Target: We need an affine function, specifying for each integral point in the iteration domain one or more points in memory space.

When give a separation of iteration domain, we need to calculate the separation of the memory space for each sub iteration domains.

Also we need evaluate the overlap of sub memory spaces.

We need find a separation which can minimum the overlap and the sub memory spaces should be be near the same,

Example:

The body of iteration domain include two memory access statements which are surrounded with an small loop.

The iteration domain is D = {i | 0 <= i <n}

The affine function is f(i) = A\*i +B, A is a constant, B is integral and b1<B<b2.

Another affine function is g(i) = A\*i +C, C is integral and c1 <C<c2.

Compare with Polyhedral Model:

1. The concept of iteration domain is the same.

2. We need chose the scattering function in Polyhedral Model, but is decided by the code in our methods.

3. In Polyhedral Model, program transformations are specified by scattering functions. In our methods, tasks and data partition are specified by a split affine hyperplane which split iteration domain.

4. In our methods, the dimension of iteration domain is not more than 3, because OpenCL and CUDA not support more than 3 dimensions index space, and hStreams only support 1 dimensions index space.

5. The scattering function in our methods can be one-to-many mapping, and its returned value is 1 dimension memory address.

6. The scattering function in Polyhedral Model can not support non-constant variables. In our methods, the scattering function contains symbolic variables.

7. In Polyhedral Model only has one scattering function, but in our method, there are more than one scattering functions. Of cause we can increase dimension to reduce the number of scattering functions.

8. Polyhedral Model split iteration domain using affine conditions, as in index set splitting.

9. The dependence model in Polyhedral Model between tow statements is characterized by a polyhedron that is in the sum of the dimensionalities of the source and target statement’s polyhedra.

10. In Polyhedral Model, it assumes that the index of array is just the iteration variables.

11. Compare with Polyhedral Model, we focus on one-dimensional affine transform, and the distance between hyperlanes, not just consider the validation to dependence.

12. The parallel execution order in Polyhedral Model may be too complex and can not be represented by a compact program. In our methods, the memory access pattern may be complex and can not suite to the 1 dimension memory transform.

Compare with Symbolic Range Analysis:

1. The Symbolic Range Analysis works on IR, our method works on AST.
2. The Symbolic Range Analysis evaluates the value range for all variables, our method only evaluates the range of specific variables.
3. Our method use one pass backward analysis?.

Compare with “Dataflow Analysis of Array and Scalar References”:

Why implement the polyhedra model analysis yourself? We want work on AST and can generate readable code, otherwise like Dawncc, it pay cost to make the code generated readable.