CS2610 - Computer Organization and Architecture

DRAM-Buffer Management System

Hemesh D J (CS20B031) & Bersilin C (CS20B013)

Objectives

- Implement a **Hybrid-Row Buffer Management Policy**, known as an **Adaptive Policy** which uses both **Open-Page** and **Closed-Page** Management policy.
- Compare the three different policies and determine which one is the best.
- Compare the DRAM performance for **two address mapping schemes** for the best policy.

Method

The main process of shifting between Open Policy and Closed Policy (in the adaptive-page policy) is done with the help of a saturating 4-bit counter. We have stored the **previously accessed row** in each bank in each of the ranks in the DRAM simulated by the USIMM simulator. We have compared these values with the row accessed by the current request taken from the read or write queue. We have incremented or decremented the counter value based on the above comparison and when the counter crosses the threshold, we flip the policy. We have followed the algorithm given in the problem statement to decide the counter movement and flipping of policy to use. We have used the **higher threshold as 12 and lower threshold as 7** and the initial value of the **counter as 10**.

We then used the total cycles, average-write-queue-latency, average-read-queue-latency and the execution times for comparison between the 3 policies calculated by the **USIMM** simulator.

We used the **USIMM** simulator standard templates for closed and open policies for implementing the Adaptive policy.

The following command was used to run the simulator

"bin/usimm input/4channel.cfg input/libq input/libq input/libq input/libq input/libq input/libq input/libq input/libq input/libq > output/libw"

, where libq is the trace to be tested, and libw is the output file. The command uses 4-channel configuration and 4Gb-x8 devices for the simulation.

Observations

The following table shows the number of cycles, read latency, write latency and the execution time of various input traces during simulation for the three different page policies:

Input	Open-Page policy			
	Cycles	Read-Latency	Write-Latency	Execution Time
comm1	353400508	229.12357	3270.82317	2053982697
comm2	508192909	319.32932	2499.96374	4058241334
black	258873284	272.30957	2499.96374	2814800565
fluid	350620065	289.73389	2220.18759	2779656570
freq	220884669	267.31074	2367.19713	1754010832
ferret	360672549	409.71111	2404.39613	2883003137
face	342860808	564.06197	1631.30905	2742184129
stream	259307220	275.11834	2381.17843	2057241761
swapt	383739573	261.45022	2300.24206	3014524210

Input	Adaptive-Page Policy			
	Cycles	Read-Latency	Write-Latency	Execution Time
comm1	331883492	201.57495	2835.0818	2643319509
comm2	478008741	287.83254	2289.96725	3815593814
black	247947176	242.7667	2110.63572	1961693221
fluid	333520093	260.04417	1990.3223	2641677497
freq	210978241	235.55733	2085.67482	1668274626
ferret	341331201	378.70829	2209.11407	2726841170
face	326565056	544.45838	1520.52107	2611844385
stream	247238800	247.46276	2124.34417	1965893669
swapt	367347349	232.85375	2024.99204	2883507606

Input	Closed-Page Policy			
	Cycles	Read-Latency	Write-Latency	Execuation Time
comm1	331653892	202.00012	2740.79704	2644965829
comm2	477856581	288.06129	2289.21022	3815128402
black	247395452	240.19647	2107.10373	1954067010
fluid	335585913	265.6545	1991.79149	2659901477
freq	210921681	235.279	2073.25443	1667070665
ferret	341017941	378.49501	2196.60147	2725417687
face	326449792	543.05073	1520.57116	2610859121
stream	246607112	246.00066	2119.08407	1958013957
swapt	364162885	230.61289	2001.3246	2843444422

Only the trace “fluid” showed least cycles in Adaptive Policy. Closed-Page Policy showed least cycles for the rest of the input traces.

The following tables shows number of cycles for each input trace for the best policies we found which were closed and adaptive policies with different address mapping:

Input	Closed-Page Policy Address Mapping - 0	Closed-Page Policy Address Mapping - 1	Adaptive-Page Policy Address Mapping - 0	Adaptive-Page Policy Address Mapping- 1
comm1	331653892	314153864	331883492	316267468
comm2	477856581	462188464	478008741	462711100
black	247395452	252973740	247947176	255701280
fluid	335585913	348774817	333520093	351999665
freq	210921681	222943964	210978241	224061601
ferret	341017941	392913781	341331201	399187901
face	326449792	399703064	326565056	400603488
stream	246607112	265007156	247238800	265396424
swapt	364162885	380634440	367347349	382320376

For Adaptive-Page Policy except for “comm1” and “comm2” traces, address mapping 0 has fewer cycles than address mapping 1.

For Closed-Page Policy also except “comm1” and “comm2” traces address mapping 0 has fewer cycles than address mapping 1.

Results

The **Adaptive Policy** we implemented successfully shifts between Open-Page and Closed-Page depending on the **Counter** value.

It is observed that **Adaptive Policy takes fewer cycles than Open-Page Policy** and takes **slightly more cycles than Closed-Page Policy**. Therefore, **Closed-Page Policy** has the best performance overall even taking slight differences in the cycles count into account.

For Adaptive-Page Policy, **Open-Page Address Mapping Scheme (Address Mapping - 0)** has better overall performance compared to **Closed-Page Address Mapping Scheme (Address Mapping - 1)**.

For Closed-Page Policy also, **Open-Page Address Mapping Scheme (Address Mapping - 0)** has better overall performance compared to **Closed-Page Address Mapping Scheme (Address Mapping - 1)**.