2 Memory Scheduling [70 points]

In lectures, we introduced a variety of ways to tackle memory interference. In this problem, we will look at the Blacklisting Memory Scheduler (BLISS) to reduce unfairness. There are two key aspects of BLISS that you need to know.

- When the memory controller services η consecutive requests from a particular application, this application is blacklisted. We name this non-negative integer η the **Blacklisting Threshold**.
- The blacklist is cleared periodically every **10000** cycles starting at $t = 0$.

To reduce unfairness, memory requests in BLISS are prioritized in the following order:

- Non-blacklisted applications' requests
- Row buffer hit requests
- Older requests

The memory system for this problem consists of 2 channels with 2 banks each. Tables 1 and 2 show the memory request stream in the same bank for both applications at different times ($t = 0$ and $t = 10$). For both tables, a request on the left-hand side is older than a request on the right-hand side in the same table. The applications do not generate more requests than those shown in Tables 1 and 2. The memory requests are labeled with numbers that represent the row position of the data within the accessed bank. Assume the following for all questions:

- A row buffer *hit* takes **100 cycles**.
- A row buffer *miss* (i.e., opening a row in a bank with a closed row buffer) takes **200 cycles**.
- A row buffer *conflict* (i.e., closing the currently open row and opening another one) takes **250 cycles**.
- All row buffers are closed at time $t = 0$

Application A (Channel 0, Bank 0)								
Application B (Channel 0, Bank 0)	Row 2	Row 2	Row 2	Row 2	Row 2	Row 3	Row 3	Row 4

Table 1: Memory requests of the two applications at $t = 0$

Application A (Channel 0, Bank 0)	Row 3	Row 7	Row 2	Row 0	Row 5	Row 3	Row 8	Row 9
Application B (Channel 0, Bank 0)	Row 2	Row 2	Row 2	Row 2	Row 2	Row 3	Row 3	Row 4

Table 2: Memory requests of the two applications at $t = 10$. Note that none of the Application B's existing requests are serviced yet.

(a) [15 points] Compute the slowdown of each application using the FR-FCFS scheduling policy after both threads ran to completion. We define:

$$slowdown = \frac{memory\ latency\ of\ the\ application\ when\ run\ together\ with\ other\ applications}{memory\ latency\ of\ the\ application\ when\ run\ alone}$$

Show your work.

$slowdown_A =\sim 1.53$
$slowdown_B = 1.25$

Explanation:
For both applications, the first request will incur row buffer miss penalty, and the rest of the requests will either be hits or conflicts.
Application A (alone) $= 200 + 100 + 250 * 6 = 1800$ *cycles*
Application B (alone) $= 200 + 100 * 4 + 250 + 100 + 250 = 1200$ *cycles*

Applications A (with B, FR-FCFS) $= 200+100*4+100+250+100+100*2+250+250*5 =$ 2750 *cycles*
Applications B (with A, FR-FCFS) $= 200 + 100 * 4 + 100 + 250 + 100 + 100 * 2 + 250 =$ 1500 *cycles*

From the two tables above we know that all requests of application B were issued before any of the application A's requests were issued. Thus, all requests of B are prioritized unless there is a row hit for A's requests.
$slowdown_A = \frac{2750}{1800} =\sim 1.53$

$slowdown_B = \frac{1500}{1200} = 1.25$

(b) [15 points] If we use the BLISS scheduler, for what value(s) of η (the Blacklisting Threshold) will the slowdowns of **both** applications be equal to those obtained with FR-FCFS?

For $\eta \geq 6$ or $\eta = 0$.

Explanation:
We want both A and B to complete without blacklisting or to complete both blacklisted, thus $\eta \geq 6$ and $\eta = 0$, respectively.

(c) [15 points] For what value(s) of η (the Blacklisting Threshold) will the slowdown of A be < 1.5?

Impossible. Slowdown for A will always be ≥ 1.5

Explanation: For the give memory requests, it is not possible to find η that blacklists B but not A. Thus, the smallest slowdown for A is the case explained in the solution of part (b).

(d) [15 points] For what value(s) of η (the Blacklisting Threshold) will B experience the maximum slowdown it can possibly experience with the Blacklisting Scheduler?

For $\eta = 5$.

Explanation: We already know that the slowdowns will be equal to the slowdown with FR-FCFS when $\eta \geq 6$ or $\eta = 0$. If we execute the memory requests for the rest of possible η values, we find that $\eta = 5$ causes application B to complete after 2150 cycles, which is the largest.

(e) [10 points] What is a simple mechanism (that we discussed in lectures) that we can use instead of BLISS to make the slowdowns of both A and B equal to 1.00?

Memory Channel Partitioning (MCP)

Explanation: With MCP, each application will operate on an independent channel, without any interference with the other application.