Contributions

- Defined Locus language:
  - describe *concisely* complex space of optimizations
  - *agnostic* of any specific traversal method
  - *decouple* performance expert role from application expert role

- Implemented a system with flexible API for plugging in:
  - *different* variant selection techniques (optimization space traversal)
  - *collection* of transformations developed internally and externally

- Optimizer and interpreter for the Locus programs:
  - *prune* the space automatically
  - speeds-up the empirical search

Contributions

- Defined Locus language:
  - describe *concisely* complex space of optimizations
  - *agnostic* of any specific traversal method
  - *decouple* performance expert role from application expert role
- Implemented a system with flexible API for plugging in:
  - *different* variant selection techniques (optimization space traversal)
  - *collection* of transformations developed internally and externally
- Optimizer and interpreter for the Locus programs:
  - *prune* the space automatically
  - speeds-up the empirical search

Locus Approach

- Baseline code: defined by the developer, no platform- or compiler-specific optimizations
- Annotated regions of interest (i.e., code regions)
- Program the application of the optimizations for each code region