

**The Arab American University**

FACULTY OF ENGINEERING

Parallel and Distributed Computing

**Parallel and Distributed Computing PROJECT 2**

|  |  |
| --- | --- |
| **ID:** | **202111358** |
| **NAME:** | **Hala Jabareen** |

Experiments:

## Hardware Specifications:

* **CPU**: Intel(R) Core(TM) i5-7300U @ 2.60GHz
* **Cores**: 2 physical cores, 4 logical processors (hyper-threaded)
* **RAM**: 16 GB
* **Operating System**: Windows 11

## Input sizes:

Tested array sizes = 32, 512, 1024, 4096, 8192, 16384, 32768.

Threads count = 2, 4, 8.

Results:

**Number of threads=2**

|  |  |  |  |
| --- | --- | --- | --- |
| **Array size** | **Seq. time** | **Par. time** | **Speed. up** |
| **32** | **0.0000100136** | **0.00038** | **0.026349** |
| **512** | **0.000265** | **0.000331** | **0.801635** |
| **1024** | **0.003029** | **0.001754** | **1.726601** |
| **4096** | **0.003475** | **0.002282** | **1.523162** |
| **8192** | **0.005973** | **0.004153** | **1.438225** |
| **16384** | **0.005973** | **0.010044** | **0.59463** |
| **32768** | **0.005973** | **0.022409** | **0.266532** |

**Number of threads=4**

|  |  |  |  |
| --- | --- | --- | --- |
| **Array size** | **Seq. time** | **Par. time** | **Speed. up** |
| **32** | **0.000010411** | **0.000653** | **0.01595** |
| **512** | **0.000287692** | **0.000807** | **0.35651** |
| **1024** | **0.000610272** | **0.217184** | **0.00281** |
| **4096** | **0.00314633** | **0.001862** | **1.690073** |
| **8192** | **0.006618023** | **0.004031** | **1.641901** |
| **16384** | **0.014811367** | **0.0083** | **1.784437** |
| **32768** | **0.032530067** | **0.01885** | **1.725767** |

**Number of threads=8**

|  |  |  |  |
| --- | --- | --- | --- |
| **Array size** | **Seq. time** | **Par. time** | **Speed. up** |
| **32** | **1.06E-05** | **0.000984** | **0.010823** |
| **512** | **0.000285** | **0.001045** | **0.272409** |
| **1024** | **0.00062** | **0.001309** | **0.473777** |
| **4096** | **0.003114** | **0.002968** | **1.049045** |
| **8192** | **0.006307** | **0.005307** | **1.188426** |
| **16384** | **0.016009** | **0.008603** | **1.86082** |
| **32768** | **0.031179** | **0.022957** | **1.358154** |

Challenges & Solutions

A number of challenges were experienced during OpenMP implementation of the project.

The Pthreads version had one of the primary problems which was the overhead of managing the thread. Thread creation and thread joining were complex and became time-consuming. This has been resolved by the update to OpenMP where #pragma omp parallel for hides thread management and thus simplifies the code to a great extent.

A more underlying problem was the fact that Bitonic Sort was naturally recursive, hence challenging deep level parallelism. Rather than parallelizing recursive calls, we only parallelized the top level in initial chunk sorting and instead leave the merge and recursion part sequential in order to ensure the algorithm is correct.

Conclusion

Essentially switching to OpenMP increased the ease of code and performance significantly.

Having concentrated parallelism at the chunk-sorting stage and proceeding over merging cautiously, we have created an observable speedup that does not affect the correctness or the stability level.

OpenMP has presented itself as efficient and powerful in parallel programming in those instances when the problem can be easily split up. In the case of Bitonic sort we were able to write cleaner code, save on thread overhead and achieve better scaling effortlessly.