

Wait-free Snapshot

Concurrent Programming

Programming Project #3

Final due date: Nov 10, 2019 (HARD DEADLINE)

1 TASK OVERVIEW

Implement the **Wait-free snapshot** algorithm that you have learned previous class.

2 TASK DETAILS

You have N worker threads, and each thread has its own *local integer value*. The value can only be updated by the owner thread, but it can be read by other threads. This task is to implement the wait-free snapshot algorithm in the textbook.

1. For each thread, keeps *update* its local variable with a random value during *1 minute*. The *update* need to be accompanied by the *scan* function (making a snapshot) as the

algorithm describes.

2. Parameterize the number of threads using the command line parameter.
3. Count the number of total updates (*throughput*) that all threads made.
4. Adjust the number of thread with 1, 2, 4, 8, 16, and 32, respectively, and analyze the performance difference. (The performance of your program does not have to be fast, but it must be correct.)
5. Write a report containing the analysis on your Wiki page of the hconnect project.

3 TEST PROTOCOL

Your program should take one command line arguments. Example of the execution is:

<code>\$. /run N</code>
<code>N: Number of worker threads</code>

After your program finished, the number of total updates made by all worker threads need to be printed on the standard output.

4 SUBMISSION

You should upload your project into **project3** directory of your “hconnect” repository. You must create a proper **Makefile** in the project3 directory. The name of your executable file must be **run**, and it must be at the project3 directory as well.

You also need to upload your assignment report in your gitlab wiki page of your hconnect project. Please set a name of the wiki page as project3.

Please enjoy this project and have fun !!