Polyhedral Compilation - II

Nikhil Hegde

Compiler Optimizations in LLVM

Lecture series @ QUALCOMM Inc.

Agenda: Demo

- SCoPs
  - SCoPs that don't exist, SCoPs that are and are not profitable,
  - Viewing Polly's activity using `-Rpass-analysis, –Rpass-missed`
  - Highlighting with `-dot-scops-only`
- Representation
  - Polly-scops : creates polyhedral description of SCoPs
- Optimization
  - Optimizing matrix multiplication with polly
- Saving the polyhedral representation in a file (export to jscop)
- Loading the saved representation into polly (import from jscop)
- Code generation based on jscop file loaded
- Target generation and performance comparison

Loop interchange performance

```
[nikhilh@hip codeexamples]$ time ./matmul_polly_interchanged

real    0m20.585s
user    0m20.507s
sys     0m0.042s
[nikhilh@hip codeexamples]$ make matmul_compare
llc matmul.preopt.ll -filetype=obj -o matmul.plain.o -relocation-model=pic
clang matmul.plain.o -o matmul_plain
[nikhilh@hip codeexamples]$ time ./matmul_plain

real    8m43.896s
user    8m43.280s
sys     0m0.044s
[nikhilh@hip codeexamples]$
```

Exercise: Is this correct polyhedral representation for the SCoP?

```
for (i = 0; i <= N; i++) {
    if (i <= N - 50)
S1:     A[5*i] = 1;
    else
S2:     A[3*i] = 2;

    for (j = 0; j <= N; j++)
S3:     B[i][2*j] = 3;
}
```

$$Context = \{[N]\}$$

$$\mathcal{D}_{S1} = \{S1[i] : i \geq 0 \wedge i \leq N \wedge i \leq N - 50\}$$
$$\mathcal{S}_{S1} = \{S1[i] \rightarrow [0, i, 0, 0, 0]\}$$
$$\mathcal{A}_{S1} = \{S1[i] \rightarrow A[5i]\}$$

$$\mathcal{D}_{S2} = \{S2[i] : i \geq 0 \wedge i \leq N \wedge i > N - 50\}$$
$$\mathcal{S}_{S2} = \{S2[i] \rightarrow [0, i, 1, 0, 0]\}$$
$$\mathcal{A}_{S2} = \{S2[i] \rightarrow A[3i]\}$$

$$\mathcal{D}_{S3} = \{S3[i, j] : i \geq 0 \wedge i \leq N \wedge j \geq 0 \wedge j \leq N\}$$
$$\mathcal{S}_{S3} = \{S3[i, j] \rightarrow [0, i, 2, j, 0]\}$$
$$\mathcal{A}_{S3} = \{S3[i, j] \rightarrow B[i][2j]\}$$