Lattice QCD on a novel vector architecture

Benjamin Huth, Nils Meyer, Tilo Wettig

Department of Physics, University of Regensburg, Germany

NEC SX-Aurora Tsubasa

- The SX-Aurora Tsubasa is the newest member of the NEC SX-series [1]
- PCIe card with gen3 x16 interconnect called vector engine (VE)

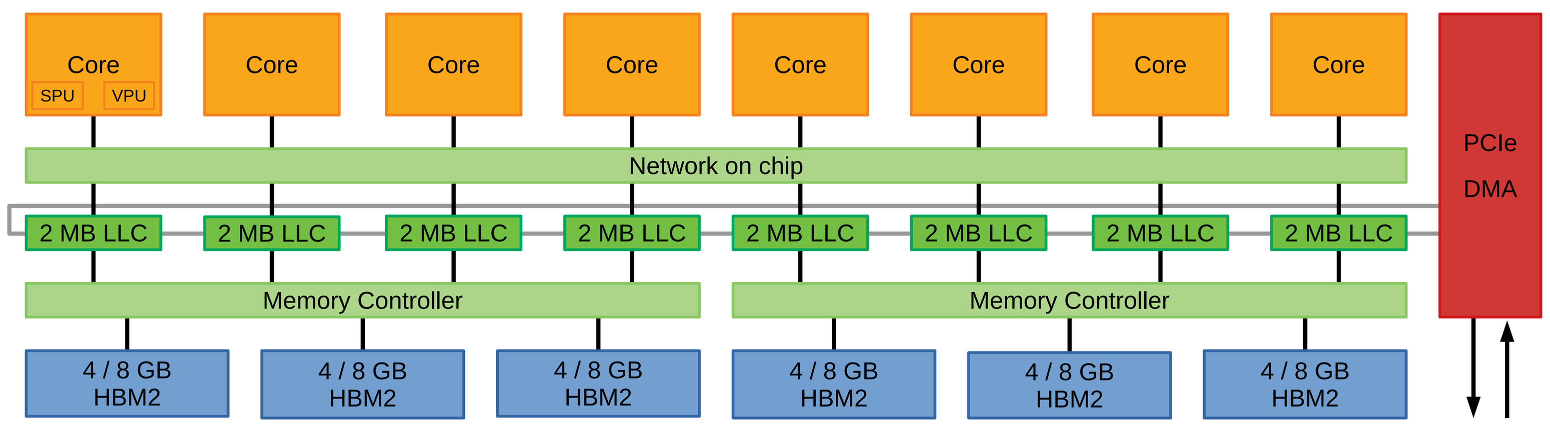
Vector engine model	Type 10A	Type 10B	Type 10C
Clock frequency [GHz]	1.6	1.4	1.4
SP/DP peak performance [TFlops/s]	4.91/2.45	4.30/2.15	4.30/2.15
Memory capacity [GB]	48	48	24
Memory throughput [TB/s]	1.20	1.20	0.75

- Multiple platforms are available, ranging from workstation to supercomputer [1]
- Up to 64 vector engines interconnected by InfiniBand fit into one NEC A500 rack, delivering 157 TFlops/s peak in double precision

Picture published with permission from NEC, © by NEC

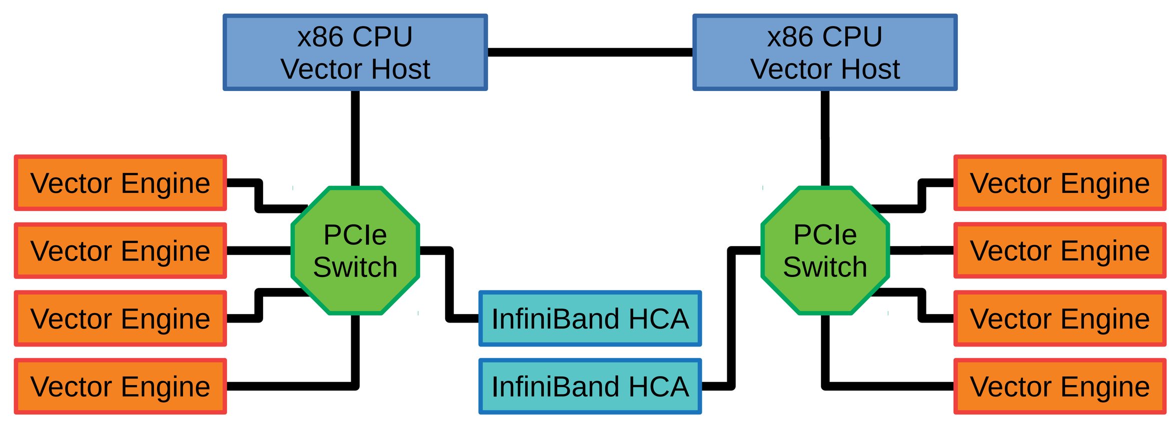
Vector engine architecture

- Novel vector architecture with vector registers of 16 kbit size each [1, 2, 3]
- Vector processor consists of 8 single-thread out-of-order cores, each with a
  - scalar processing unit (SPU) with L1 and L2 caches
  - vector processing unit (VPU), which processes the (optionally masked) 16-kbit vector registers in 8 chunks of 2 kbits each
  - coherent last-level cache (LLC) directly accessible by the VPU
- 6 HBM2 stacks in-package
- 2d mesh network on chip for memory access
- Ring bus for direct memory access (DMA) and PCIe traffic



System setup and programming models

- An Intel x86 host CPU is necessary, called vector host (VH)
- Up to 8 VEs connect to a dual-socket VH using PCIe switches [1, 3]



- Programming models [4]

Model	Description
VE execution	The program is executed on the VEs, system calls are offloaded to the VH.
VE offload	The program is executed on the VH, parallelization-friendly parts are offloaded to the VEs.
VH offload	The program is executed on the VEs, scalar-friendly parts and system calls are offloaded to the VH.

Software stack

- Proprietary software by NEC [1]
  - Linux kernel modules and user space daemons for VE operation
  - C, C++ and Fortran VE compilers support auto-vectorization and OpenMP, but do not support intrinsics
  - Profiler and debugger
  - Optimized math libraries including BLAS and LAPACK
  - MPI libraries compliant to MPI-3.1 specification
- LLVM VE backend is under development [5]
  - Vectorization must be guided by pragma directives
  - OpenMP and intrinsics support

Grid on the NEC SX-Aurora Tsubasa

- Grid Lattice QCD framework [6]
  - Targets massively parallel architectures supporting SIMD + OpenMP + MPI
  - Grid is optimized for, and restricted to, up to 512-bit SIMD registers
  - Need low-level extensions of Grid to support the NEC SX-Aurora Tsubasa
- Extending Grid to registers with $2^n \times 128$ bits
  - Due to implementation of shift/stencil operations, current maximum SIMD layout in Grid is {2,2,2,1} (sites per dimension, complex single precision)
  - For SX-Aurora we need, e.g., {4,4,4,4} → Implement shift/stencil with new `splitRotate` transformation, replacing original `permute`-based implementation

```
splitRotate(m, r, vector)
  split vector in m equal parts
  rotate each part by r
```

– All dimensions/directions possible
– Vectorizes with good performance

- Illustration: shift on 2d lattice with real numbers
  - Lattice volume is {8,4}, SIMD width is 8 real numbers with layout {4,2}
  - Data layout of lattice

0	8	1	9	2	10	3	11
16	24	17	25	18	26	19	27
4	12	5	13	6	14	7	15
20	28	21	29	22	30	23	31

Shift 2 to the left →

1	9	2	10	3	11	0	8
17	25	18	26	19	27	16	24
5	13	6	14	7	15	4	12
21	29	22	30	23	31	20	28

  - Data layout in memory

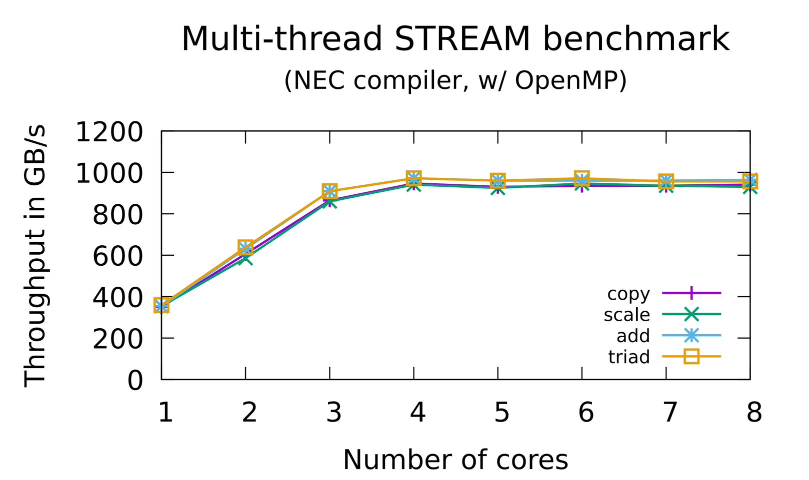
`[0 1 2 3 4 5 6 7]` → `[1 2 3 0|5 6 7 4]`

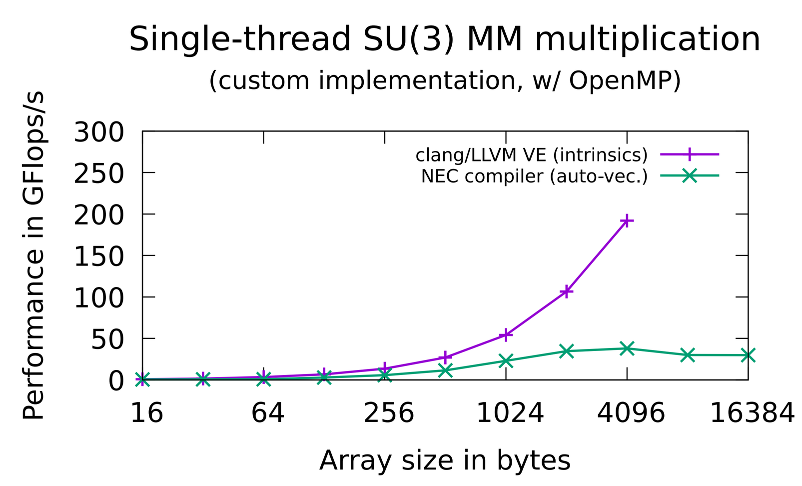
  - Transformation not supported by original `permute`/`rotate`, need `splitRotate`

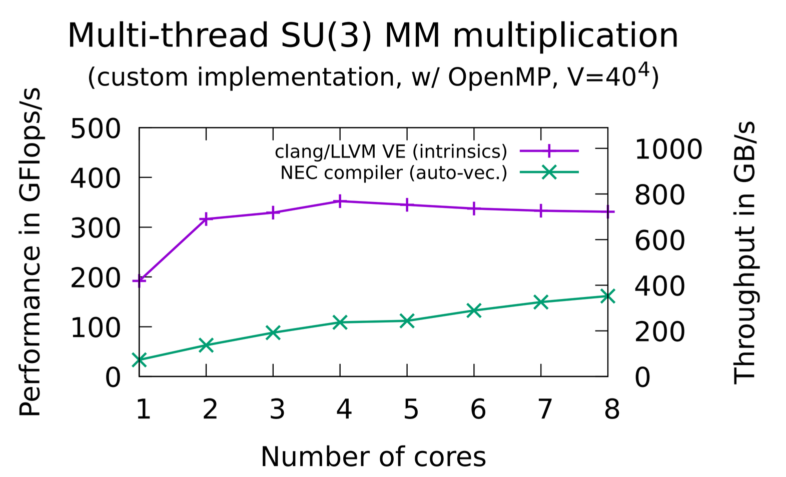
Status of porting Grid to the SX-Aurora

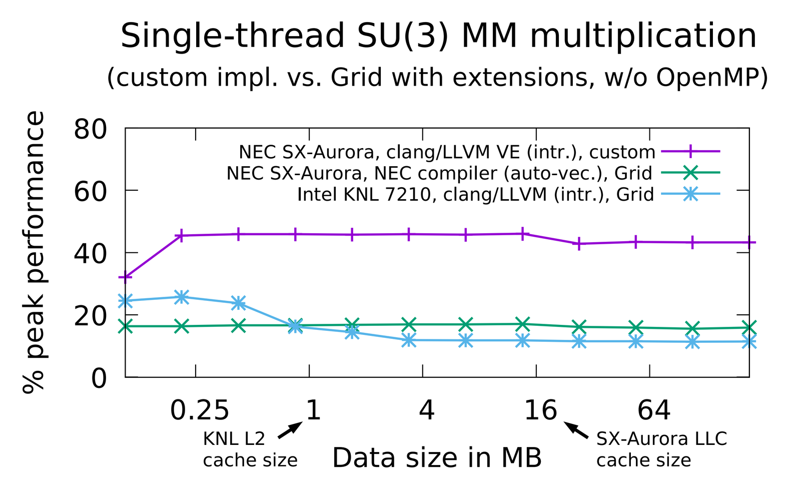
- We pursue the VE execution model, porting Grid to the vector engine
- At present we focus on Grid's generic C/C++ implementation and rely on the NEC C++ compiler's auto-vectorization capabilities
- Pragmas/intrinsics as implementation option only if LLVM VE matures
- Deficiencies of VE compilers slow down progress in development

Preliminary performance evaluation of the SX-Aurora Type 10B









- Optimal array size for processing complex is twice the register size (4096 bytes)
- clang/LLVM VE OpenMP `parallel for` boosts performance on single core
- clang/LLVM VE fails to compile Grid
- NEC compiler fails to compile Grid when enabling OpenMP

Summary and outlook

- NEC SX-Aurora Tsubasa is a novel vector architecture with 16-kbit registers
- Low-level extensions of Grid are required to support this architecture
- We enable any register size of $2^n \times 128$ bits in Grid
- We stimulate bug fixes and suggest improvements to VE compilers

References

[1] https://www.nec.com/en/global/solutions/hpc/sx/

[2] "Vector Engine Processor of NEC's Brand-New Supercomputer SX-Aurora TSUBASA", Y. Yamada and S. Momose, *Hot Chips* (2018)

[3] https://fuse.wikichip.org/news/1833/a-look-at-necs-latest-vector-processor-the-sx-aurora/

[4] K. Komatsu et al., *Proceedings of SC 18* (2018) 54

[5] https://sx-aurora.github.io/

[6] P. Boyle et al., *Proceedings of LATTICE 15* (2016) 023 [arxiv:1512.03487]

Supported by the German Research Foundation (DFG) in the framework of SFB/TRR-55