

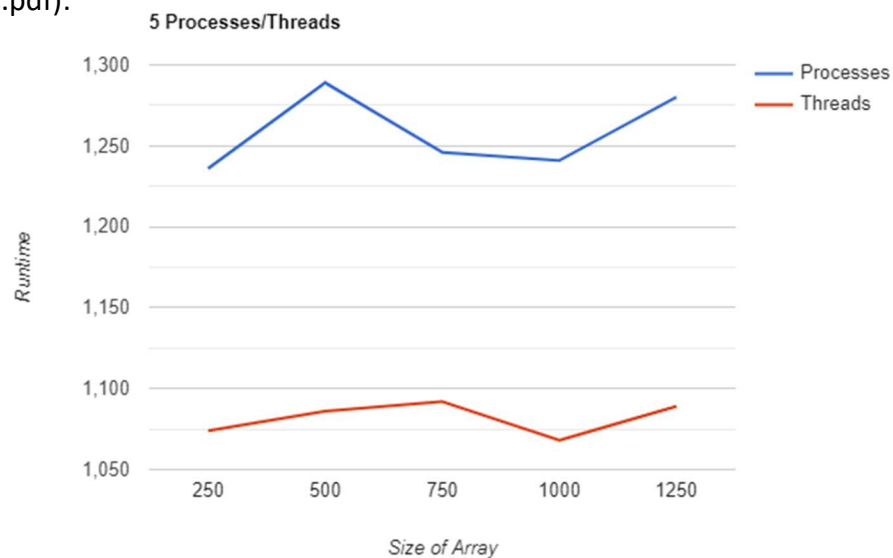
Test 1: variable array size, constant number of processes and threads

These tests will include 5 different sized intervals, 50, 100, 150, 200, and 250. They will be tested on the size interval * #procs/threads. This test will show that the search takes roughly the same amount of time regardless of the size of the interval and array if the number of processes or threads is consistent. (Hopefully the threads are consistently faster)

1. 5 Processes/Threads

- ./searchtest 250 50
- ./searchtest 500 100
- ./searchtest 750 150
- ./searchtest 1000 200
- ./searchtest 1250 250

Sample Graph (not sure on the accuracy of ilab runtimes, graphs would be in results.pdf):



2. 10 Processes/Threads

- ./searchtest 500 50
- ./searchtest 1000 100
- ./searchtest 1500 150
- ./searchtest 2000 200
- ./searchtest 2500 250

3. 15 Processes/Threads

- ./searchtest 500 50
- ./searchtest 1000 100

- ./searchtest 1500 150
- ./searchtest 2000 200
- ./searchtest 2500 250

4. 20 Processes/Threads

- ./searchtest 1000 50
- ./searchtest 2000 100
- ./searchtest 3000 150
- ./searchtest 4000 200
- ./searchtest 5000 250

Test 2: constant array size, variable number of processes and threads

To activate this test mode, input -1 as the interval. The program will calculate all intervals (up to 50 processes/threads maximum) that the array is divisible by and run search on them. EX) For an array of size 100, the possible intervals are 100, 50, 25, 20, 10, 5, 4, 2 which correspond to 1, 2, 4, 5, 10, 20, 25, 50 procs/threads respectively. (Graph will be runtime for the y-axis and number of processes for the x-axis). Run this on arrays of various sizes (each generating a graph) to see the overall trend of number of processes or threads and runtime. (Again hopefully threads are faster in all tests)

1. Various sized arrays

- ./searchtest 100 -1
- ./searchtest 250 -1
- ./searchtest 500 -1
- ./searchtest 1000 -1
- ./searchtest 5000 -1

We could have each test for procs and threads have their own graph or combine them into a single graph showing both like in the sample graph above. This would give us 9-14 graphs which can all be duplicated. Can expand the sizes of either test to generate more graphs.