Alston Godbolt

Program number 2

4/24/17

Ecampus

1. Tell what machine you ran this on

   The experiment was conducted on a Lenovo G50 Laptop. Below are the product specifications

   OS: Windows 10 Home

   Processor: Intel Core i3 – 4030U CPU @ 1.90 GHz

   Installed RAM: 4 GB

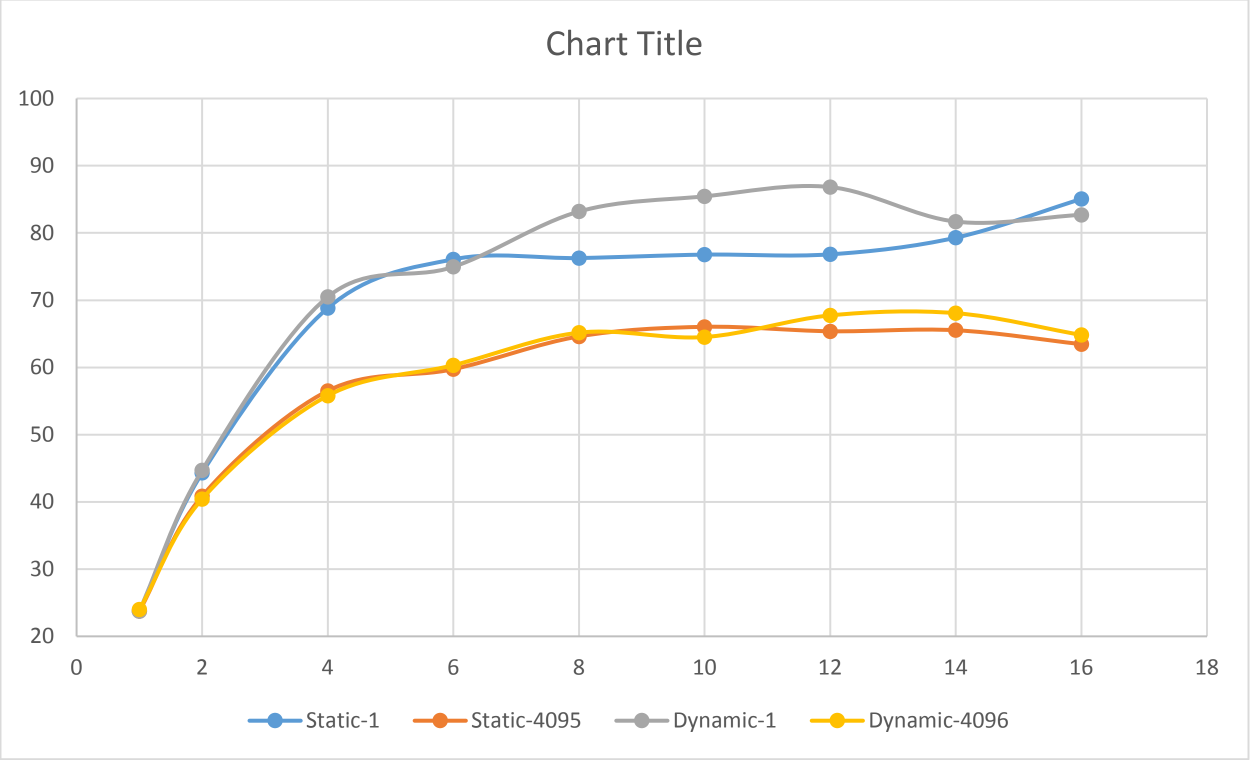
   System type: 64 bit operating system

   In addition, I run the program in Visual Studio 2017

2. Create a table with your results.

	Static-1	Static-4095	Dynamic-1	Dynamic-4096
1	23.75	23.74	23.76	23.98
2	44.32	40.87	44.72	40.44
4	68.83	56.52	70.49	55.8
6	76.08	59.74	74.96	60.33
8	76.27	64.61	83.19	65.17
10	76.78	66.04	85.45	64.52
12	76.82	65.38	86.83	67.74
14	79.3	65.53	81.69	68.07
16	85.06	63.46	82.7	64.83

3. Draw a graph. The X axis will be the number of threads. The Y axis will be the performance in whatever units you sensibly choose. On the same graph, plot 4 curves:
   - static,1
   - static,4096
   - dynamic,1
   - dynamic,4096



4. What patterns are you seeing in the speeds?

The threads with the chunksize of 1 vastly outperformed the threads with the chunksize of 4096. In addition, the dynamic scheduling outperformed the static scheduling between threads 4 and 14. A scheduling type of static are assigned to the threads before the loop is executed, whereas with scheduling type dynamic the iterations are assigned to the threads while the loop is executing (Zahran, 2017).

5. Why does chunksize 1 vs. 4096 matter like this?

   The chunksize is important because it is how the program is assigning iterations to each of the threads. With a chunksize of 1, each of the threads is receiving the next iteration. However, a chunksize of 4096 each thread is given 4096 “chucks” of the iteration. See below for a clearer example.

Thread	chunksize 1		
1	0	10	20
2	1	11	21
3	2	12	22
4	3	13	23
5	4	14	24
6	5	15	25
7	6	16	26
8	7	17	27
9	8	18	28
10	9	19	29
Thread	Chunksize 4096		
1	0		
2	4096		
3	8192		
4	8190		
5	12286		
6	16382		
7	20478		
8	24574		
9	28670		
10	32766		

6. Why does static vs. dynamic matter like this?

   As previously mentioned static scheduling occurs before the loop is executed, which means all of the iterations are scheduled before the program knows how much work the program will require from each thread. The result can be a program that talks longer to complete because a thread can take longer to complete the given tasks. With dynamic scheduling iterations are broken up into chunks of consecutive iterations. Each thread executes a chunk, and when a thread finishes a chunk, it requests another one from the run time system (Zahran, 2017).

Works Cited

Zahran, M. (2017, April 24). Lecture 14: OpenMP III. New York, New york, United States of America.