Threaded Programming

Coursework Part 3

epcc

THE UNIVERSITY OF EDINBURGH

Overview

- Implement a loop scheduling algorithm which is not one of the OpenMP schedule options
- Compare the results to those you obtained in Part 2
- Write a report that describes your implementation and presents and analyses the results
- Submit both code (30 marks) and report (30 marks)
- Deadline is 16:00, Fri 3rd December.
- Report length guideline is 6-8 pages
- Submit all code versions that you report results for

Affinity scheduling

- Each thread is initially assigned a (contiguous) local set of iterations.
- For a loop with n iterations, and p threads, each thread's local set is initialised with (approximately) n/p iterations.
- Every thread executes chunks of iterations whose size is a fraction 1/p of the remaining iterations its local set, until there are no more iterations left in its local set.
  - Note that once a chunk is assigned to a thread, that thread should complete all the iterations in the chunk
  - The remaining iterations are those that have not yet been assigned

- If a thread has finished the iterations in its local set, it determines the thread which has most remaining iterations (the “most loaded” thread) and executes a chunk of iterations whose size is a fraction 1/p of the remaining iterations in the “most loaded” thread’s local set.

- Threads which have finished the iterations in their own local set repeat the previous step, until there are no more iterations remaining in any thread’s local set.

- Think carefully about what shared data structures you need
  - Keep these to a minimum
- Take great care to synchronise accesses to these correctly
  - No race conditions allowed, even if they appear harmless!
- Don’t overcomplicate – get something simple working first, then refine your solution if needed.
- Get the basics of software engineering right
  - Formatting, modularity, comments, clear naming, etc.
  - Focus on making the synchronization pattern(s) clear
- Not looking for a reusable/extensible/overcomplicated solution
  - Just design to the given specification