**High Performance Computing**

2nd Midterm - December 18, 2018

Write your name, surname, student identification number (numero di matricola), e-mail. The answers can be written in English or in Italian. Please, present the work in a legible and readable form. All the answers must be properly and clearly explained, focusing on the requested issues.

1. A system Σ at the processes level is composed of four modules as shown in the figure below. Processes $P_1$ and $P_2$ generate a stream of $L = 10^5$ arrays of size $M = 8K$ integers by reading two large in-memory matrices $Z_1[L][M]$ and $Z_2[L][M]$ (each stream element is a row of the matrix read by the corresponding process). Each message generated by the process $P_2$ is transmitted either to $P_3$ or to $P_4$ with the probabilities in the figure. The system is executed on the architecture with the specifications at the end of this text. Process $P_3$ executes the following computation on each message $A$ received from one of its input channels:

```
P_3 :: \{ \ldots s = 0; \text{for } i=0 \text{ to } M-1 \text{ do } s = s + A[i]; \text{send}(ch\textunderscore out1, s); \ldots \}
```

Process $P_4$ executes the following code on each input message $A$ received from $P_2$ and using an internal (statically initialized) matrix $B[M^2]$:

```
P_4 :: \{ \ldots y=0; 

\text{for } i=0 \text{ to } M-1 \text{ do } 

\text{for } j = 0 \text{ to } M-1 \text{ do } 

B[i][j] = B[i][j] * A[i]; 

y = y + B[i][j];

\text{send}(ch\textunderscore out2, y);

\}
```

Give proper answers, by providing detailed explanations, to the following points:

a) Study the system as it is by finding the ideal service times and the effective ones of each process. For the evaluation, assume that the home node of each channel is always the destination PE, while all the cache blocks of $Z_1/Z_2$ have home node in the PE of the process $P_1/P_2$;

b) Show the firmware messages exchanged between all the caches of the PEs executing the four processes in the figure, identify the most stressed secondary cache and show the values of the under-load latency cost model ($R_{Q-o}, p, T_S, T_p$).

2. In a shared-memory architecture with snoopy-based cache coherence, a process $P$ accesses a cache block $b$. Show the MESI diagram of the possible state transitions both in case of a read and a write memory access to the cache line performed by $P$. Discuss the semantics of those transitions by explaining in detail the goal of each specific MESI state.

**Architectural specifications for Exercise 1**

- Symmetric multi-processor architecture composed of 64 CMPs, each with 4 Processing Elements (PEs) connected to an on-chip MINF unit through a 1x4 crossbar;
- External SMP network based on a butterfly with wormhole flow control, flits of 32 bits and link transmission latency of $T_{fr} = \tau$. Memory macro-modules mutually interleaved with $T_M = 50\tau$;
- Each PE has a primary data cache of 4K and a secondary cache of 128K data words (of 32 bits). Each PE has a dedicated communication processor/thread. Block size $\sigma = 8$ words (32 bits each);
- Each PE has a D-RISC pipelined processor with average service time per instruction of $3/2\tau$;
- Directory-based cache coherence with home flush semantics;
- Inter-process communication run-time support using the RDY-ACK implementation with shared synchronization variables. Each channel is assumed to have asynchrony degree equal to one;
- Assume a random distribution of the processes onto the available CMPs (for simplicity, assume that only one process can be mapped onto the same CMP).
Solution Overview

1. We start with the analysis of the first process. Its role is to read each row of the matrix which is store in the memory (each block is uniformly distributed among the existing macro-modules) and copy it in the corresponding target variable of the channel. Since the home node is the destination PE, each synchronous STORE on the channel/VTG generate a store notification message with the home-flush semantics. The ideal service time is:

\[ T_{id-p1} = T_{send}(M) \sim T_{copy}(M) \]

\[ T_{copy}(M) = \frac{M}{\sigma} (L_{read-c1}(\sigma) + L_{write-c2c2}(\sigma)) + M \cdot \beta_{code} \]

Read latency from the memory: \( C1-C2-W-CROSSBAR-MINF-WW-SMP\_NETWORK-Im-M \) equal to \( d = 8 + d_{net} \). All the paths in the butterfly network have distance \( d_{net} = 6 \). Thus:

\[ L_{read-c1}(\sigma) = (3 + 14 - 2) \cdot T_{hop} + \tau_{M} + (9 + 14 - 2) \cdot T_{hop} = 30\tau + 50\tau + 42\tau = 122\tau \]

Flush latency from C1 (source) to C2 (dest): \( C1-C2-W-CROSSBAR-MINF-WW-SMP\_NETWORK-WW-MINF-CROSSBAR-W-C2 \) equal to \( d = 11 + d_{net} \). The SMP network is now used as a fat tree with height equal to \( h = 6 \). According to the general model, \( d_{net} \sim 1.9 \cdot h \sim 12 \). The latency is:

\[ L_{write-c2c2}(\sigma) = (11 + 23 - 2) \cdot T_{hop} + (1 + 23 - 2) \cdot T_{hop} = 64\tau + 44\tau = 108\tau \]

The ideal service time is (equal for the second process):

\[ T_{id-p1} = T_{id-p2} = 34.75M\tau \]

The inter-arrival time to the third process is:

\[ T_{A-p3} = \frac{1}{34.75M\tau} + \frac{1}{104.25M\tau} \sim 26M\tau \]

The ideal service time of the third process is dominated by the instruction in the for loop:

LOOP: LOAD RA-addr, Ri, Ra
ADD RS, Ra, RS
INCR Ri
IF < Ri, RM LOOP

Where, for each cache line of the array \( A \), we pay a miss with the transfer latency from C2. Therefore:

\[ T_{calc-p3} = M(4 \cdot T_{instr}) + \frac{M}{\sigma} L_{c2-c1}(\sigma) = 7.25M\tau \]

Since the output message is a scalar, the calculation time is also the ideal service time of the process which is not a bottleneck.

The fourth process has inter-arrival time:

\[ T_{A-p4} = \frac{34.75M\tau}{2/3} \sim 52.13M\tau \]

In the computation of the process we have reuse on the blocks of A (which we find in the C2), and we pay locality in the block of the matrix B whose size is \( M^2 = 64 \) MB. The instructions in the innermost loop:

LOOP_J: LOAD RB-index, Rj, Rb
MUL Rb, Ra, Rb
STORE RB-index, Rj, Rb
ADD Ry, Rb, Ry
INCR Rj
IF < Rj, RM LOOP_J

The calculation time is:
The cache misses for the cache blocks of the input target variable (from C2 to C1) are negligible. Furthermore, also in this case the ideal service time is by far dominated by the calculation time of the process, which is a large bottleneck. The following table summarizes the result of the first point a):

<table>
<thead>
<tr>
<th>Process</th>
<th>Ideal TS</th>
<th>Effective TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>$34.75M\tau$</td>
<td>$34.75M\tau$</td>
</tr>
<tr>
<td>P2</td>
<td>$34.75M\tau$</td>
<td>$16.17M^2\tau$</td>
</tr>
<tr>
<td>P3</td>
<td>$7.25M\tau$</td>
<td>$\sim 34.75M\tau$</td>
</tr>
<tr>
<td>P4</td>
<td>$24.25M^2\tau$</td>
<td>$24.25M^2\tau$</td>
</tr>
</tbody>
</table>

To answer the second point b), we show the messages exchanged between caches in the figure below:

It is easy to recognize that the most stressed cache is without any doubt the secondary cache of the fourth process, because although its contention parameter is $p = 2$ as the secondary cache of the third process, it receives much more firmware messages (most of them from its primary cache). Since the majority of the requests come from the primary cache, the parameters of the under-load latency are quite easy to determine:

- $p = 2$
- $T_s = \sigma\tau = 8\tau$
- $R_{Q-o} = L_{c2-c1}(\sigma) = 10\tau$

The grain-size parameter can be easily determined since we count only the requests from the primary cache:

$$C_{c1} \sim 2 \cdot \frac{M^2}{\sigma}$$

$$T_p = \frac{1}{2 \cdot \frac{M^2}{\sigma}} = 97\tau$$

$$\frac{1}{24.25\tau}$$
2. The answer to the second exercise can be found in the block of slides titled “Cache Coherence” and, more precisely, with the state transition diagram presented at slide number 36. Further details for the semantics of the block states were given during that lecture and can be found in the same slide block.