Parallel Computation: Models And Methods

Selim G Akl

If we perform the computations only in the mesh nodes, the postprocessing time takes some hundredths of percent of the total simulation time. To provide the compatibility with COMSOL format a parser was developed; it works with m-le representation of COMSOL model and extracts the information for creating the model. In the presented model dynamic focusing method is used to calculate the unknown potential of the middle electrode using the known potentials of the upper and the lower electrodes and potential equality constraint for middle electrodes separated by dielectrics. The parallel simulation is performed with even distribution of parametric variants on cluster nodes. The results are presented in the table 2.
Analysis of Parallel Execution Times. Parallel Computational Models. Exercises for Chap. 4. To use parallel computers or cluster systems, the computations to be performed must be partitioned into several parts which are assigned to the parallel resources for execution. These computation parts should be independent of each other, and the algorithm performed must provide enough independent computations to be suitable for a parallel execution. Chapter 3 considers popular parallel programming models and paradigms and discusses how the inherent parallelism of algorithms can be presented to a parallel runtime environment to enable an efficient parallel execution. An important part of this chapter is the description of mechanisms for the coordination. Three parallel methods (OpenMP, MPI, and OpenACC) are evaluated for the computation of a two-dimensional dam-break model using the explicit finite volume method. A dam-break event in the Pangtoupao flood storage area in China is selected as a case study to demonstrate the key technologies for implementing parallel computation. The subsequent acceleration of the methods is also evaluated. The simulation results show that the OpenMP and MPI parallel methods achieve a speedup factor of 9.8— and 5.1—, respectively, on a 32-core computer, whereas the OpenACC parallel method achieves a speedup factor. Local computation in graphical models. Parallel methods for local computation. Monte Carlo simulation. Introduction. Parallel MCMC is the topic of Section 18.4. There are two essentially different strategies which can be used for parallelising an MCMC scheme (though these may be combined in a variety of ways). One is based on running multiple MCMC chains in parallel and the other is based on parallelisation of a single MCMC chain. There are different issues related to the different strategies, and each is appropriate in different situations.