High-Level Languages for High-Performance Computing

Semyon Grigorev

Saint Petersburg State University

April 14, 2022

High-Level Languages For High-Performance Computing (HLL for HPC)

- Functional, functional-first programming languages for
  - GPGPU programming
  - FPGA programming (program specific processors)
  - Hardware synthesis

High-Level Languages For High-Performance Computing (HLL for HPC)

- Functional, functional-first programming languages for
  - GPGPU programming
  - FPGA programming (program specific processors)
  - Hardware synthesis
- Expressivity, high-level composable primitives
- Type safety, static code checks
- Specific optimizations
  - Fusion (stream fusion)
  - Partial evaluation
  - Deforestation

High-Level Languages For High-Performance Computing (HLL for HPC)

- Functional, functional-first programming languages for
  - GPGPU programming
  - FPGA programming (program specific processors)
  - Hardware synthesis
- Expressivity, high-level composable primitives
- Type safety, static code checks
- Specific optimizations
  - Fusion (stream fusion)
  - Partial evaluation
  - Deforestation
- Specific hardware

HLL for HPC: Projects

- LIFT: high-level functional data parallel language for portable HPC
  - University of Edinburgh, University of Glasgow
  - Supported by HIRP FLAGSHIP
- Haflang: special purpose processor for accelerating functional programming languages
  - Heriot Watt University
  - Supported by Xilinx and QBayLogic
- AnyDSL: A partial evaluation framework for programming high-performance libraries
  - Saarland University, German Research Center for Artificial Intelligence (DFKI)
- Futhark: high-performance purely functional data-parallel array programming
  - University of Copenhagen
- ...

Our projects

- Brahma.FSharp: F# to OpenCL C translator and respective runtime

Our projects

- Brahma.FSharp: F# to OpenCL C translator and respective runtime
- Software-hardware platform for functional programming language
  - Powerful fusion-like optimization (distillation)
  - Special hardware for functional programming language

Brahma.FSharp[1]

- Transparent integration GPGPU computations to .NET applications

[1] https://github.com/YaccConstructor/Brahma.FSharp

Brahma.FSharp[1]

- Transparent integration GPGPU computations to .NET applications
- Runtime translation and compilation F# functions to OpenCL kernels
- Data transfer
  - Primitive types: int, float, bool, ...
  - Structures
  - Discriminated Unions
- Runtime for kernels execution

[1]https://github.com/YaccConstructor/Brahma.FSharp

Software-Hardware Platform for Functional Programming Languages

- Final goal: high-performance sparse linear algebra
- Problems
  - Intermediate data structures $\rightarrow$ memory traffic
  - Sparsity $\rightarrow$ irregular parallelism

Solution (Work in Progress)

- Distillation[2]
  - High-level program technique
  - Includes fusion-like optimization
- Special hardware
  - Reduceron[3]
    - Lambda-processor
    - Migration to Haflang
  - FHW[4]
    - Functional program to hardware translator
    - Program-specific accelerator

[2] https://github.com/YaccConstructor/Distiller
[3] https://github.com/tommythorn/Reduceron
[4] https://github.com/sedwards-lab/fhw

Publications

- Optimizing GPU programs by partial evaluation (PPoPP, Core A)
- Distilling Sparse Linear Algebra (SRC@ICFP)

Preliminary Evaluation: Input

A set of functions for sparse matrices manipulation

- `addMask m1 m2 m3 = mask (mtxAdd m1 m2) m3`
- `kronMask m1 m2 m3 = mask (kron m1 m2) m3`
- `addMap m1 m2 = map f (mtxAdd m1 m2)`
- `kronMap m1 m2 = map f (kron m1 m2)`
- `seqAdd m1 m2 m3 m4 = mtxAdd (mtxAdd (mtxAdd m1 m2) m3) m4`

Preliminary Evaluation: Results

In emulator

Function	Matrix size				Interpreter		Reduceron	FHW
	m1	m2	m3	m4	Red-s	Reads	Ticks	Ticks
seqAdd	64×64	64×64	64×64	64×64	2.7	1.9	1.8	1.4
addMask	64×64	64×64	64×64	–	2.1	1.8	1.4	1.4
kronMask	64×64	2×2	128×128	–	2.2	1.9	1.4	2.7
addMap	64×64	64×64	–	–	2.5	1.7	1.7	1.5
kronMap	64×64	2×2	–	–	2.9	2.2	1.8	2.0

Table: Evaluation results: original program to distilled one ratio of measured metrics

Contact Info

- Lead: Semyon Grigorev
  - PhD (2016), Associate professor (2016, SPbSU)
  - dblp: https://dblp.org/pid/181/9903.html
  - h-index (scopus): 5
  - s.v.grigoriev@spbu.ru
- Collaboration with Daniil Berezun