Annie Wang

305-532-153

Professor Reinman

CS 33

Homework 4

Problem 6.45: In this assignment, you will apply the concepts you learned in Chapters 5 and 6 to the problem of optimizing code for a memory-intensive application. Consider a procedure to copy and transpose the elements of an N °ø N matrix of type int. That is, for source matrix S and destination matrix D, we want to copy each element si,j to dj,i . This code can be written with a simple loop,

1 void transpose(int \*dst, int \*src, int dim)

2 {

3 int i, j;

4

5 for (i = 0; i < dim; i++)

6 for (j = 0; j < dim; j++)

7 dst[j\*dim + i] = src[i\*dim + j];

8 }

where the arguments to the procedure are pointers to the destination (dst) and source (src) matrices, as well as the matrix size N (dim). Your job is to devise a transpose routine that runs as fast as possible.

void transpose(int \*dst, int \*src, int dim)

{

int i, j;

int bound = dim-8;

for (i = 0; i < bound; i+=8)

{

int zero = dim \* i;

int first = dim \* (i + 1);

int second = dim (i + 2);

int third = dim \* (i + 3);

int fourth = dim (i + 4);

int fifth = dim \* (i + 5);

int sixth = dim (i + 6);

int seventh = dim \* (i + 7);

for (j = 0; j < dim; j++)

{

int temp = j \* dim;

dst[temp + i] = src[zero + j];

dst[temp + i + 1] = src[first + j];

dst[temp + i + 2] = src[second + j];

dst[temp + i + 3] = src[third + j];

dst[temp + i + 4] = src[fourth + j];

dst[temp + i + 5] = src[fifth + j];

dst[temp + i + 6] = src[sixth + j];

dst[temp + i + 7] = src[seventh + j];

}

}

for (; i < dim; i++)

{

for (j = 0; j < dim; j++)

{

dst[j \* dim + i] = src[i \* dim + j];

}

}

}