CS304

Assignment 3

Ira Syamira Sukimin

isuk218 (155407288)

Processor: Intel Core i5-4210U CPU @ 1.70GHz × 4

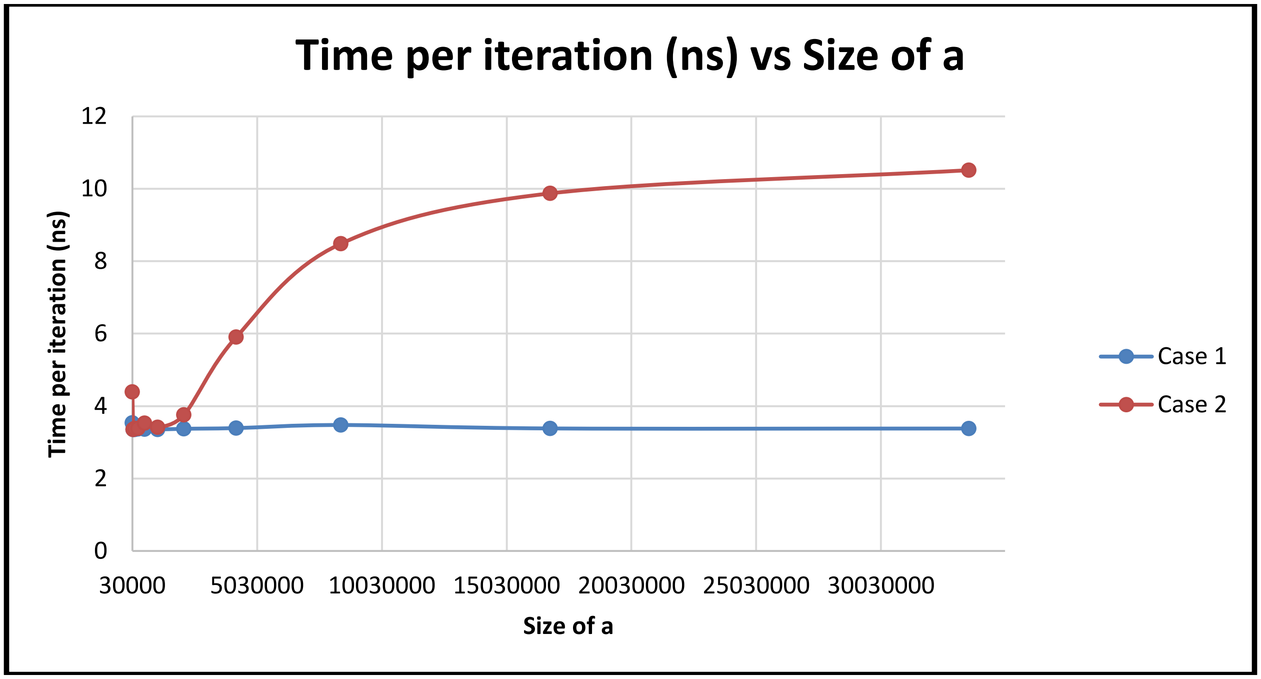
L1 Data Cache = 2 x 32 KBytes

L2 Cache = 2 x 256 KBytes

L3 Cache = 4 MBytes

Task 1 - Cache Measurement

N	Size of a	M	Iterations (N*M)	Case 1		Case 2	
				Time taken (s)	Time per iteration (ns)	Time taken (s)	Time per iteration (ns)
8192	32768	10000	81920000	0.29	3.540039063	0.36	4.39453125
16384	65536	10000	163840000	0.55	3.356933594	0.55	3.356933594
32768	131072	10000	327680000	1.1	3.356933594	1.11	3.387451172
65536	262144	10000	655360000	2.21	3.372192383	2.22	3.387451172
131072	524288	10000	1310720000	4.41	3.364562988	4.63	3.532409668
262144	1048576	5000	1310720000	4.4	3.356933594	4.48	3.41796875
524288	2097152	5000	2621440000	8.85	3.37600708	9.86	3.761291504
1048576	4194304	2000	2097152000	7.12	3.395080566	12.39	5.90801239
2097152	8388608	2000	4194304000	14.6	3.480911255	35.59	8.48531723
4194304	16777216	1000	4194304000	14.2	3.385543823	41.42	9.875297546
8388608	33554432	500	4194304000	14.19	3.383159637	44.1	10.51425934



From the graph above:

1. Case 1 (sequential method) takes a shorter amount of time to complete one iteration as compared to case 2 (random access method).
    - This is because the subsequent elements of the array would be in the cache, given that it is within the cache line size. Therefore, accessing sequential elements would take a shorter amount of time as the data is readily available in the cache due to increased probability of a cache hit. Accessing the array randomly results in a higher amount of time per iteration as there is a higher probability for a cache miss.
2. Case 1 is more consistent in time per iteration as size of a increases meanwhile time per iteration for case 2 increase significantly as size of a increases.
    - As the size of a increases, the time taken per iteration for case 2 increases significantly as the probability of a cache miss is increased. In contrast, case 1 has a consistent time per iteration when the value of a increases as the cache hit probability is consistent.

Task 2 – Matrix Product

1. Time taken: 4.11 seconds
    - A matrix is an array of array (2D array) consisting of rows and columns
    - Cache is stored row first, followed by column
    - Matrix b fetches the next element in the new row
    - If there is a cache miss (element is not in cache), it will be redirected to memory which consumes more time
2. Time taken: 4.17 seconds
3. Time Taken: 0.08 seconds
    - The blocking method works using smaller matrix blocks to reduce execution time
    - The large matrix is separated into smaller blocks of dimension that matches the cache line size
    - This ensure all the values in the smaller block is in the cache which enables faster execution and fetching of elements