Lattice QCD on upcoming ARM architectures
Nils Meyer, Dirk Pleiter, Stefan Solbrig, Tilo Wettig
"Department of Physics, University of Regensburg, Germany
Forschungszentrum Jülich, Germany

QPACE 4 project

- Aims to design an ARM SVE environment optimized for Lattice QCD
- Prototype at the University of Regensburg, Germany, by 2020
- Pursues evaluation and enhancement of existing SVE software toolchain and upcoming SVE hardware technologies for Lattice QCD applications
- Enables Lattice QCD code optimized for SVE architectures, e.g., the Japanese flagship supercomputer "Post-K" announced for 2021

ARM SVE

- The ARM Scalable Vector Extension (SVE) targets the HPC market [1]
- Key features of SVE hardware
  - Wide vector units, ranging from 128 bit to 2048 bit
  - Vectorized native 16 bit floating point operations, including arithmetics
  - Vectorized arithmetics of complex numbers
  - The silicon provider chooses the vector register length and defines the performance characteristics of the SVE hardware
- The first SVE hardware officially announced is the Post-K, which comprises 512 bit wide vector units, 48 compute cores per node and high performance stacked memory [2]
- Key features of the SVE instruction set architecture (ISA)
  - SVE follows a vector-length agnostic (VLA) programming model that adapts itself to the available vector length (VL)
  - VLA predication allows for selection of vector elements to be used for processing, which enables, e.g., complex control flows within loops
  - Support for structure load/store, e.g., load of an array of two-element structures into two vectors, with one vector per structure element
  - The ARM C Language Extensions for SVE (ACLE) intrinsics provide convenience access to features of SVE hardware in C/C++ [3]

SVE code development tools

- Compilers
  - ARM provides us with their LLVM/clang-based armclang 18 compiler (evaluation version available [4])
  - RIKEN provides us with the Fujitsu SVE compiler
- We test the compilers, stimulate bug fixes and propose improvements in the context of research contracts
- ARM Instruction Emulator (ArmIE)
  - ArmIE allows for functional verification of SVE binaries emulating the SVE ISA with user-defined VL (freely available [4])
  - ArmIE is designed on top of the open-source DynamoRIO toolset enabling code instrumentation at run-time [5]
- We contribute to DynamoRIO enabling advanced SVE code analysis, e.g., instruction mixing and branching statistics (≈ Lattice 18 proceedings)
- gem5
  - The gem5 simulator is a modular platform for computer architecture research, encompassing system-level and processor architecture [6]
  - RIKEN provides us with access to their gem5 simulator of the Post-K CPU
- We optimize our code guided by simulations of the Post-K CPU (future work)

"Grid" Lattice QCD framework

- Grid [7, 8] is a portable open-source Lattice QCD framework written in C++ II, maintained by Peter Boyle (Edinburgh) and co-workers
- Targets massively parallel architectures supporting SIMD + OpenMP + MPI
- Implements more than 100 tests and benchmarks
- ISA-specific code is implemented using intrinsics and assembly

Enabling SVE in Grid

- We use ACLE for our SVE implementation to access the SVE features
- We minimize optimizations of the VLA programming model and bypass restrictions on usage of ACLE data types
  - We declare the vector length VL (in bytes) as a compile-time constant
  - Superfluous loops implied by VLA are omitted
- The variety of predication is minimized to fit into the register file
- The templated struct acle<T> (T = double, float) provides convenient access to ACLE definitions, e.g., predication and data types
- Code example: templated multiplication of two vectors of complex numbers

References

[8] https://github.com/paboyle/Grid