

The AllScale API and Programming Environment

Herbert Jordan, Philipp Gschwandtner

*Distributed and Parallel Systems Group, University of Innsbruck
Technikerstrasse 21a, 6020 Innsbruck, Austria
{herbert,philipp}@dps.uibk.ac.at*

The tremendous challenge of efficiently developing applications that utilize the hardware provided by contemporary parallel systems of all scales is among the most limiting factors for the continuous growth of high performance computing. The number of components of parallel computing systems, their complexity and especially the increasing degree of parallelism on multiple layers has reached a level that is becoming prohibitively costly to manage. Current HPC design paradigms in industry rely on well-established parallel libraries and/or language extensions addressing specific hardware resources (e.g. MPI on the inter-node level, OpenMP on the intra-node level). However, this imposes severe issues on the software design of HPC applications, including limited composability and maintainability of software components, hard-coded problem decomposition, and increased management responsibilities of the user. All these issues impair the productivity of application developers as well as the (performance) portability of their applications.

In this tutorial, we present a novel architecture taking on this challenge by providing an infrastructure for the effective development of such applications. Our design combines the expressive power of modern C++, advanced compiler technology, and sophisticated runtime system solutions, providing a clean separation of domain-specific algorithms, resource management activities, and low-level hardware interactions. The programming interface to this architecture is offered via a single C++ API that provides both data structures and parallel operators in order to implement algorithms on a high level of abstraction, with inherent support for both shared and distributed memory parallelism. Our approach leverages the advantages of nested recursive parallelism in order to dynamically adapt to a variety of target architectures and overcome the limitations of flat parallelism.

This tutorial consists of two parts. First, a talk covers the architecture design, its key aspects and research challenges. The second part demonstrates the power of the AllScale environment by porting a 2-D heat diffusion application in multiple steps from a naive sequential implementation in C/C++ to the AllScale API, improving usability and performance along the way.

References

- [1] <http://www.allscale.eu/>
- [2] <https://github.com/allscale>

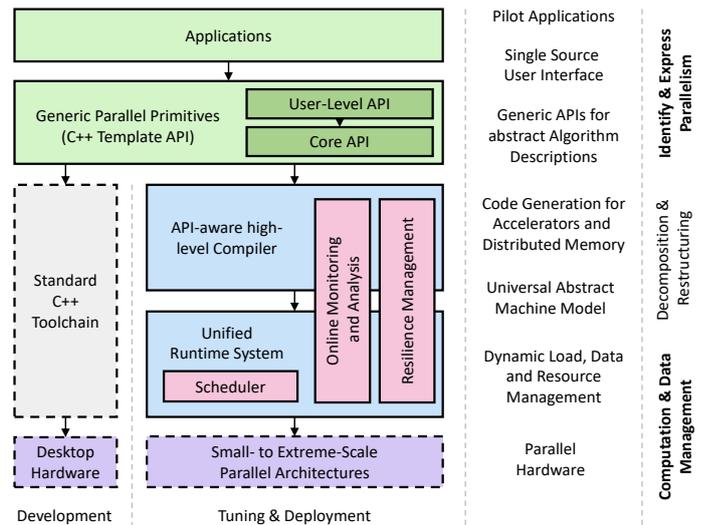


Fig. 1: AllScale environment components.

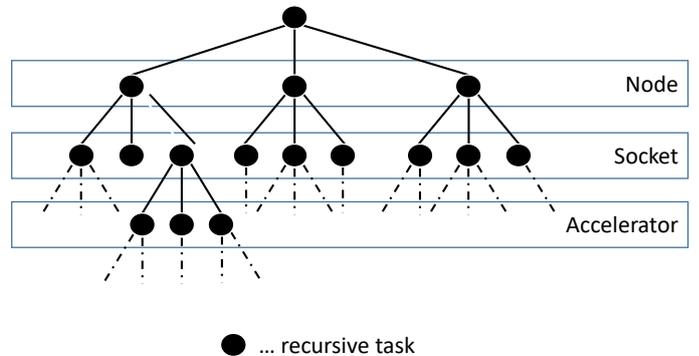


Fig. 2: Nested recursive parallelism.