Assignment 4/5

Alexandre David
1.2.05
adavid@cs.aau.dk

Overview

- OpenMP
  - Easy matrix multiplication.
- Continue on matrix inversion case-study.
  - More fine-grained synchronization.
  - Use of condition variables.
  - Use of split-barrier.

Finer Data Dependency Analysis

- Explained in the assignment, you need to spend time to understand it.
  - 1st loop – pivot.
    Max needs only column k of elements (below row k).
    Swapping needs only 2 rows.
    Division step depends on the row k (after swapping).
  - 2nd loop – back-substitution.
    The next iteration depends on the last row of the current iteration.

Todo

- Use owner-compute rule: associate each row to a thread. *Always the same thread*.
- Split computations to extract these dependencies.
- 1st loop: use a condition variable. The trick is that the threads do not know what will be needed, so we use ok=1 (1st element) and ok=2 (whole row) to say what is ready.
  - The right thread will wait for what it needs only.

Todo

- 2nd loop – Use the concept of interleaving computation and communication with a split-barrier.
  - The thread responsible for the last row arrive at the barrier when it has computed it – the others just arrive.
  - All threads wait before the next iteration.
  - Synchronization is more loose: All threads do not need to wait at the same time.

Optional Todo

- We can replace the split-barrier by a finer condition variable.
  - In fact to proceed to iteration k, we need the row k. Use “current k ≥ ready k” as the condition – the k is decreasing.
  - One thread will update the “ready k”.