Relation-based Polyhedral Optimization

Louis-Noël Pouchet

pouchet@cse.ohio-state.edu

Dept. of Computer Science and Engineering, the Ohio State University

February 2012

888.11

Matrix-multiply

Example

```
for (i = 0; i < n; ++i)
  for (j = 0; j < n; ++j) {
     C[i][j] = 0;
     for (k = 0; k < n; ++k)
        C[i][j] += A[i][k] * B[k][j];
  }
```

Matrix-multiply: ISCC

Domain:

Example

```
s:= [n]->{ s1[i,j] : 0 <= i,j < n ; s2[i,j,k] : 0 <= i,j,k < n};
```

Scattering:

Example

```
trans:= {s1[i,j] -> [i,j,0,0]; s2[i,j,k] -> [i,j,1,k]};
```

Matrix-multiply: ISCC codegen

codegen:

Example

```
s:= [n]->{ s1[i,j] : 0 <= i,j < n ; s2[i,j,k] : 0 <= i,j,k < n};
origsched:= {s1[i,j] -> [i,j,0,0]; s2[i,j,k] -> [i,j,1,k]};
codegen(origsched * s);

if (n >= 1) {
  for (c1=0;c1<=n-1;c1++) {
    for (c2=0;c2<=n-1;c2++) {
      s1(c1,c2);
      for (c4=0;c4<=n-1;c4++) {
        s2(c1,c2,c4);
      }
    }
  }
}
```

Comments on Scatterings

- Names (namespace) should match the names used in the domain
- Multiple statements in the domain require a scattering for codegen
- (sub-)domains can be of various input dimensions
- output dimensions of all scattering functions must be the same

- Good practice (but not required): use 2d+1 encoding

A Word on ISCC syntax

- `;` represents disjunction
- `*` represents intersection (in the above case, intersect a map with a domain)
- `a(b)` represents the image of `b` under `a`, or the composition of functions

Interchange (naive)

Example

```
theta := {s1[i,j] -> [j,i,0,0]; s2[i,j,k] -> [j,i,1,k]};
codegen (theta * s);

if (n >= 1) {
  for (c1=0;c1<=n-1;c1++) {
    for (c2=0;c2<=n-1;c2++) {
      s1(c2,c1);
      for (c4=0;c4<=n-1;c4++) {
        s2(c2,c1,c4);
      }
    }
  }
}
```

Interchange (via composition)

Example

```
inter3d := {[i,j,b,k] -> [j,i,b,k]};
theta := inter3d(origsched);
codegen (theta * s);

if (n >= 1) {
  for (c1=0;c1<=n-1;c1++) {
    for (c2=0;c2<=n-1;c2++) {
      s1(c2,c1);
      for (c4=0;c4<=n-1;c4++) {
        s2(c2,c1,c4);
      }
    }
  }
}
```

Strip-mine

Example

```
stripfirst32:= {[i,j,b,k] -> [it,ii,j,b,k] : i = 32 it + ii and 0 <= ii < 32};
theta := stripfirst32(origsched);
codegen (theta * s);

if (n >= 1) {
  for (c1=0;c1<=floord(n-1,32);c1++) {
    for (c2=0;c2<=min(31,-32*c1+n-1);c2++) {
      for (c3=0;c3<=n-1;c3++) {
        s1(32*c1+c2,c3);
        for (c5=0;c5<=n-1;c5++) {
          s2(32*c1+c2,c3,c5);
        }
      }
    }
  }
}
```

Strip-mine (other way)

Example

```
stripfirst32:= [i,j,b,k] -> [it,i,j,b,k] : it mod 32 = 0 and it <= i < it + 32;
theta := stripfirst32(origsched);
codegen (theta * s);
if (n >= 1) {
  for (c1=0;c1<=n-1;c1+=32) {
    for (c2=c1;c2<=min(c1+31,n-1);c2++) {
      for (c3=0;c3<=n-1;c3++) {
        s1(c2,c3);
        for (c5=0;c5<=n-1;c5++) {
          s2(c2,c3,c5);
        }
      }
    }
  }
}
```

Tiling

Tiling is stripmine+stripmine+interchange

Example

```
stfirst:= {[i,j,b,k] -> [it,i,j,b,k] : it mod 32 = 0 and it <= i < it + 32};
stsecond:= {[it,i,j,b,k] -> [it,i,jt,j,b,k] : jt mod 32 = 0 and jt <= j < jt + 32};
intertiling:= {[it,i,jt,j,b,k] -> [it,jt,i,j,b,k] };
theta := intertiling(stsecond(stfirst(origsched)));
codegen (theta * s);
if (n >= 1) {
  for (c1=0;c1<=n-1;c1+=32) {
    for (c2=0;c2<=n-1;c2+=32) {
      for (c3=c1;c3<=min(c1+31,n-1);c3++) {
        for (c4=c2;c4<=min(c2+31,n-1);c4++) {
          s2(c3,c4,0);
          s1(c3,c4);
          for (c6=1;c6<=n-1;c6++) {
            s2(c3,c4,c6);
          }
        }
      }
    }
  }
```

Jacobi 1D

Example

```
domj1d:= [N,T]->{ s1[t,i] : 0 <= i < N and 0 < t < T; s2[t,i] : 0 <= i < N and 0 < t < T};
origsj1d := { s1[t,i] -> [t,0,i]; s2[t,i] -> [t,1,i]};
codegen (origsj1d * domj1d);
if ((N >= 1) && (T >= 2)) {
  for (c1=1;c1<=T-1;c1++) {
    for (c3=0;c3<=N-1;c3++) {
      s1(c1,c3);
    }
    for (c3=0;c3<=N-1;c3++) {
      s2(c1,c3);
    }
  }
}
```

Skewing

Example

```
skewj1d := { [t,b,i] -> [t,b,s] : s = 2t + i};
codegen (skewj1d(origsj1d) * domj1d);
if ((N >= 1) && (T >= 2)) {
  for (c1=1;c1<=T-1;c1++) {
    for (c3=2*c1;c3<=2*c1+N-1;c3++) {
      s1(c1,-2*c1+c3);
    }
    for (c3=2*c1;c3<=2*c1+N-1;c3++) {
      s2(c1,-2*c1+c3);
    }
  }
```

Tiling

Example

```
stripfirst32:= {[i,j,k] -> [it,i,j,k] : it mod 32 = 0 and it <= i < it + 32};
codegen (stripfirst32(skewj1d(origsj1d)) * domj1d);
if ((N >= 1) && (T >= 2)) {
  for (c1=0;c1<=T-1;c1+=32) {
    for (c2=max(1,c1);c2<=min(T-1,c1+31);c2++) {
      for (c4=2*c2;c4<=2*c2+N-1;c4++) {
        s1(c2,-2*c2+c4);
      }
      for (c4=2*c2;c4<=2*c2+N-1;c4++) {
        s2(c2,-2*c2+c4);
      }
    }
  }
}
```