Abstract:

As high-performance computing is starting to reach exascale and the amount of interconnected nodes of supercomputers is ever increasing, parallel applications must be extremely scalable, a property which is limited by the degree of parallelization the used algorithms provides, but which can also suffer greatly from load imbalances and processor downtimes due to the delay of data movement. For parallel runtime systems, it is highly desirable to promote the overlap of communication and computation, to evenly distribute load and to aid programmers in designing applications which exploit the given hardware as best as possible. This work presents the different notions and paradigms used by the parallel runtime systems UPC++, Charm++ and AMPI in order to achieve these goals. An overview of previous benchmarks is given, which already show significant performance increases in some applications by leveraging the features of the respective parallelization system. The Shallow Water Equations model will be introduced and the discretization, domain decomposition and parallelization strategy for implementing it will be elaborated on. The actual implementations using each of the aforementioned frameworks are described, which
differ only in the way data is being communicated. Finally, the performance benchmarks of each implementation are illustrated, showing that all runtime systems are able to consistently perform on an equivalent level on par with the MPI reference implementation.

Übersetzter Abstract:
As high-performance computing is starting to reach exascale and the amount of interconnected nodes of supercomputers is ever increasing, parallel applications must be extremely scalable, a property which is limited by the degree of parallelization the used algorithms provides, but which can also suffer greatly from load imbalances and processor downtimes due to the delay of data movement. For parallel runtime systems, it is highly desirable to promote the overlap of communication and computation, to evenly distribute load and to aid programmers in designing applications which exploit the given hardware as best as possible. This work presents the different notions and paradigms used by the parallel runtime systems UPC++, Charm++ and AMPI in order to achieve these goals. An overview of previous benchmarks is given, which already show significant performance increases in some applications by leveraging the features of the respective parallelization system. The Shallow Water Equations model will be introduced and the discretization, domain decomposition and parallelization strategy for implementing it will be elaborated on. The actual implementations using each of the aforementioned frameworks are described, which differ only in the way data is being communicated. Finally, the performance benchmarks of each implementation are illustrated, showing that all runtime systems are able to consistently perform on an equivalent level on par with the MPI reference implementation.

Stichworte: SWE; UPC++; Charm++; MPI; HPC; AMPI; OpenMP

Fachgebiet: DAT Datenverarbeitung, Informatik

DDC: 000 Informatik, Wissen, Systeme

Aufgabensteller: Bader, Michael

Betreuer: Pöppl, Alexander; Samfass, Philipp

Jahr: 2018

Quartal: 4. Quartal

Jahr / Monat: 2018-10

Monat: Oct

Seiten: 32

Sprache: en
de

Hochschule / Universität: Technical University of Munich

Fakultät: Fakultät für Informatik

Occurences: · Einrichtungen > Fakultäten > Fakultät für Informatik > Lehrstühle der Informatik > Informatik 5 - Lehrstuhl für Wissenschaftliches Rechnen (Prof. Bungartz) > 2018
Entries: