Drexel University

Electrical and Computer Engineering Dept.

ECEC-413

Assignment 1:
Ways to SAXPY

Minjae Park

John Truong

Professor Naga Kandasamy

Date: 04/26/2020

SAXPY Version 1 and Version 2 Explanation:

For this assignment, version one of SAXPY (Single-Precision AX plus Y) code used the chunking method. In this method, elements of the data array were split into chunks using the number of threads. Each thread is given an offset and number of elements to work on as arguments packed into the structure, along with the memory addresses of the input and output arrays. Within the thread function, a check is in place to see whether the thread has a tid of numthreads-1, in which case that thread will work till the end of the array.

Version two uses the striding method. In this method, the arguments passed to each thread are simpler: number of threads and the tid, along with the memory addresses of the input and output arrays. Within the thread function, the tid (thread ID) of each thread becomes the offset, and the "stride" or how many elements the thread skips over is indicated by the total number of threads. For this instance, if four threads are created: thread 0 will take elements 0, 4, 8, and so on; thread 1 will take elements 1, 5, 9; thread 2 will take elements 2, 6, 10; and lastly thread 3 will take elements 3, 7, 11.

The actual computation within the loops is the same as the single threaded version, where the ith element of x is multiplied by a and then added to the ith element of y.

Results: Chunking-Method vs. Striding-Method:

When the size of data is small, such as in the instance of 10^4 elements, the overhead of creating threads and termination of threads is much greater than the actual time it would take to calculate the problem single-threaded. But as the data size grows, the multi-threaded implementations start to show efficiency over the single-threaded version. Version 2, which uses the striding method, takes more time than the chunking method due to the more frequent cache misses due to striding. The reason is due to false sharing that is occuring in the cache which then leads to invalidation of the cache; leading to longer execution times.

	Number of Threads								
	Reference			Chunking			Striding		
Data Size	4	8	16	4	8	16	4	8	16
10^4	0.000009	0.001811	0.285337	0.000754	0.001411	0.167092	0.000224	0.003188	0.357863
10^6	0.000010	0.001916	0.333181	0.001393	0.001407	0.128756	0.000660	0.004999	0.657573
10^8	0.000010	0.001705	0.485722	0.001393	0.001888	0.118506	0.000660	0.008037	1.121406