

- High Level overview of a potential thesis work
- Detailed map of tasks that are done and should (can) be done.
- Publication-orientated map of possible routes ahead

Centralized SQL databases

- SC + SER
- Relational Tables

Real-World ECDS Implementations

- Replicated complex objects
- Geo-distributed objects
- EC + adhoc CC

Centralized SQL databases

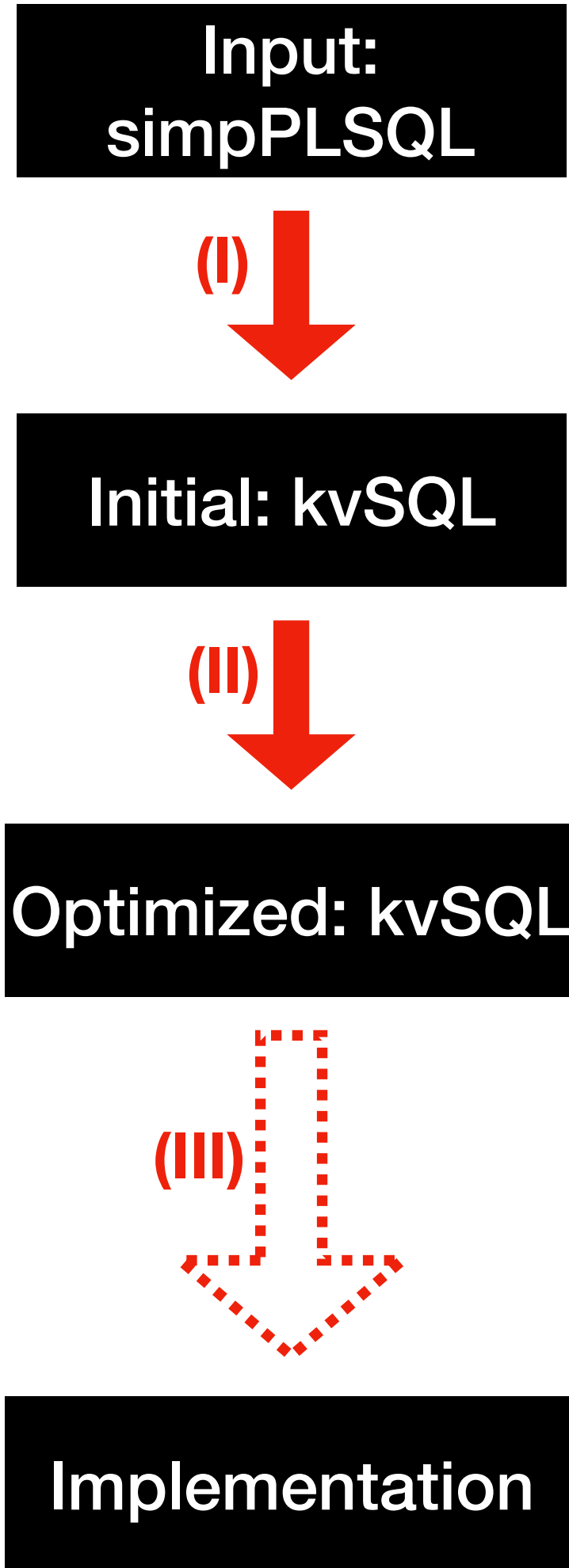
- SC + SER
- Relational Tables

Generic Distributed Key Value Store Interface

- Replicated generic kv objects
- Generic CC annotations
- Denormalized

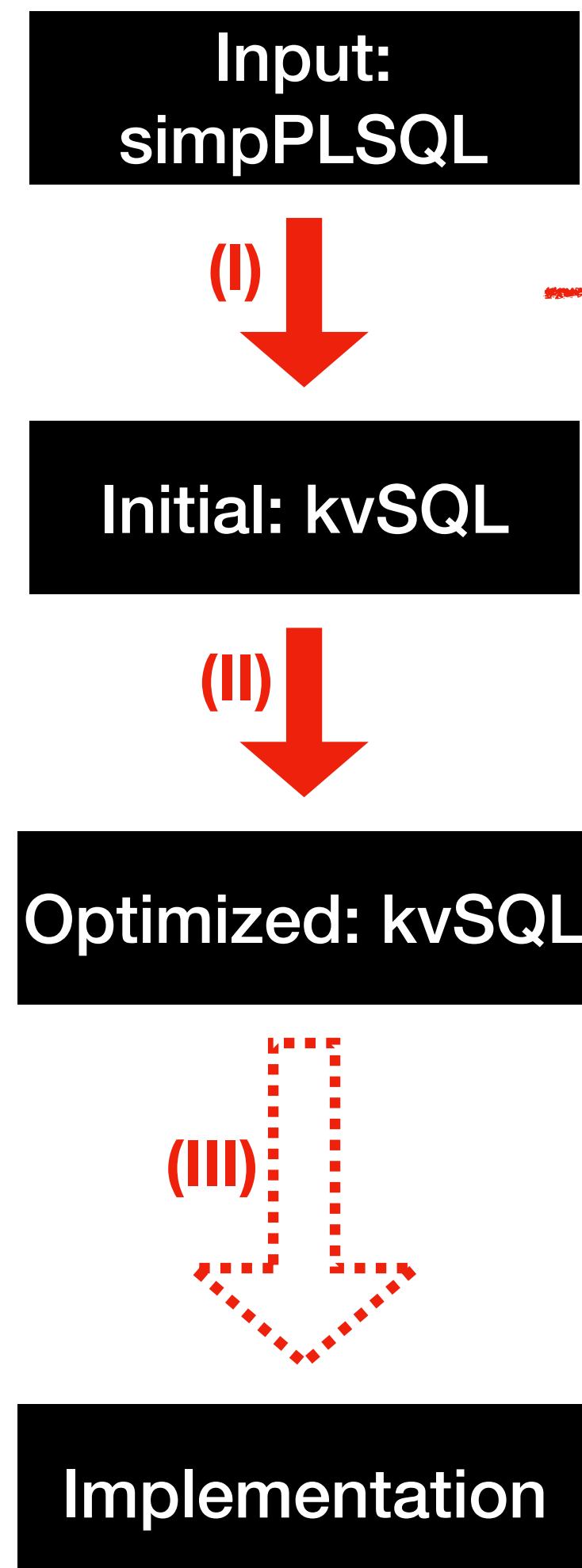
Real-World ECDS Implementations

- Replicated complex objects
- Geo-distributed objects
- EC + adhoc CC



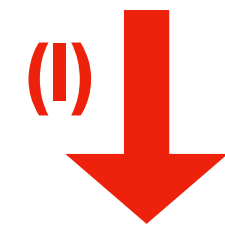
Must be “equal” to the input

Must be the “optimal” version

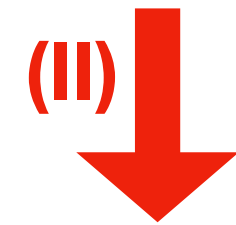


- (I) A set of syntactic program rewrite rules:
 - Should prove that all SER executions of kvSQL version are equal to a SER execution in simPLSQL
- (Done) Syntax for both languages are defined and a benchmark application is rewritten
- (Almost done) Rewrite ruled are vaguely determined -> must be formalized
- (not done) Op semantics and correctness guarantees must be defined and proved.

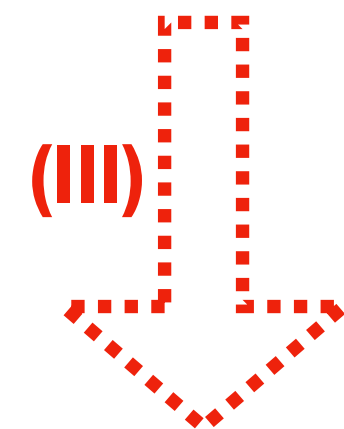
Input:
simpPLSQL



Initial: kvSQL



Optimized: kvSQL

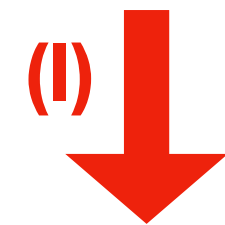


Implementation

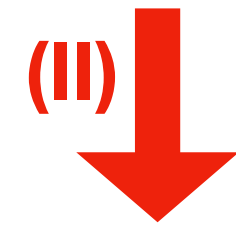
Two possibilities:

- Start from the strongest version
 - Systematically weaken (?)
 - Show that by weakening no anomaly is introduced
 - prove optimality
-
- Start from the weakest version
 - Systematically look for anomalies
 - update the annotations (strengthen the program) (?)
 - prove similar anomalies will not occur (or not)
 - prove global optimality (?)

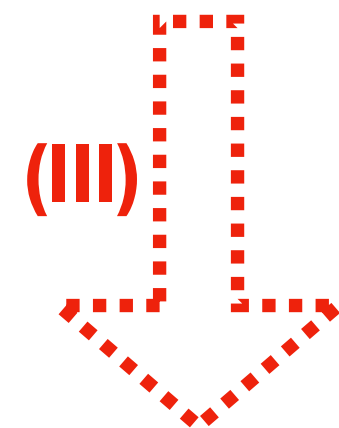
Input:
simpPLSQL



Initial: kvSQL



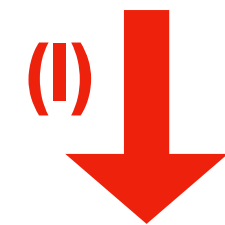
Optimized: kvSQL



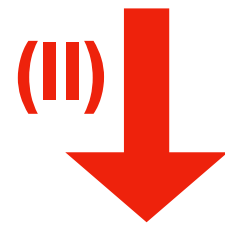
Implementation

- Systematically look for anomalies:
 - Using Kartik's analysis technique find (bounded) anomalies (tool does not exist yet)
 - Define a language Specify the anomalies (is not defined, depends on the tool)
 - Define the strengthening rules (vaguely determined, depends on the the anomaly specification language)
- update the annotations (strengthen the program)

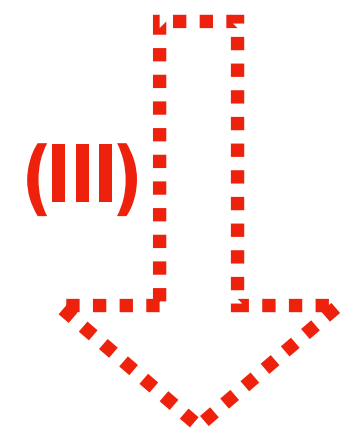
Input:
simpPLSQL



Initial: kvSQL



Optimized: kvSQL



Implementation

- Specify arbitrary real-world systems (?)
- Automatically generate executables programs from the optimized kvSQL version(?)
- Prove it correct (?)

