

Assignment 2

COTTON: A light-weight work-stealing runtime for async-finish parallelism

Speedup

Nqueens Test Case (N = 12)

We have averaged the runtime over five executions to remove any scheduling effect.

numThreads	1	2	3	4
First Execution (s)	2.012	1.302	1.175	1.086
Second Execution (s)	2.021	1.337	1.167	1.043
Third Execution (s)	2.114	1.320	1.128	1.024
Fourth Execution (s)	2.048	1.320	1.172	1.023
Fifth Execution (s)	2.006	1.276	1.151	1.089
Mean T_{parallel}	2.040	1.311	1.158	1.053

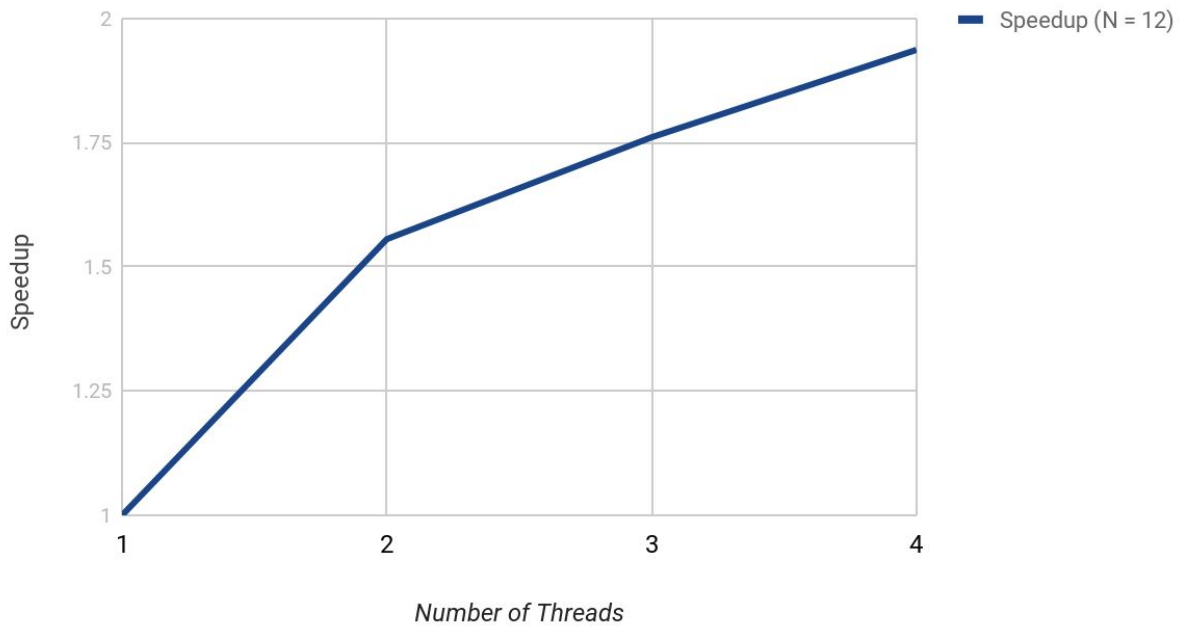
$$T_{\text{serial}} = 2.040$$

Speedup obtained using Mean parallel runtime -

1	2	3	4
1.000	1.556	1.761	1.937

Speedup Graph

Speedup Graph



Individual Contribution

1. Dattatreya Mohapatra

- `cotton_runtime::lib_key_init()`
- `cotton_runtime::get_threadID()`
- `cotton_runtime::find_and_execute_task()`
- `cotton::async()`
- `cotton_runtime::Deque::isEmpty()`
- `cotton_runtime::push_task_to_runtime()`
- `cotton_runtime::grab_task_from_runtime()`
- `Deque()`

2. Viraj Parimi

- a. `cotton::finalize_runtime()`
- b. `cotton::end_finish()`
- c. `cotton::start_finish()`
- d. `cotton::init_runtime()`
- e. `cotton_runtime::thread_pool_size()`
- f. `cotton_runtime::Deque::push_to_deque()`
- g. `cotton_runtime::Deque::pop_from_deque()`
- h. `cotton_runtime::Deque::steal_from_deque()`