Abstract

High-performance & ease-of-programming: we want both
• Designing high-performance accelerators demands a great deal of programming effort
• Transforming software programs directly into FPGA circuits becomes feasible via HLS, but for good QoR heavy reconstruction of the software code is often required

From C to high-quality HLS-C: dealing with large design space
• A large number of pragmas at many legal insertion points
• Complex performance-resource trade-offs
• Long evaluation time (tens of minutes per design points)

Automated accelerator generation: automatically producing a high-performance design in a reasonable amount of time
• Machine learning is used to decrease the number of design points that need to be evaluated
• Microarchitecture templates is used to confine the design choices, and, more importantly, enable analytical-model-based fast design space exploration

PolySA: Polyhedral-Based Systolic Array Auto-Compilation

Example code of MM

```
for (int i = 0; i < N; i++)
for (int j = 0; j < N; j++){
C[i][j] = 0;
for (int k = 0; k < N; k++) ... 0
```

SODA: Stencil with Optimized Dataflow Arch.

- Optimized dataflow architecture
- Accurate performance and resource modeling
- Automatic transformation from SODA DSL all the way to bitstreams

AutoDSE: Learning-Based Design Space Exploration Framework

AutoAccel: Automated Accelerator Generation w/ CPP Microarchitecture

- Composable, Parallel, Pipeline (CPP) Microarchitecture
  - Optimized off-chip communication
  - Explicit data caching
  - Parallel & pipeline loop scheduling
  - Scratchpad Reorganization

Pros and Cons of Approaches

Learning-based approach
- Not being constrained to a specific microarchitecture
- Demanding a great deal of time to achieve the optimal solution due to time-consuming design point evaluation

Microarchitecture-based approach
- Coming up with a high-quality design in much less time if the input program fits into the microarchitecture well
- May not work well for all kinds of compute domains

References

Cong et al, Automated Accelerator Generation and Optimization with Composable, Parallel and Pipeline Architecture, DAC ’18
Yu et al, S2FA: An Accelerator Automation Framework for Heterogeneous Computing in Datacenters, DAC ’18
Chi et al, SODA: Stencil with Optimized Dataflow Architecture, ICCAD ’18
Cong et al, PolySA: Polyhedral-Based Systolic Array Auto-Compilation, ICCAD ’18