Efficient Math Algorithms with pthread library LWPs

Jan Peleska
peleska@uni-bremen.de

2018-10-30

A sequential matrix multiplication program. Consider the program `sequential` which has been provided as zip-archive `sequential.zip`. The program creates first a `dimM` $\times$ `dimN` matrix A of `double` values with `dimM` rows and `dimN` columns. Next, a loop is executed with `numMultiplications` cycles. In each cycle,

1. A `dimN` $\times$ `dimM` matrix B is filled with random `double` values.
2. The matrix product $C = A \cdot B$ is calculated.

The task: parallelisation. Create a new program `threads` which performs the same task as program `sequential`, but with a hight degree of parallelisation and compare the execution times. For designing program `threads`, please follow these guidelines.

1. Create $k+1$ LWPs using the pthread library, with thread id's numbered from 0 to k.
2. Thread k Creates the initial Matrix `A` and the new matrices `B`.
3. Thread $i,\ i = 0, \ldots, k-1$, calculates the lines $j \cdot k + i,\ j = 0, 1, \ldots$ of matrix `C` $=$ `A` $\cdot$ `B`.
4. While the multiplication is being performed by the LWPs $1, \ldots, k-1$, LWP k prepares the next matrix `B`. In order not to interfere with the other threads, two versions of `B` are managed, with a pointer pointing

to the actual version to be used in the multiplication. LPW k writes to the `B`-version which is currently not used by the threads $0, \ldots, k-1$.

5. Create a suitable synchronisation mechanism where the threads $0, \ldots, k-1$ indicate to thread k that the multiplication is finished and thread k indicates to the other threads that the new version of `B` is now ready, identified by the actual pointer value.

Evaluation. Compare the execution time of `sequential` to the execution time of `threads` for different values of k and give an explanation for the optimal value of k.

Return solution via email (C-Code and Makefile) to peleska@uni-bremen.de before the tutorial session on 2018-11-22.