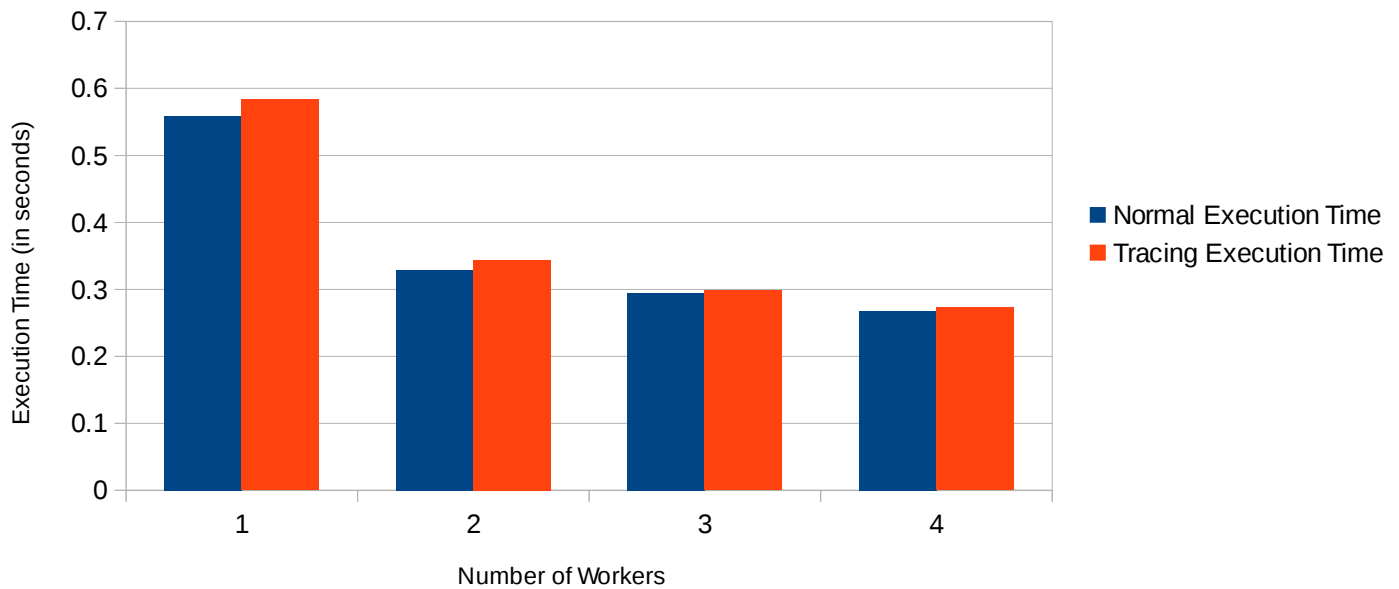


Steal Tree: Low-Overhead Tracing of Work Stealing Scheduler

Ngeens Execution Overhead by Tracing

Represent Difference in Execution Time of Default Cotton-Runtime and with Tracing



Contributors :

Sagar

* Creating all the structures for the Task and WorkingPhases and there member functions and static functions including data structures for :

- ContinuationHdr (86)
- Task (112)
- PipeDeQueue (150)
- WorkingPhaseInfo (268)
- HelpFirstInfo (306)
- WorkingStateHdr (396)
- ReplayWorkingStateHdr (449)

* functions for reading and writing WorkingStateHdrs for all workers (all tracing info) this also includes other member function of the structures that help writing them into files

- writeWorkerStateHdrs (518)
- readReplayWorkerStateHdrs (529)

* changes in the steal (for tracing and replay) and async

- line number 739 (tracing)
- line number 816 (replay-steal)

- line number 1089 (replay-async)

* creating library functions like :

- updateStealTask (1057)
- markStolen (575)
- init_executionTask (556)
- getEnvVariable (949)
- init_tracing (999)
- init_replay (990)