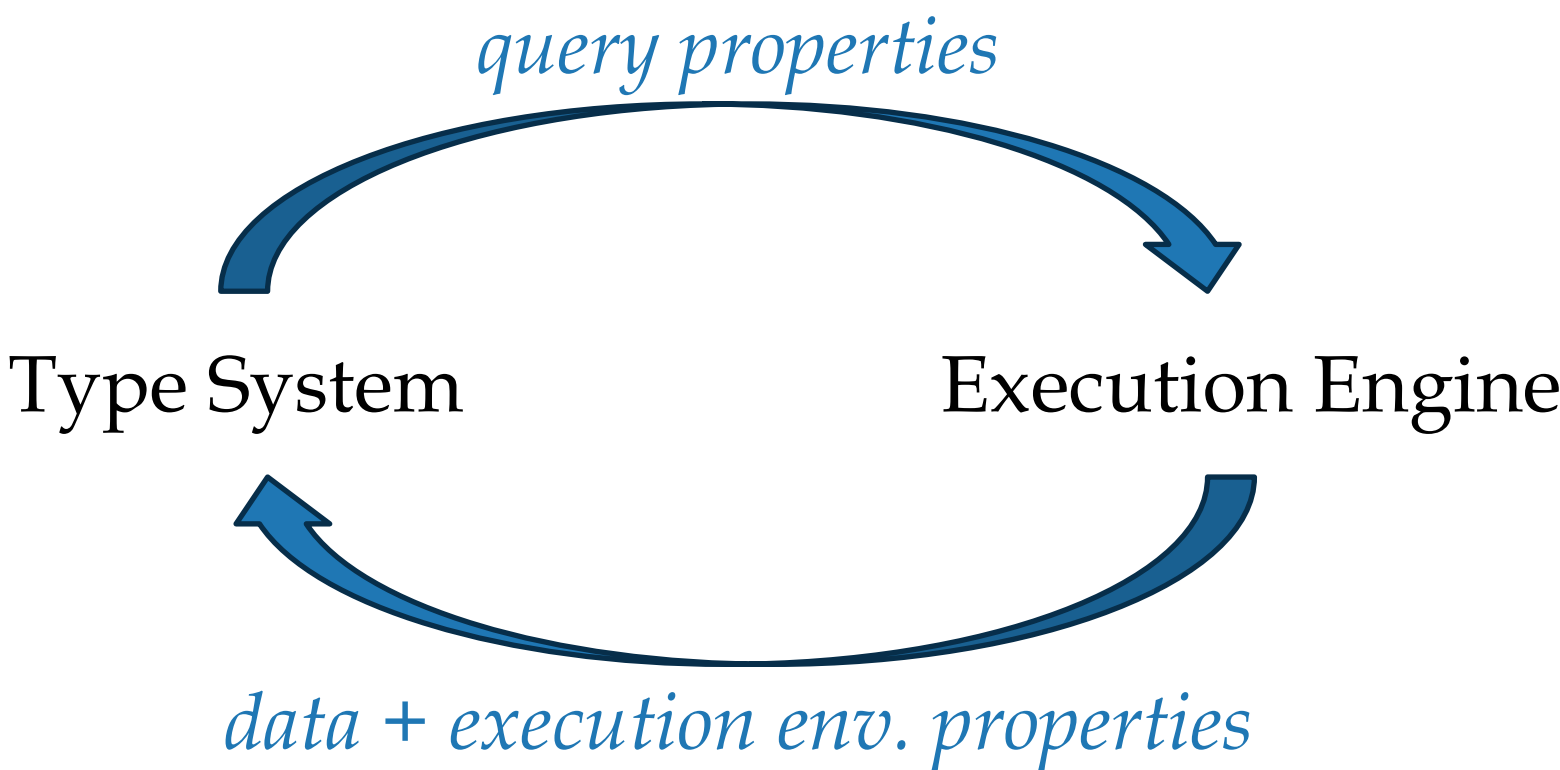
WITH RECURSIVE can show up to 20,000x speedup over application-only implementations



WITH RECURSIVE unlocks new ways for queries to « go wrong »



Compilers are conservative. More information about data and execution environments allow type systems to relax constraints and unlock new functionality

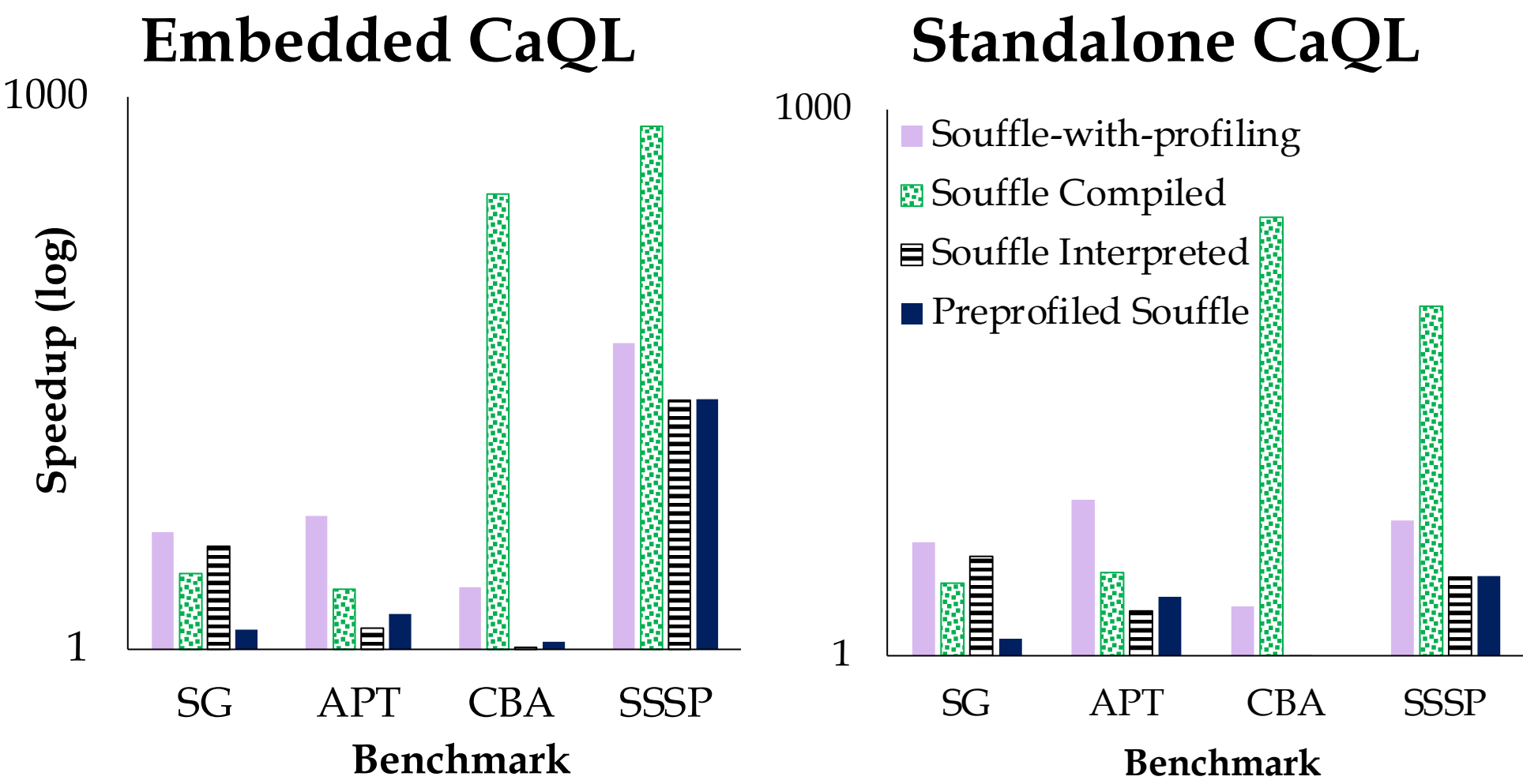
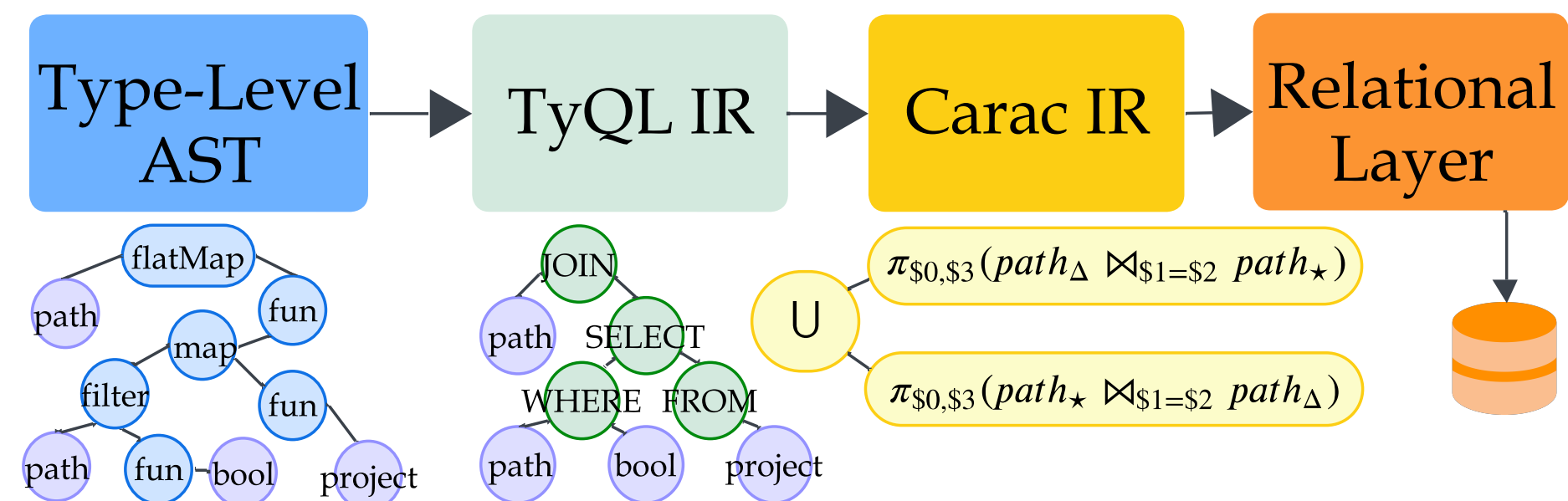


Runtimes are speculative. Better guarantees about runtime program behavior enables more aggressive optimization and better performance

CaQL unifies two existing systems : TyQL and Carac

TyQL is a language-integrated query library providing type-safe queries at compile-time
Carac is an adaptive, JIT Datalog engine providing runtime optimization of recursive queries

CaQL: unification of TyQL + Carac
Goal: Expressive, safe, performant recursive queries



Benchmark	Data Size	Benchmark Domain	Linear
Same-generation (SG)	100MB	Recursive SQL	No
Andersen's Points-To (APT)	36K	Program analysis	Yes
Constraint-based analysis (CBA)	180K	Datalog benchmark	No
Single-source shortest path (SSSP)	10MB	Graph query	Yes

CaQL unifies the type system and runtime
to improve performance and make type systems more precise