A Framework for Automatic OpenMP Code Generation

Raghesh A (CS09M032)

Outline

- Introduction
- The Polyhedral Model
- LLVM
- Polly
- OpenMP Code Generation in Polly
- Testing with PolyBench
- Conclusion and Future Work
- Setting up the environment
- Various Tools Used in Polyhedral Community

Introduction

- Parallelism in programs
  - Parallelism and locality
  - Realizing parallelism
- Auto parallelization
- The polyhedral model
- LLVM
- Polly and OpenMP code generation

The Polyhedral Model

- Program transformations with polyhedral model
  - Transformation for improving data locality

```
for(i = 1; i <= 10; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

The Polyhedral Model

- Program transformations with polyhedral model
  - Transformation for improving data locality

```
for(i = 1; i <= 10; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

```
for(i = 1; i <= 5; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

The Polyhedral Model

- Program transformations with polyhedral model
  - Transformation for improving data locality

```
for(i = 1; i <= 10; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

```
for(i = 1; i <= 5; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

  - Scalar expansion

The Polyhedral Model

- Program transformations with polyhedral model
  - Transformation for improving data locality

```
for(i = 1; i <= 10; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

```
for(i = 1; i <= 5; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

  - Scalar expansion

```
for (i = 0; i < 8; i++)
  sum += A[i];
```

The Polyhedral Model

- Program transformations with polyhedral model
  - Transformation for improving data locality

```
for(i = 1; i <= 10; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

```
for(i = 1; i <= 5; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

  - Scalar expansion

```
for (i = 0; i < 8; i++)
  sum += A[i];
```

```
<create and initialize an array 'tmp'>
for (i = 0; i < 8; i++)
   tmp[i % 4] += A[i];
sum = tmp[0] + tmp[1] + tmp[2] + tmp[3];
```

The Polyhedral Model

- Program transformations with polyhedral model
  - Transformation for improving data locality

```
for(i = 1; i <= 10; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

```
for(i = 1; i <= 5; i++)
  A[i] = 10;
for(j = 6; j <= 15; j++)
  A[j] = 15;
```

  - Scalar expansion

```
for (i = 0; i < 8; i++)
  sum += A[i];
```

```
<create and initialize an array 'tmp'>
for (i = 0; i < 8; i++)
   tmp[i % 4] += A[i];
sum = tmp[0] + tmp[1] + tmp[2] + tmp[3];
```

```
parfor (ii = 0; ii < 4; ii++)
  tmp[ii] = 0;
  for (i = ii * 2; i < (ii+1) * 2; i++)
    tmp[ii] += A[i];
sum = tmp[0] + tmp[1] + tmp[2] + tmp[3];
```

- Polyhedral representation of programs

LLVM

- LLVM (Low Level Virtual Machine)
  - Framework for implementing compilers
  - Common low level code repersentation
  - Lifelong analysis and transformation of programs

Polly

- Polly (Polyhedral Optimization in LLVM)
  - Implementing Polyhedral Optimization in LLVM
  - Effort towards Auto Parallelism in programs.
- Implementation
  - LLVM-IR to polyhedral model
    - Region-based SCoP detection
    - Semantic SCoPs
  - Polyhedral model
    - The integer set library
    - Composable polyhedral transformations
    - Export/Import
  - Polyhedral model to LLVM-IR
    - Detecting parallel loops
- Related work

OpenMP Code Generation in Polly

- Introduction
- Code generation pass in Polly
- Detecting parallelism in Polly
- Generating OpenMP library calls
- Support for inner loops
- Dealing with memory references
  - Adding memory references
  - Extracting memory references
- Enabling OpenMP code generation in Polly
- OpenMP testcases

Testing with PolyBench

- PolyBench
- Experimental results

Conclusion and Future Work

- Conclusion
- Support for memory access transformations in Polly
- Increasing coverage of Polly
  - Increasing SCoP coverage
  - Increasing the system coverage
- Integrating profile guided optimization into Polly

Setting up the environment

- CLooG
- PoCC
- Scoplib
- Building LLVM with Polly

Various Tools Used in Polyhedral Community

- ClooG
- PLUTO
- VisualPolylib