Homework 1

platform

CPU information	memory size	kernel version	machine type
Intel(R) Core(TM) i7-5500U CPU @ 2.40GHz Logic processor: 4	12G	4.15.0-29-generic	physical machine

The measurement result

I generate random arrays of integers of different lengths. The execution times of the query elements measured under single process and single thread conditions were used as a control group to measure the execution times under different process and thread conditions.

The parameters are listed in the table below.

Array type	Integer array
Array length	256, 512, 1024, 2048, 3072, 4096, 8192
Number of processes	2, 4, 6, 8
Number of threads	2, 4, 6, 8

The following list shows the execution times for different array sizes

- **Array length: 256**
  Single process execution time: 0.001 (ms). the element number is: 2
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	1.254	2.84	5.659
4	1.978	2.896	1.454
6	12.962	22.375	1.55
8	3.292	13.476	5.209

- **Array length: 512**
  Single process execution time: 0.001 (ms). the element number is: 3.
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	1.186	3.686	1.796
4	1.328	1.183	7.801
6	12.477	1.429	1.593
8	5.626	1.875	10.129

- **Array length: 1024**
  Single process execution time: 0.003 (ms) the element number is: 7.
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	1.907	4.41	2.389
4	22.657	26.8	51.059
6	55.022	24.212	42.989
8	29.64	32.732	35.117

- **Array length: 2048**
  Single process execution time: 0.007 (ms) the element number is: 12.
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	26.591	1.296	2.878
4	7.702	16.935	18.063
6	11.028	15.224	14.844
8	43.278	20.432	14.506

- **Array length: 3072**
  Single process execution time: 0.009 (ms). the element number is: 21.
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	3.883	5.641	0.832
4	2.423	1.671	2.444
6	7.532	11.946	10.374
8	6.689	17.083	12.051

- **Array length: 4096**
  Single process execution time: 0.012. the element number is: 22
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	1.756	0.976	1.055
4	5.348	3.611	8.75
6	5.897	8.074	7.026

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
8	3.05	7.993	6.222

- **Array length: 8192**
  Single process execution time: 0.025 (ms). the element number is: 51.
  - multi-process and multi-threads execution time table list:

Number of processes/threads	Used times (ms) (process)	Used times (ms) (thread with global counter)	Used times (ms) (thread with pthread_exit)
2	1.353	0.373	0.413
4	1.008	2.128	3.374
6	4.384	11.029	8.059
8	3.499	10.36	3.974

possible justification for the results

1. The array length is not particularly large and the single process calculation performs well. Multi-process and multi-thread performance degrades because both multi-process and multi-thread require resource request and allocation scheduling, as well as memory I/O, which results in increased execution time spent on resource scheduling and I/O.
2. Multi-process computation requires multiple memory copies. With small array sizes, frequent memory copies lead to increased memory I/O time, resulting in performance degradation.
3. As the number of CPU cores on the physical machine is 4, when the number of multi-processes equals the number of cores, each process is allocated to a different core for computation, so the performance is excellent compared to other process count cases.
4. The performance of multi-threaded computing is better than that of multi-processing because of the shorter I/O times due to shared resources of multi-threaded computing.