Hessisches Kompetenzzentrum
für Hochleistungsrechnen

Fault Tolerance and Locality-Aware Work Stealing for Dynamically Generated Dependent Tasks on Clusters

Project Manager
Björn Fries

Principal Investigator
Prof. Dr. Claudia Fohry

Project Term
2024 - 2025

Clusters
Lichtenberg II Cluster Darmstadt

Additional Software
Open MPI, ULFM, OpenMP

Institute
Department of Electrical Engineering,
Department of Computer Science

University
University Kassel

Introduction

Programming environments for today's supercomputers must support the design of efficient programs and handle issues such as: programmer productivity, i.e., the human effectiveness in writing programs; application irregularity, i.e., the limited planability of the computation; and fault tolerance, i.e., the ability to cope with hardware failures during program execution.

While these issues are hard to achieve with traditional parallel programming environments, a promising paradigm to tackle all of them together is Asynchronous Many-Task (AMT) programming. Here the computation is coded into many small work packages (tasks), which may depend on each other. The tasks are processed by a limited number of workers (e.g. processes), which often balance their load via work stealing. AMT programming environments enjoy growing popularity on single multicore computers, where they provide powerful functionalities such as dynamic task generation at runtime to facilitate the expression of irregularity, and load balancing via work stealing to improve resource utilization.

On large supercomputers, i.e., clusters of such machines, in contrast, AMT functionalities are still limited. A major hurdle for AMT deployment there is the need to combine load balancing with low communication costs. In particular, tasks should run close to their data, which is denoted as locality. Similarly, fault tolerance takes on crucial importance in clusters, due to their larger number of hardware components. Although AMT is potentially well-suited to provide fault tolerance, concrete algorithms and techniques are still rare.

Methods

We are designing algorithms and other techniques towards the above goals. For experimentally evaluating these, we have developed an own prototypical cluster AMT programming library named ItoyoriFBC that supports dynamic task generation at runtime, global work stealing, and the future construct. In this context, we used the Lichtenberg High-Performance Computing (HPC) cluster for a subgoal towards the realization of fault tolerance, namely the implementation of a resilient store. Our store is based on one-sided MPI communication and achieves fault tolerance with the User-level failure mitigation (ULFM) extension of MPI. We used the machine to experiment with several design options, settings, and software versions during the store design, as well as to test the usability of the store within our AMT library.

Results

Usage of ULFM in the context of one-sided MPI communication proved difficult, because of limited support in the current Open MPI implementation. We identified various specific problems and possible solutions. Moreover, we implemented a resilient store that is functional, but has several shortcomings.

Discussion

Usage of the Lichtenberg Hochleistungsrechner was helpful, since it allowed us to experiment in another hardware/software environment than what we have available locally and at another supercomputing site that we have access to. Within the project period, we only needed this opportunity for the specific aspect of the resilient store design, but generally our research includes experimental investigations on portability and performance portability, which require runtime measurements on different scales and architectures. For the near future, for instance, we plan to evaluate the performance and scalability of various locality optimization techniques for our AMT system. Regarding the specific topic of the resilient store, we are currently working on alternative designs that are based on two-sided MPI functions, and another communication library, respectively, and plan to experimentally assess them afterwards.

Publications

C. Fohry; R. Fink: User Experiences with MPI RMA and ULFM in a Resilient Key-Value Store Implementation (Poster). Euro-MPI/USA (2025)

Reference

M. Reitz; J. Hundhausen; C. Fohry: Fail-stop Failure Protection for Coordinated Work Stealing of Tasks that Communicate Through Futures; Proceedings Workshop on Asynchronous Many-Task Systems and Applications (WAMTA) pp. 44–55 (2025)

Last Update: 2026-02-02 11:21