HPC: High-Performance Computing
Academic Year: 2023 - 24

Dr. Praveen Kumar Alapati
praveenkumar.alapati@mahindrauniversity.edu.in

Department of Computer Science and Engineering

Ecole Centrale School of Engineering

Mahindra™ University
Global Thinkers. Engaged Leaders.

Assignment 4 (Due Date: May 20, 2024)

Create the following concurrent data structures with 1 Million nodes and perform the basic operations (*contains*, *insert*, and *remove*) by varying the number of threads from 1 to 20 (i.e., 1, 2, 4, 6, 8, 10, 12 ... 20) and for different workloads (100C-0I-0D, 90C-9I-1D, 50C-25I-25D, 30C-35I-35D, 0C-50I-50D). Prefill the data structure with 50% of elements and duraion of each run is 10 seconds.
To measure the throghput, consider the average of FIVE runs and also measure the cache misses per 1000 operations using **perf** tool.

1. Concurrent Binary Search Tree
2. Concurrent AVL Tree
3. Concurrent Skiplist
4. Concurrent Stripped Hashset
5. Concurrent Refinable Hashset

All the best ☺

Assignment3: Due Date April 4, 2024

Develop parallel codes for the following problems using JAVA (or C++). Report the speedup of your implementations by varying the number of threads from 1 to 16 (i.e., 1, 2, 4, 6, 8, 10, 12, 14, and 16). Use the systems in our IT Lab and consider *gettimeofday*() **to measure the runtime of an execution. Repeat the experiment five times and consider the average of five runs. The running time of each run should be reported. Use an Excel workbook with the name TEAMNUM_ASSIGN_NUM.xlsx to maintain all runtimes of your executions. Finally, draw appropriate plots using the GNU plot.**

Submission Guide Lines:

- Attach. Name and Type: TEAMNUM_ASSIGN_NUM.zip
- Write a readme file to understand your solutions.
- Submit source files only.

Learn the art of multi-core and many-core programming

Assignment3: Due Date April 4, 2024

Create a sorted linked List with 1000 nodes using lazy synchronization technique (assume that the nodes consist of fields: key and next. The range of keys from the set $\{0 \ldots 2^{12}\}$).

Measure the time to perform 100 Million operations for the workload (50C, 25I, 25D) by varying the number of threads from 1 to 16 and using the following locks (or synchronization objects).

1. CLH Lock
2. MCS Lock
3. TTAS with Exponential Back-off
4. Cohort Lock
5. Filter Lock

Assignment2: Due Date March 12, 2024

Develop parallel codes for the following problems using JAVA (or C++). Report the speedup of your implementations by varying the number of threads from 1 to 16 (i.e., 1, 2, 4, 6, 8, 10, 12, 14, and 16). Use the systems in our IT Lab and consider *gettimeofday*() to measure the runtime of an execution. Repeat the experiment five times and consider the average of five runs. The running time of each run should be reported. Use an Excel workbook with the name TEAMNUM_ASSIGN_NUM.xlsx to maintain all runtimes of your executions. Finally, draw appropriate plots using the GNU plot.

Submission Guide Lines:

- Attach.Name and Type: TEAMNUM_ASSIGN_NUM.zip
- Write a readme file to understand your solutions.
- Submit source files only.

Learn the art of multi-core and many-core programming

Assignment2: Due Date March 12, 2024

1. **Compute X^{15} of a matrix X**: Assume that X is a complex-valued matrix of order 2048 X 2048. It is represented by a pair of matrices (A, B) where A and B contain values from the set {-1, 0, 1}. Write a multi-threaded code to compute X^{15}.
2. Develop a program to create a **Sorted Linked List** with 1000 nodes (assume that the nodes consist of fields: key and next) and the range of keys from the set $\{-2^{10} \ldots 2^{10}\}$. Consider the workload (x,y,z) (x%-contains(), y%-insert(), and z%-remove()) and perform 100 million operations on this list **by varying the number of threads**. Assume that (x,y,z)∈{ (100,0,0), (50,25,25), (0, 50, 50)}.
3. **Multi-threaded Sorting:** Sort 10^9 integers in the range 0 to 10^3 using the following sorting techniques:
   a. Merge Sort
   b. Quick Sort
   c. Counting Sort

Assignment2: Due Date March 12, 2024

4. **Sorting and Merging Subsequences:** You are given a sequence of 10^9 elements to sort. The input sequence consists of 10^3 subsequences, each of which contains 10^6 elements. The elements in a given subsequence are all smaller than the elements in the succeeding subsequence and larger than the elements in the preceding subsequence. Sort the input sequence by varying the number of threads.
5. **Find a median of 10^9 elements using the median-of-medians.** Assume that the elements from the set $\{-2^{30} \ldots 2^{30}\}$.

Verify all your results by using sequential code. If the results are the same, then only we consider the code (or testcase is passed).

Assignment1: Due Date February 13, 2024

1. **Compute A^2 of a matrix A** for the orders 2048x2048 and 4096x4096. Initialize the elements of the matrix A with random values from the set {-1, 0, 1}. Assume that the matrix is given in row-major order. If you perform any transformation, that also has to be accounted for in the runtime as well. Consider the following implementations to find the A^2,
   - Ordinary Matrix Multiplication (OMM).
   - Block Matrix Multiplication (BMM) using block sizes: 4,8,16,32,64.
2. **N-Queens Problem:** Given an N × N chessboard and N-Queens. Place N-Queens on the chessboard in non-attackable positions. Consider the different values of N = 12 and 16. Store all the solutions.

Assignment1: Due Date February 13, 2024

3. **Minimum, Maximum, and Mean:** We have an array of size 2^{28}, and all elements of the array are initialized with random integers from the set $\{0, 1, \ldots, 10^9\}$. Find the minimum, maximum, and mean of all.
4. **Prime Numbers:** Generate all the prime numbers between 2^{40} and 2^{41}.
5. **Dot Product:** Calculate the dot product of two vectors. Assume that the length of each vector is 10^9 and the vectors are initialized with random integers from the set $\{ -1, 0, 1\}$.

Verify all your results by using sequential code. If the results are the same, then only we consider the code (or testcase is passed).