

Performance Portability with alpaka

Jan Stephan, Michael Bussmann
CASUS // Computer Science



Motivation

Exascale is the next big step in the field of high-performance computing (HPC). However, the hardware configurations of supercomputers are becoming more and more heterogeneous. Programmers have to adapt to different processor architectures (x86, ARM) as well as various accelerator types (GPUs, FPGAs) and their accompanying tools. Our goal is a stack of programming libraries which form a performance-portable ecosystem.



Challenges

The main challenge is to find a way to easily express the problem in an abstract and user-friendly fashion. These expressions then need to be turned into hardware-aware algorithms and data structures as they are passed down to the lower levels. Additionally, we need to provide users with a detailed performance analysis of the various ecosystem layers both at runtime and after program execution.

$$\vec{y} \leftarrow a\vec{x} + \vec{y}$$

State of the Ecosystem

alpaka – parallel kernel programming library – regular release schedule

LLAMA – memory access library – ongoing research project

vikunja – high-level algorithms library – first release available

bactria – performance analysis library – work in progress

mallocMC – on-device memory allocator – ready to use

RedGrapes – task-graph library – ready to use

Hardware Support Library – planned

Memory Allocation Library – planned

Methods

Our ecosystem utilizes the template metaprogramming capabilities of ISO C++17. This comes with several benefits:

- The reliance on an ISO standard ensures portability between compilers.
- The access to C++ template metaprogramming facilities resolves the necessary abstraction layers at compile-time. At runtime this results in performance that is extremely close to native programming models.
- The utilization of modern C++ hides the verbose implementation details behind a concise programming interface.

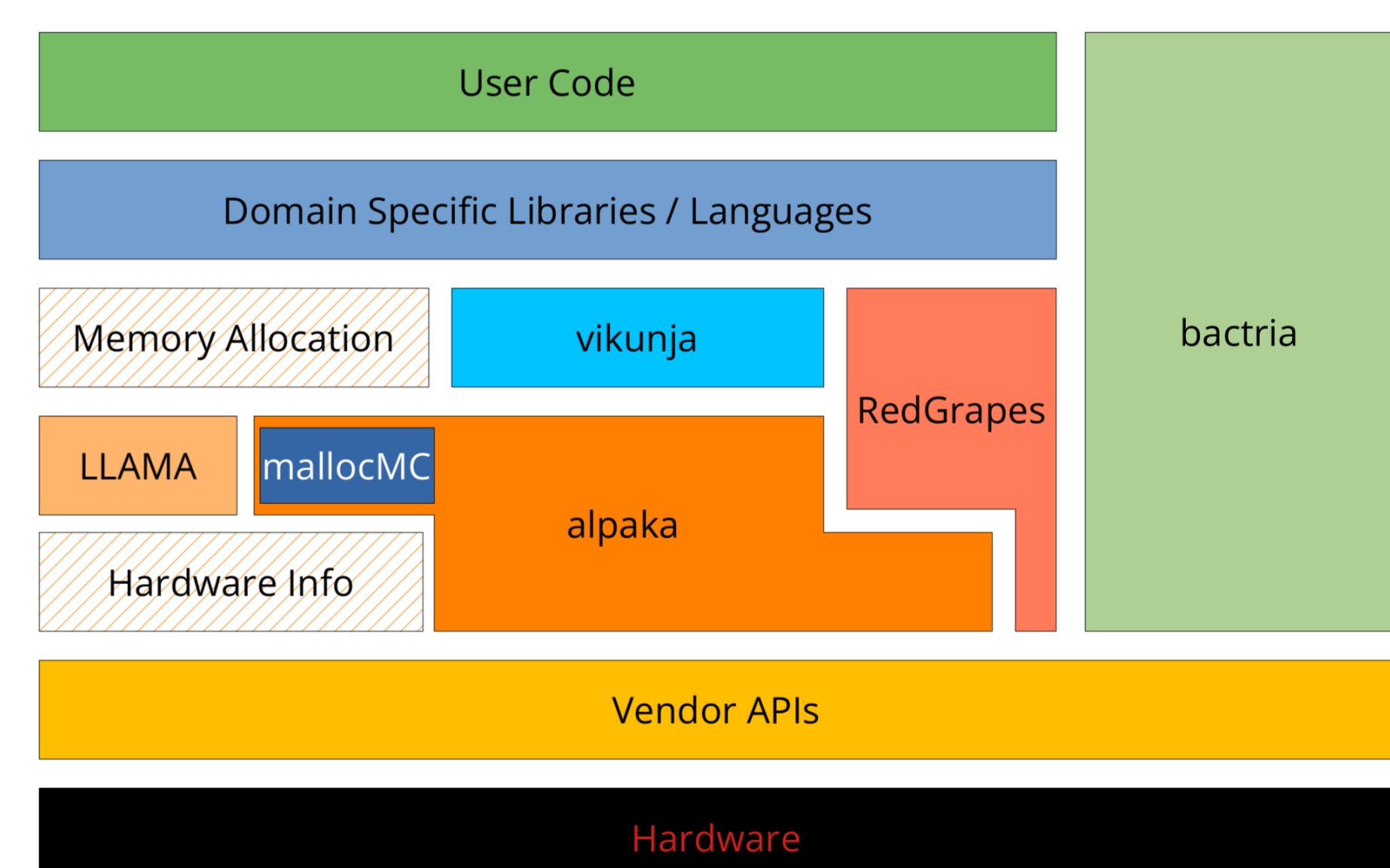
Modern C++ contributes to our goal of performance-portable software.

Future Goals & Applications

In the future we aim to improve the integration of our individual libraries without introducing any hard interdependencies. An envisioned set of shared abstract concepts will help us in this matter.

Our work will provide scientists from various fields of research with an Exascale-ready and user-friendly ecosystem which they can easily use for developing performance-portable applications. In addition, our performance analysis library *bactria* will enable users to gain in-depth knowledge about their programs' behaviour – both offline and in real time.

Performance Portability is the future of high-performance computing.



Strategic Impact

Our *alpaka* library for parallel kernel programming is used by important HZDR-internal projects (such as PIconGPU) as well as multiple groups within Europe, such as the CERN-CMS experiment and the German aerospace agency DLR. In addition, we provide trainings for our ecosystem, for example with an one-week virtual workshop for CERN in 2020 or as lecture for the International School of the Italian Institute for Nuclear Physics (INFN) in 2021.

Thanks to the increasing adoption of our libraries the number of external contributors is growing which further advances our project.

