CS5214 Design of Optimizing Compilers

Professor Weng-Fai Wong
Assignment One

April 18, 2016

Abstract

This is my solutions for assignment three problem set. This is the answer presented by Pan An. My student number is A0134556A.

1 Loop Analysis

The flow dependency of the given loop psudo-code concerns only variable list X. Here we denote the only value assignment in the loop as

$$X[\vec{f}(i,j,k)] = X[\vec{g}(i,j,k)]$$

1.1 Formulation

So in this case if we need to take care of the formulation there are three equations that we need to satisfy:

$$\begin{aligned} 3j_s + k_s = &\ 2i_t + 4k_t \\ i_s + 2j_s + k_s = &\ 5j_t + 7k_t \\ 2j_s + 3k_s = &\ 3i_t + 5j_t \end{aligned} \tag{1}$$

And i, j, k subject to:

$$\begin{aligned} 0 < i_s, i_t < 11 \\ 0 < j_s, j_t < 51 \\ 0 < k_s, k_t < 21 \end{aligned} \tag{2}$$

So here I'm gonna go with formulating the problem with distance vector. The problem defined can be expressed in matrix form:

$$\boldsymbol{V} \cdot \boldsymbol{A} < \boldsymbol{b}$$

where:

$$\boldsymbol{V} = \begin{bmatrix} 0 & 1 & 0 \\ 3 & 2 & 2 \\ 1 & 1 & 3 \\ -2 & 0 & -3 \\ 0 & -5 & -5 \\ -4 & -7 & 0 \end{bmatrix} \tag{3}$$

Which has to satisfy:

$$[i_s, j_s, k_s, i_t, j_t, k_t] \begin{bmatrix} 0 & 1 & 0 \\ 3 & 2 & 2 \\ 1 & 1 & 3 \\ -2 & 0 & -3 \\ 0 & -5 & -5 \\ -4 & -7 & 0 \end{bmatrix} = [0, 0, 0] \tag{4}$$

And after Echelon Reduction we have:

$$\boldsymbol{E} = \left\{ \begin{matrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{matrix} \right\} \quad and \quad \boldsymbol{U} = \begin{bmatrix} -\frac{4}{7} & \frac{3}{7} & -\frac{2}{7} & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ -\frac{1}{7} & -\frac{1}{7} & \frac{3}{7} & 0 & 0 & 0 \\ -\frac{17}{21} & \frac{1}{3} & 0 & \frac{12}{7} & 0 & 0 \\ \frac{30}{7} & -\frac{5}{7} & \frac{15}{7} & 0 & 1 & 0 \\ \frac{59}{21} & \frac{12}{7} & -\frac{8}{7} & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

And in order to do finish the back substitution have to let $\boldsymbol{t} \cdot \boldsymbol{E} = \mathbf{0}$. Next we will be conducting Fourier-Motzkin Elimination.

1.2 Fourier-Motzkin Elimination

Okay so when I finish the last chapter I suddenly relize that what I was required to do is to give the process of Fourier-Motzkin Elimination. Anyway I just leave the stuff there.

And according the the previous result. $\boldsymbol{t}$ must satisfy:

$$\begin{aligned} 0 < & -\frac{17}{21}t_4 + \frac{30}{7}t_5 + \frac{59}{21}t_6 && < 11 \\ 0 < & \quad \frac{1}{3}t_4 - \frac{5}{7} + \frac{12}{7} && < 51 \\ 0 < & \quad \frac{15}{7}t_5 - \frac{8}{7} && < 21 \\ 0 < & \quad \frac{12}{7}t_4 && < 11 \\ 0 < & \quad t_5 && < 51 \\ 0 < & \quad t_6 && < 21 \end{aligned} \tag{6}$$

There is actually solution to this one.

1.3 Omega Test

2 Loop Tiling

Considering that in this problem X and Y has exactly the same size. There is no need to take into consideration the utilization of loop interchange.

Anyway, the idea of loop tiling is to increase cache. Consider the given case:

```
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) {
        X[i, j] = Y[j, i];
    }
}
```

This however, will result in loading the data in X and Y unnecessary number of times. The program will miss every access to one of the array and every step of C on the other, where C is the size of the cache. And the total cache miss would be:

$$N^2 + \frac{N^2}{C}$$

And in this case it's $N^2 + \frac{N^2}{8}$.

Loop tiling code is shown as below(notice that both X and Y are $N \times N$ matrix):

```
for(ii=0; ii<N; ii+=block_size)
    for(jj=0; jj<N; jj+=block_size)
        for(i=ii; i<min(ii+block_size); i++)
            for(j=jj; j<min(jj+block_size); j++)
                X[i, j] = Y[j, i]
```

The loop will be excuted as the following graph shows:

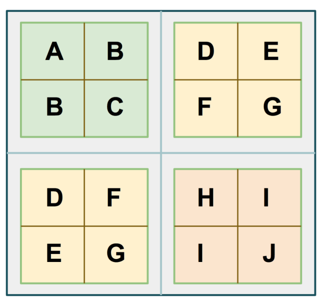

Figure 1: Simple Graph of Tiling.

In 1 the input means the matrix of y and the output is the matrix of x. Each block represents a 2×2 block.

The cache hit, before the tiling, should be 50%. Considering that only 8 digits were in the cache. And after the tiling it should be 100%, since it only loads the piece of matrix that is needed to do the transpose.