HY425

Homework Problem Set 4

Assignment: 14/5/2008

Due Date: 31/5/2008

Instructions: Solve all problems in a **.pdf** file and send them via e-mail to Stamatis Kavadias (kavadias@ics.forth.gr), with a copy to the instructor (dsn@ics.forth.gr). Use the following subject in your e-mail: **HY425: Homework 4 Submission**. Please the aforementioned subject only, so that your homework is read and graded.

Problem 1 (48 points)

The transpose of a matrix interchanges its rows and columns and is illustrated below:

$$\begin{bmatrix} A11 & A12 & A13 & A14 \\ A21 & A22 & A23 & A24 \\ A31 & A32 & A33 & A34 \\ A41 & A42 & A43 & A44 \end{bmatrix} \Rightarrow \begin{bmatrix} A11 & A21 & A31 & A41 \\ A12 & A22 & A32 & A42 \\ A13 & A23 & A33 & A43 \\ A14 & A24 & A34 & A44 \end{bmatrix}$$

Here is a C loop to show the transpose.

```
for (i=0;i<3;i++) {
   for (j=0;j<3;j++) {
     output[j][i] = input[i][j];
   }
}
```

Assume both the input and the output matrices are stored in the row major order (row major order means row index changes faster). Assume you are executing a 256 × 256 double-precision transpose on a processor with 16 KB fully associative (so you don't have to worry about cache conflicts) LRU replacement level 1 data cache with 32-byte blocks. Assume level 1 cache misses or prefetches require 8 cycles, always hit in the level 2 cache, and the level 2 cache can process a request every 2 processor cycles. Assume each iteration of the inner loop above requires 4 cycles if the data is present in the level 1 cache. Assume the cache has a write-allocate fetch-on-write policy for write misses. Unrealistically, assume writing back dirty cache blocks requires 0 cycles.

a. For the simple implementation given above, the execution order would be not ideal for the input matrix. However, applying a loop interchange optimization would create a not ideal order for the output matrix. Because loop interchange is not sufficient to improve its performance, it must be blocked instead. What block size should be used to completely fill the data cache with one input and one output block? [12 points]
b. How do the relative number of misses of the blocked and the unblocked versions compare if the level 1 cache is direct mapped? [18 points]

c. Write code to perform a transpose with a block size parameter B that uses B × B blocks. [18 points]

Problem 2 (14 points)

Assume you are redesigning a hardware prefetcher for the unblocked matrix transposition code above. The simplest type of hardware prefetcher only prefetches sequential cache blocks after a miss. More complicated "non unit stride" hardware prefetchers can analyze a miss reference stream, and detect and prefetch non unit strides. In contrast, software prefetching can determine non unit strides as easily as it can determine unit strides. Assume prefetches write directly into the cache and no pollution (overwriting data that needs to be used before the data that is prefetched). In the steady state of the inner loop, what is the performance (in cycles per iteration) when using an ideal non unit stride prefetcher? [14 points]

Problem 3 (38 points)

Assume you are redesigning a hardware prefetcher for the unblocked matrix transposition code as in the previous problem. However, in this case we evaluate a simple two-stream sequential prefetcher. If there are level 2 access slots available, this prefetcher will fetch up to 4 sequential blocks after a miss and place them in a stream buffer. Stream buffers that have empty slots obtain access to the level 2 cache on a round-robin basis. On a level 1 miss, the stream buffer that has least recently supplied data on a miss is flushed and reused for the new miss stream.

a. In the steady state of the inner loop, what is the performance (in cycles per iteration) when using a simple two-stream sequential prefetcher assuming performance is limited by prefetching? [19 points]
b. What percentage of prefetches are useful given the level 2 cache parameters? [19 points]