International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

http://ijpsat.ijsht-journals.org
Vol. 17 No. 1 October 2019, pp. 10-16

Analyzing The Impact of Memory Controller's Scheduling Policies on DRAM's Performance

Yee Yee Soe

Faculty of Computer System and Technologies,
University of Computer Studies, Hpa-an, Kayin State

Abstract- In designing a computing system, it is necessary that comprehensive importance should be set on two common goals (i.e. increasing performance and decreasing power usage). To achieve either more performance at same power level or minimum power consumption for same performance level has needed. Main memory of a system is one of the key resources that a program needs to run hence it acts as a major contributor towards both system's performance and power consumption. Main memory's performance depends on the way it accesses its contents. It is memory controller's access scheduler that decides which command to issue in every DRAM clock cycle on the basis of employed memory access scheduling policy. Based on basic access strategy DRAM operations are scheduled in a way that it reduces DRAM's latency and power consumption by utilizing low power modes. In this paper, the various memory access scheduling algorithms on the basis of page hit rate, energy-delay product, total execution time and maximum slowdown time have compared and analyzed. This analysis contributes to better understand how the performance and power consumption of DRAM memory system is affected by the underlying memory controller's scheduling policies.

Keywords- DRAM, Performance, Power Consumption, Memory Controller, Memory Access Scheduling.

I. INTRODUCTION

Mobile computing devices require extended battery life, e.g. medical applications demand reduced power consumption in order to meet fan noise or heat limitations. The desktops and systems should be efficient in terms of energy for environmental and economic considerations [1]. Memory power is a major concern while designing all types of computing systems and devices [2]. DRAM is a main contributor in total system's power consumption [3].

Two primary ways to optimize DRAM power consumption include reducing standby power consumption and minimizing active power consumption [4]. DRAM's active power consumption can be reduced either by reducing row-buffer misses or by minimizing read-write switches. DRAM's standby power consumption can be reduced by opting any of these strategies namely, frequency scaling, self-refresh mode and power down. Hence, main memory's performance and energy consumption is largely dependent on the way it schedules the memory requests for service and accesses its contents. It is memory controller that decides which memory command to issue every DRAM clock cycle. This decision is based on memory controller's scheduling algorithm.

Several scheduling policies are employed by memory controller, such as, close page policy, first-come-first-serve (FCFS), first ready-first come first serve, Pre-Read and Write-leak (PRWL), Priority Based Fair Scheduling (PBFS), Row Buffer Locality based Drain policy (RLDP)etc. A comparative study and analysis of Pre-Read and Write-leak (PRWL) scheduling algorithm proposed in [5] and Row Buffer Locality based Drain (RLWP) scheduling algorithm proposed in [6] with two conventional memory scheduling algorithms, i.e., first-come-first-serve (FCFS) scheduling

policy and close page policy to explore the impact of simulated algorithms on DRAM's energy consumption and performance for several workload conditions.

II. THEORETICAL BACKGROUND

2.1. Main Memory System

Main memory of a computer system is one of the major resources that a program needs to run. It acts as an intermediate repository for operating data as it is located between cache system and secondary memory. The data that is useful in future is stored temporarily in main memory system. It provides temporary storage for the data that has been ejected from the cache system. In order to meet these system requirements, main memory should be faster than the secondary storage [7].

Dual Data Rate (DDR) SDRAM technology is employed in modern main memory system [8]. In DDR SDRAM, dual data rate (DDR) means its operating frequency is twice as that of command and address bus frequency because it activates output on both edges of each clock cycle (i.e., rising as well as falling edges), potentially doubling the throughput. In DDR SDRAM, the devices service row access strobe and column access strobe at the falling edge of clock signal not immediately after receiving the changed strobe signal, this is synchronous [7]. DRAM stores data in the form of charge in the capacitor which is accessed via a transistor, shown in Fig.1. The charge stored in the capacitor leaks over time. So, periodic refreshes are required to maintain data integrity and hence DRAM is dynamic in nature. A basic DRAM system supports one or more DRAM channels which consist of one or more memory modules. A DIMM is basically a collection of ranks, where each rank is subdivided into multiple banks and each bank consists of array of rows and columns. All the accesses to the main memory are made through memory controller. Memory controller is responsible for managing movement of data into and out of memory. The memory controller schedules memory commands that advances the execution of a pending read or write requests (i.e. PRECHARGE, ACTIVATE and COLUMN READ/COLUMN WRITE) or a command that manages the general DRAM state(i.e. Power-Down-Fast, Power-Down-Slow, Power-Up, Refresh, PRECHARGE, PRECHARGE-ALL-BANKS) on the basis of underlying scheduling algorithm.

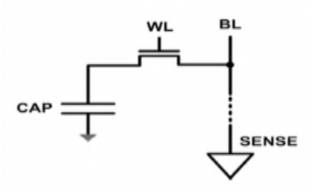

Fig.1. Basic DRAM cell structure

2.2. Memory access scheduling

Memory scheduling is the most important task of memory controller. The impact of memory scheduling has reported in determining overall system's power and performance [9]. Transaction scheduler part of memory controller picks a pending read or write request from the read or write queue and splits it into a series of DRAM commands. Then, the command scheduler part of memory controller schedules these DRAM commands while meeting associated minimal timing constraints.

Memory scheduling algorithms such as FCFS scheduling policy, close page scheduling policy, PRWL scheduling policy and RLWP scheduling policy have chosen for analysis work. **FCFS-** a variant of FCFS in which requests are ordered in the read queue by their arrival time then read queue is scanned in sequential manner until an instruction that satisfies timing constraints and can be issued in the current cycle is found. Write requests are maintained in separate write queue. Writes are drained when the write queue size exceeds a high water mark until a low water mark is reached.

Close- In a basic close-page policy, pre-charge command is issued immediately after column read/write command but a variant of close page policy is selected for analysis purpose in which precharge command is issued to the bank that last serviced a column read/write in every idle cycle.

PRWL scheduling policy focuses on reducing the frequency of entering write drain mode by interleaving memory read and write accesses. When command bus is idle, several memories read commands can be issued during drain-write. Similarly selected memory write commands can also be issued during read mode. Random Write-Leak Selection variant of PRWL have chosen because Random Write-Leak Selection performed better than Write-Leak.

RLDP scheduling algorithm exploits row buffer locality. This algorithm prioritizes requests employing row buffer locality thus improves row hit rate of both write and read requests. In this policy, switching between read to write mode or write to read mode occurs only when there are no

more row hit read requests or write requests, respectively to service.

III. Performance Analysis

This section describes the simulation environment used for the analysis of simulated memory scheduling algorithms (FCFS, Close, PRWL and RLDP). Performance analysis of FCFS scheduling policy, Close page scheduling policy, PRWL scheduling policy and RLDP scheduling policy is done by resorting "USIMM"DRAM main memory system simulator [10].

3.1. System Performance Analysis

In order to analyze the impact of memory scheduling algorithms on main memory's performance and energy consumption, several sets of experiment and adopted very widely used performance metrics such as page hit rate, energy-delay product, total execution time and maximum slowdown time have run for performance evaluation. Other than these performance metrics have evaluated these results for RLDP and PRWL scheduling policies on the basis of % decrease in EDP, total execution time and maximum slowdown time with respect to FCFS and Close page policy.

Page Hit rate- Page hit is a condition in which the page which a read/write request wants to access is already in the row buffer. A page hit results in less number of operations to be performed to service a request.

Energy-Delay product (EDP) - In order to evaluate energy efficiency, Energy-Delay product performance metric has used. This performance metric captures the goal of minimizing energy consumption (Joules) while achieving high performance (seconds) [10]. The percentage decrease in energy-delay product is used as an evaluation metric here and can be calculated by the following equation:

$$\%\ decrease\ in\ EDP = \frac{E_x - E_y}{E_x} \quad (1)$$

The equation (1) gives percentage decrease in EDP of y scheduling algorithm with respect to x scheduling algorithm. E_x is the energy-delay product of x scheduling algorithm and E_y is the energy-delay product of y scheduling algorithm.

Total Execution Time- It gives the total execution time of simulated scheduling policy. In case of multi core, total execution time is calculated by adding execution time of each core. The percentage decrease in energy-delay product is used as a performance metric here and can be calculated by the following equation:

$$\%\ decrease\ in\ Total\ Execution\ Time = \frac{T_x - T_y}{T_x} \quad (2)$$

The equation (2) gives percentage decrease in total execution time of y scheduling algorithm with respect to x scheduling algorithm. T_x is the total execution time of x scheduling algorithm and T_y is the total execution time of y scheduling algorithm.

Maximum Slowdown Time- The worst case performance is typically captured by maximum slowdown time. In order to avoid starvation, the limitations to the slowdown time of each job rather than sum of stretches of all jobs are required [11]. This motivates towards finding maximum slowdown time. We have used percentage decrease in maximum slowdown time as a performance metric for performance evaluation in our work and can be calculated by the following equation:

$$\%\ decrease\ in\ Maximum\ Slowdown\ Time = \frac{H_x - H_y}{H_x} \quad (3)$$

The equation (3) gives percentage decrease in maximum slowdown time of y scheduling algorithm with respect to x scheduling algorithm. H_x is the maximum slowdown time of x scheduling algorithm and H_y is the maximum slowdown time of y scheduling algorithm.

3.2. Experimental Results

DRAM's main memory system simulator as simulation platform has used USIMM [11]. In this simulator, memory controller issues device level memory commands on the basis of current state of channel, rank and bank. Two memory configurations named 1-channel configuration and 4-channel configuration have used for analysis purpose. The name 1-channel configuration self explains that it supports only one channel in the memory system. Both memory configurations support two ranks per channel and four banks per rank. Micron's power calculation methodology is provided in simulator for power calculations. Details of the power simulation are provided in [11]. In comparative and analytic study, the experiments to simulate multi-threaded workloads carried out from PARSEC [12] and commercial transaction processing workload in varied multi-core environment ranging from one, two, four, eight and sixteen cores. Workload details are listed in Table 1. For ease of understanding, mCo_nChi to represent m-core, n-channel simulation running workload 'i' have used. Other than workloads detailed in Table1, MT-Canneal workload is also used but MT-Canneal workload does not participate in calculating maximum slowdown time. The maximum slowdown time is calculated only for multithreaded workloads.

Table1-Workload Description

Trace File/s	Workloads with 1-Channel Configuration File	Workloads with 4-Channel Configuration File
comm2	1Co_1Ch1	1Co_4Ch1
comm1 comm1	2Co_1Ch1	2Co_4Ch1
comm1 comm1 comm2 comm2	4Co_1Ch1	4Co_4Ch1
fluid swapt comm2 comm2	4Co_1Ch2	4Co_4Ch2
face face ferret ferret	4Co_1Ch3	4Co_4Ch3
black blackfreqfreq	4Co_1Ch4	4Co_4Ch4
stream streamstreamstream	4Co_1Ch5	4Co_4Ch5
fluid fluidswaptswapt comm2 comm2 ferret ferret	-	8Co_4Ch1
fluid fluidswaptswapt comm2 comm2 ferret ferretblack blackfreqfreq comm1 comm1 stream stream	-	16Co_4Ch9

3.3. Results Analysis

In this section, simulation results of simulated memory scheduling algorithms have compared and analyzed under varied workloads and memory configurations using chosen performance metrics.

3.3.1. Page Hit Rate

Table 2-Comparison of simulated scheduling policies on the basis of page hit rate.

Workload	Read Page Hit Rate				Write Page Hit Rate			
	FCFS	Close	RLDP	PRWL	FCFS	Close	RLDP	PRWL
MT-c1	0.0033	-0.0290	0.0144	-0.0356	-0.2097	-0.2099	-0.0345	-0.4018
4Co_1Ch4	0.6291	0.5178	0.5709	0.527	0.1603	0.1508	0.3823	0.1356
2Co_1Ch1	0.5996	0.4846	0.5053	0.4746	-0.1653	-0.2475	0.1673	0.1274
4Co_1Ch1	0.5294	0.4167	0.4728	0.4272	-0.1619	-0.2084	0.1164	-0.2847
1Co_1Ch1	0.5749	0.4605	0.4743	0.4498	-0.2850	-0.2890	0.0854	-0.0189
4Co_1Ch3	0.6996	0.5952	0.6543	0.6011	0.3990	0.3860	0.5761	0.3563
4Co_1Ch2	0.5545	0.4430	0.4982	0.4528	-0.0861	-0.1163	0.1861	-0.1125
4Co_1Ch5	0.6461	0.5340	0.5930	0.5444	0.1837	0.1662	0.3985	0.1452
MTc-4	0.0185	0.0073	0.0065	0.0002	-0.6412	-0.7449	0.0091	-0.0476
4Co_4Ch4	0.0479	0.0057	0.0096	0.0026	-0.4024	-0.4406	0.0129	-0.0300
2Co_4Ch1	0.0595	0.0074	0.0063	0.0038	-0.0707	-0.1321	0.0968	0.0075
4Co_4Ch1	0.0141	0.0041	0.0041	-0.0001	-0.4801	-0.5352	0.0242	-0.0254
1Co_4Ch1	0.0160	0.0026	0.0026	0.0016	-0.0699	-0.0853	0.0040	-0.0013
4Co_4Ch3	0.0638	-0.0038	0.0130	-0.0013	-0.3775	-0.4282	0.0534	-0.0613
4Co_4Ch2	0.0197	0.0025	0.0038	-0.0002	-0.3988	-0.4230	0.0034	-0.0315
4Co_4Ch5	0.0466	0.0043	0.0085	0.0002	-0.3885	-0.4572	0.0097	-0.0595
8Co_4Ch1	0.0074	-0.0058	0.0025	-0.0091	-0.4052	-0.4768	-0.0037	-0.1556
16Co_4Ch9	-0.0140	-0.0222	-0.0025	-0.0270	-0.2988	-0.3438	-0.0169	-0.3256
1-Channel	0.5296	0.4279	0.4729	0.4301	-0.0206	-0.0460	0.2347	-0.0067
4-Channel	0.0280	0.0002	0.0054	-0.0029	-0.3533	-0.4067	0.0193	-0.0730
Overall	0.2509	0.2132	0.2132	0.1896	-0.2054	-0.2464	0.1150	-0.0435

RLDP scheduling algorithm developed in excessive page hit rate (i.e. sum of read page hit rate and write page hit rate), the results shown in **Table 2.** In terms of read page hit rate as shown in **Table 2**, it is FCFS policy that performed best among all simulated algorithms. It is PRWL scheduling algorithm that performed better in terms of general page hit rate, after RLDP. Performance of close page policy is worst among all simulated scheduling algorithms.

3.3.2. Energy-Delay Product

The simulation trend seen in Fig.2 (a) for EDP depicts that EDP of both RLDP and PRWL is less than FCFS and close scheduling policy for all scenarios. Results show that RLDP decreased EDP by 17.33% and 21.84% in 1-channel and 4-channel memory configurations respectively with respect to FCFS and 12.95% and 10.98% decrease in EDP with respect to close policy in 1-channel and 4-channel configuration respectively, Fig.2 (b) and Fig.2(c). RLDP reduced overall EDP by 19.97% and 11.93% when compared to FCFS scheduling algorithm and close page scheduling algorithm respectively, Fig.2 (d).

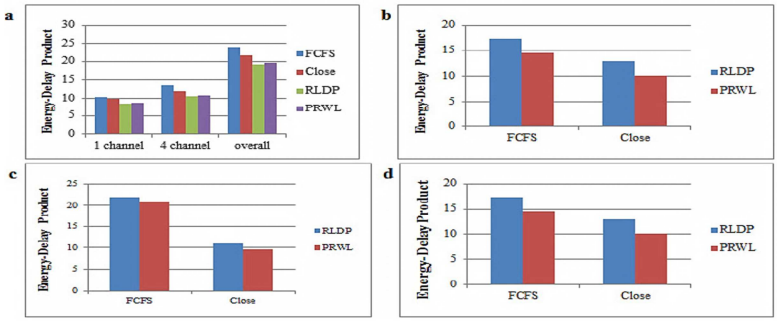

Fig.2 (a) EDP (Js) Comparison (b) % decrease in EDP for 1-channel configuration. (c) % decrease in EDP for 4-channel configuration (d) % of overall decrease in EDP.

3.3.3. Total Execution Time

Fig.3 (d) show that RLDP performed best among all the simulated policies for all memory configurations according to the results shown in Fig.3 (a). After RLDP, it is PRWL that managed to perform better. RLDP decreased overall total execution time by 9.99% with respect to FCFS and 6.05% with respect to close scheduling policy. Whereas PRWL reduced overall total execution time by 9.30% and 5.33% with respect to FCFS and close scheduling policies respectively.

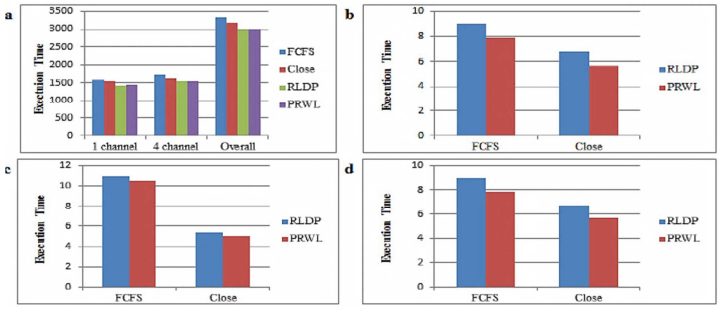

Fig. 3(a) Total Execution Time (mCyc) Comparison; (b) % decrease in Total Execution Time for 1-channel configuration (c) % decrease in Total Execution Time for 4-channel configuration; (d) % of Overall decrease in Total Execution Time

3.3.4. Maximum Slowdown Time

The maximum slowdown time again RLDP is best among all simulated policies from Fig.4 (a) to Fig.4 (d). RLDP shows 8.89% and 11.02% decrease in maximum slowdown time for 1-channel and 4-channel memory configurations respectively, whereas PRWL reduced maximum slowdown time by 7.40% for 1-channel and 9.45% for 4-channel memory configurations, when compared to FCFS. RLDP and PRWL reduced overall maximum slowdown time by 10% and 9.23% with respect to FCFS and 5.64% and 4.83% with respect to PRWL.

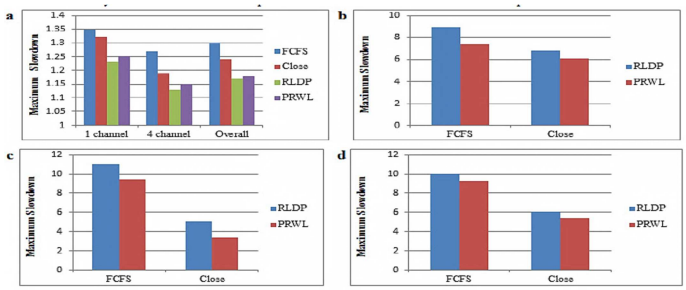

Fig.4 (a) Maximum Slowdown Time Comparison (b) % decrease in Maximum Slowdown Time for 1-channel configuration (c) % decrease in Maximum Slowdown Time for 4-channel configuration (d) % decrease in overall Maximum Slowdown Time

IV. Conclusion And Future Work

In this paper, the memory controller's scheduling algorithms *i.e.* FCFS, close page policy, PRWL and RLDP have compared and analyzed. The simulation results obtained show that RLDP scheduling policy out performs among all simulated scheduling policies for both 1-channel and 4-channel memory configurations in terms of row-hit rate, energy-delay product, maximum slowdown time and total execution time. The evaluation metrics used for analysis and comparative study are not fully independent of each other. An algorithm having highest hit rate results in reduced number of operations to be performed and thus reduced access time. FCFS scheduling algorithm have highest hit rate during read access but showed worst performance among all in terms of maximum slowdown time, total execution time and EDP.

Worst performance of FCFS is because it does not exploit bank level parallelism and row buffer locality hence results in increased read/write stall time. Increased stall time results in increased maximum slowdown time. Increase in maximum slowdown time results in increased execution time thereby increased energy delay product. The performance of RLDP scheduling algorithm was best among all simulated algorithms because it combines the features of row buffer locality, delayed write drain and delayed close policy which results in improved hit rate for both read and write requests. Row hit requests require fewer operations to be performed which in turn reduces the service time thereby decreased maximum slowdown time, total execution time and energy-delay product. After RLDP scheduling algorithm, PRWL scheduling policy performed better because it reduces the frequency of entering write drain mode and thus reduces the time for which memory reads are stalled. Reduced memory read stall time results in minimized execution time, slowdown time and energy-delay product.

Both RLDP and PRWL scheduling algorithm work on reducing the number of operations to be performed. Because of reduced number of operations both RLDP scheduling policy and PRWL scheduling policy managed to perform better as compared to FCFS scheduling policy and close page scheduling policy.

As a further extension in this field, memory access scheduling algorithms can be analyzed for some other performance parameters like read/write latency, read-write switching etc. On the basis of performed analysis a new approach for memory scheduling algorithm can be proposed. In future, the analysis work can be extended for nonvolatile memory (NVM) enabled hybrid memories.

References

[1] X. Fan, C. Ellis, and A. Lebeck. Memory controller policies for DRAM power management. In Proceedings of the 2001 International Symposium on Low Power Electronics and Design, pages 129-134, 2001.

[2] T. Vogelsang. Understanding the energy consumption of dynamic random access memories. In MICRO, 2010.

[3] I. Hur and C. Lin. A comprehensive approach to DRAM power management. In HPCA-14, 2008.

[4] KarthikChandrasekar. High-Level Power Estimation and Optimization of DRAMs [PhD Thesis]. Netherlands, Delft University of Technology; 2014.

[5] Long Chen, YananCao, Sarah Kabala andParijatShukla. Pre-Read and Write-Leak Memory Scheduling Algorithm. In 3rd JILP Workshop on Computer Architecture Competitions: Memory Scheduling Championship, MSC, 2012.

[6] Young-Suk Moon,Yongkee Kwon, Hong-Sik Kim, Dong-gun Kim, Hyungdong Hayden Lee and

Kunwoo Park. The Compact Memory Scheduling Maximizing Row Buffer Locality. In 3rd JILP Workshop on Computer Architecture Competitions: Memory Scheduling Championship, MSC, 2012.

[7] GoranNarancic. A Preliminary Exploration of Memory Controller Policies on Smartphone Workloads [MS Thesis]. Canada, University of Toronto; 2012.

[8] JEDEC Solid State Technology Association. DDR3 SDRAM specification. Tech. Rep. JESD79-3E, Arlington, VA, 2010.

[9] M. Bojnordi and E. Ipek. PARDIS: A programmable memory controller for the DDRx interfacing standards. In Proceedings of ISCA, 2012.

[10] R. Gonzalez and M. Horowitz, "Energy Dissipation in General Purpose Processors," in Proceedings of the IEEE Symposium on Low Power Electronics, Oct. 1995, pp. 12-3.

[11] M. A.Bender, S.Chakrabarti, and S. Muthukrishnan. Flow and stretch metrics for scheduling continuous job streams. In Proceedings of the ACM Symposium on Discrete Algorithms (SODA), 1998.

[12] C. Bienia, S. Kumar, J. P. Singh, and K. Li. The PARSEC Benchmark Suite: Characterization and Architectural Implications. In Proceedings of PACT, 2008.