Workshop on MLIR for HPC

October 21, 2019, Georgia Institute of Technology, Atlanta, GA

Rooms 1123 and 1116, Klaus Advanced Computing Building

Held in conjunction with International Workshop on Languages and Compilers for Parallel Computing (LCPC 2019)

Organizing Committee:

Uday Bondhugula, Indian Institute of Science
Albert Cohen, Google
Tobias Grosser, ETH
Mary Hall, University of Utah
Santosh Pande, Georgia Tech
P. Sadayappan, University of Utah
Vivek Sarkar, Georgia Tech
Michelle Strout, University of Arizona
Reid Tatge, Google

Current State of HPC Compilers

Proprietary

- Robust
- High-quality implementations for supported architectures
- Support HPC community
- Code not performance portable across systems
- Often conservative

Open Source

- Research compilers
  - State-of-the-art
  - Experimental, untrusted
  - Difficult to track language changes
  - Gaps, such as Fortran frontend
- LLVM and gcc
  - Gaps in HPC support
  - Conservative

Current State of HPC Compilers, cont.

Challenges:

- HPC market not large enough to drive significant change to open source or even proprietary compilers
- Meanwhile, research systems not sufficiently robust for production codes

Impact:

- Productivity improvements for HPC not being exploited
- Heterogeneity will make this a bigger concern

Convolutional Neural Network Forward Layer Code (in C)

```
for (n=0; n<N; n++) { // minibatch size
  for (k=0; k<K; k ++) { // output feature map
    for (c=0; c<C; c ++) { // input feature map
      for (p=0; p<P; p ++) { // output height
        ij = p * u; // input height
        for (q =0; q<Q; q ++) { // output width
          ii = q * v; // input width
          for (r=0; r<R; r ++) { // filter height
            for (s =0; s< S; s ++) {// filter width
              output_seq[n][k][p][q] +=
                    input [n][c][ij+r][ii+s] * weight[k][c][r][s];
} } } } } } }
```

Goal: HPC Support in Open Source Compilers

Short-term

(ECP time frame)

Extend LLVM

- Parallel IR
- Loop transformations
- OpenMP/OpenACC
- Autotuning
- Fortran frontend

Potential longer term

(But need to start now)

Collaborate on MLIR

- Higher level of abstraction
- Composability of different views (parallelism?)
- Built-in polyhedral transformations and code generation
- Multiple backends via LLVM
- Missing frontends

Final Program

Time	Title	Speaker
08:45AM	Welcome and Workshop Plan	Mary Hall, University of Utah
09:00AM	LLVM/MLIR in Exascale Computing Project	Hal Finkel, Argonne
09:15AM	MLIR Tutorial	Jacques Pienaar, Google; Sana Damani, Georgia Tech
10:15AM	BREAK	
10:40AM	Loop Nest Optimization, Parallelization and Acceleration in MLIR	Albert Cohen, Google
11:00AM	On Using MLIR to Optimize Tensor Contractions	Rui Li & P. Sadayappan, U. Utah; Atanas Rountev, Ohio State
11:20AM	Compiler-Managed Cache on Commodity Hardware	Chen Ding, University of Rochester
11:40AM	Accelerating Climate Modeling: GPU Mapping for Stencil Graphs	Jean-Michel Gorius, ENS Rennes & ETH
12:00PM	MLIR for Fortran	Vivek Sarkar, Georgia Tech; Nelson Amaral, U. Alberta
12:20PM	LUNCH	
01:30PM	Flash Talks	Student Poster Presenters
01:45PM	Breakout Groups	
03:15PM	BREAK and Student Posters	
03:45PM	Breakout Group Reports	Scribes
04:45PM	Panel Discussion on Breakout Groups	Scribes
05:30PM	Adjourn	

Sign up for breakout session interests, and we will group into four topics.