Delite Compiler Framework

- Compiler framework for easy DSL development
- Framework built on top of lightweight modular staging (LMS)
- DSLs embedded in Scala - leverages scalac front end
- Framework exposes common parallel patterns
- e.g. Map, Zip, Reduce, Filter, GroupBy, Foreach
- Staged DSL program creates parallelism-rich IR
- Applications for Languages

Intermediate Representation (IR)

Domain-Specific IR: Domain operation semantics
- Apply domain-specific optimizations
- Linear algebra simplification
Parallel IR: Common Parallel Patterns
- Expose structured parallelism of domain operations
- Apply loop fusing optimizations
Generic IR: Symbol and Definition
- Apply traditional compiler optimizations
- CSE, Constant Propagation, Code Motion

Delite Hardware Generation Workflow

- High-level transformations
- Dataflow IR
- Lowering Memory resource allocation
- Optimization phase
- Bit-blasting
- Code Generation
- High-level optimizations
- Lowering Memory resource allocation
- Optimization phase
- Bit-blasting
- Code Generation
- Delite User
- Delite Author
- Delite Compiler
- Delite User Interface
- Domain Analysis & Opt.
- Generic Analysis & Opt.
- Parallelism Analysis & Opt.
- Code Generation
- Delite
- Delite
- Delite

Applications

- Predictive Analytics
- Social Network Analysis
- Data Wrangling
- Data Transform

Delite Hardware Generation Workflow

- Automatic design space exploration
- Compute and memory modeling
- Communication and locality aware optimizations
- Mapping parallelism-rich IR to architectural templates
- Dataflow engine generation using MaxJ along with C driver code

Motivation

- Using HLS tools often still requires significant hardware knowledge
- Most HLS tools produce designs that do not utilize off-chip memory efficiently
- Need locality-aware optimizations and custom memory hierarchies
- Design space is extremely large
- Need design estimation tools with autotuning

Existing Solutions

HLS Tools
- Built for hardware designers
- No full-system synthesis
- Good for small/moderate designs

Modeling Tools
- Need C or lower level description
- Need user hints in design, representative inputs

Contact Information

Group Website
http://ppl.stanford.edu

E-mail Addresses (Students)
raghu@cs.stanford.edu
dkoeplin@stanford.edu

E-mail Addresses (Faculty Advisors)
kunle@stanford.edu
christos@ee.stanford.edu