Karlsruher Institut für Technologie
Fakultät für Informatik

Lehrstuhl Algorithm Engineering
Prof. Dr. Peter Sanders

Master's Thesis (en/de)

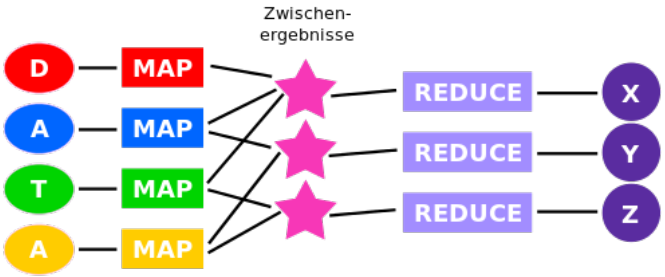
Scalable Fault-Tolerant MapReduce

Description

MapReduce is a programming model for processing large data sets with a parallel, distributed algorithm on a high performance computer (HPC). It consists of two main functions: the Map function that processes a set of data into intermediate key-value pairs, and the Reduce function that aggregates the intermediate values based on their keys to produce the final output. This allows for efficient and scalable parallel processing of large amounts of data, as the Map and Reduce tasks can be distributed across multiple nodes in a cluster.

With the increasing number of processors in high performance computing clusters, the probability that some processors fail during a computation rises. Handling such failures constitutes a major challenge for future exascale systems. For example ORNL's Jaguar Titan Cray XK7 system had on average 2.33 failures/day between August 2008 and 2010. In upcoming systems, we expect a hardware failure to occur every 30 to 60 minutes. Manually making algorithms fault-tolerant and scalable puts an additional burden on the programmer who might thus be willing to trade a more powerful programming model for automatic fault-tolerance and scalable parallelization – for example by programming in the MapReduce paradigm.

As part of this thesis, an existing fault-tolerant MapReduce implementation is to be improved. This includes implementing and evaluating an efficient hybrid parallelization (shared memory + distributed memory) as well as improving the automatic load balancing – for example by developing a scalable work stealing subsystem. Common MapReduce algorithms extend the MapReduce paradigm to allow static data to be stored by the processes which is part of the messages in each round. By implementing this extension in a fault-tolerant manner, the performance of the MapReduce implementation would improve for a variety of algorithms.



Relevant Skills

- Interest in parallel algorithms and data structures
- Excellent knowledge of modern C++ or Rust
- Experience in implementing distributed algorithms
- Willingness to engage in meticulous scientific work

Lukas Hübner
Informatikgebäude am Fasanengarten, Raum 221, huebner@kit.edu