The NEC SX-Aurora TSUBASA Vector Engine

Tim Cramer

IT Center | RWTH AACHEN UNIVERSITY

Motivation

- Interesting Architecture
- Very high memory bandwidth
  - User codes of the RWTH Compute Cluster are often memory-bound
    → might benefit from SX-Aurora capabilities
- Performance portability: Single application for multiple types off devices / architectures
  - standard-compliance, e.g., MPI, OpenMP

SX-Aurora TSUBASA Vector Engine (VE)

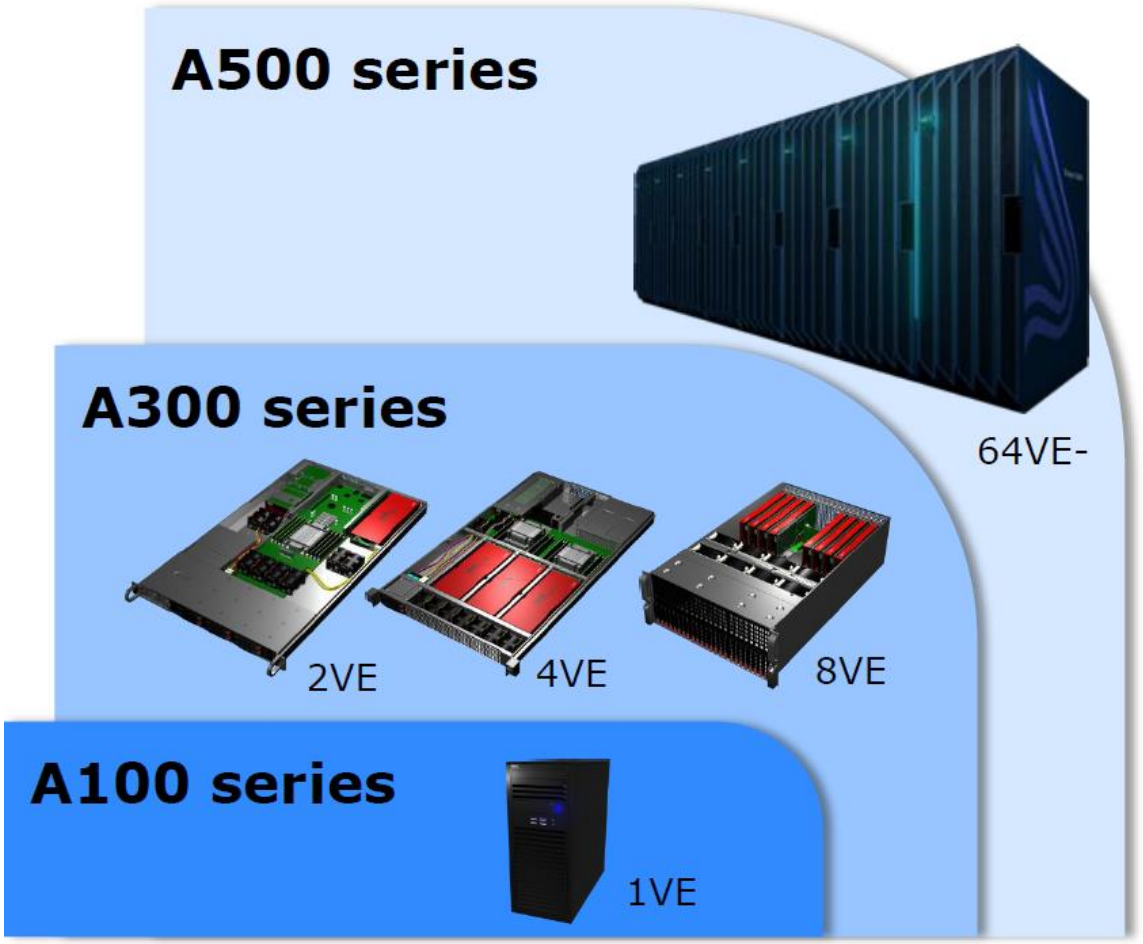
- Vector processor as accelerator in x86 environment
- PCIe card
- Combines SIMD and pipelining
  - vector register length: 256 * 64 = 16384 bits
- OS for the VE (VEOS) runs on the vector host (VH)
- Specs

VE 1.0

Vector Cores	8
FMAs	3
Core Frequency	1.4 – 1.6 GHz
Theor. Peak Performance	307 GFLOPs (DP) 604 GFLOPs (SP)
Memory Capacity	24/48 GB
Main Memory Bandwidth	1.2 TB/s (6x HBM2 stacks)
TPD	300 Watts
Technology	16 nm FinFET process

SX-Aurora in a Cluster Environment

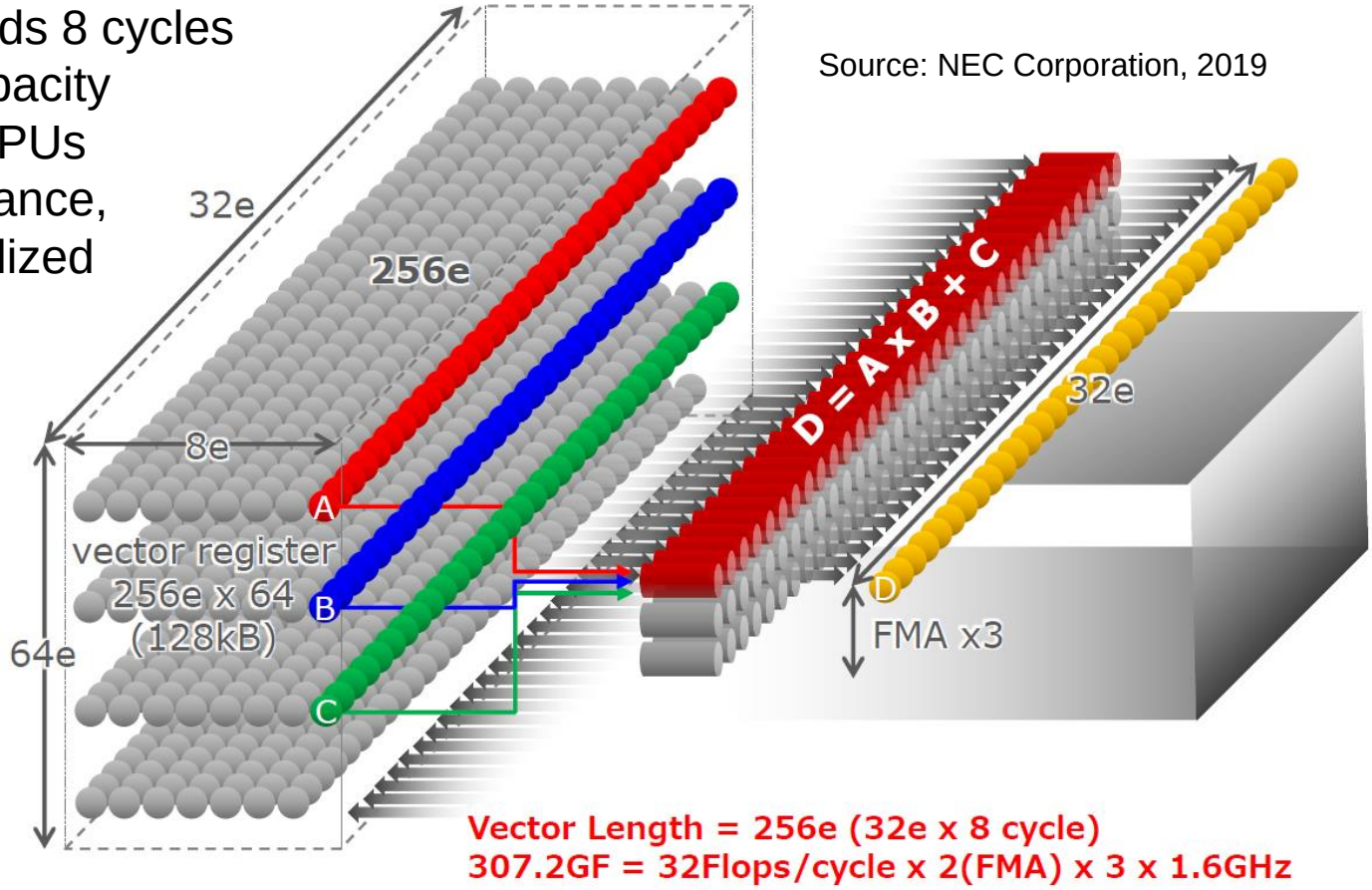
- Supercomputer Model
  - DLC with 40° C
- Rack Mount Model
  - Air cooled
- Tower Model



Source: NEC Corporation, 2019

Vector Execution

- SX-Aurora combines vectors with pipelining
  → vector length = 16384 bit
  → single FMA needs 8 cycles
- Higher register capacity than L1 in many CPUs
- Very poor performance, if FMAs are not utilized



Source: NEC Corporation, 2019

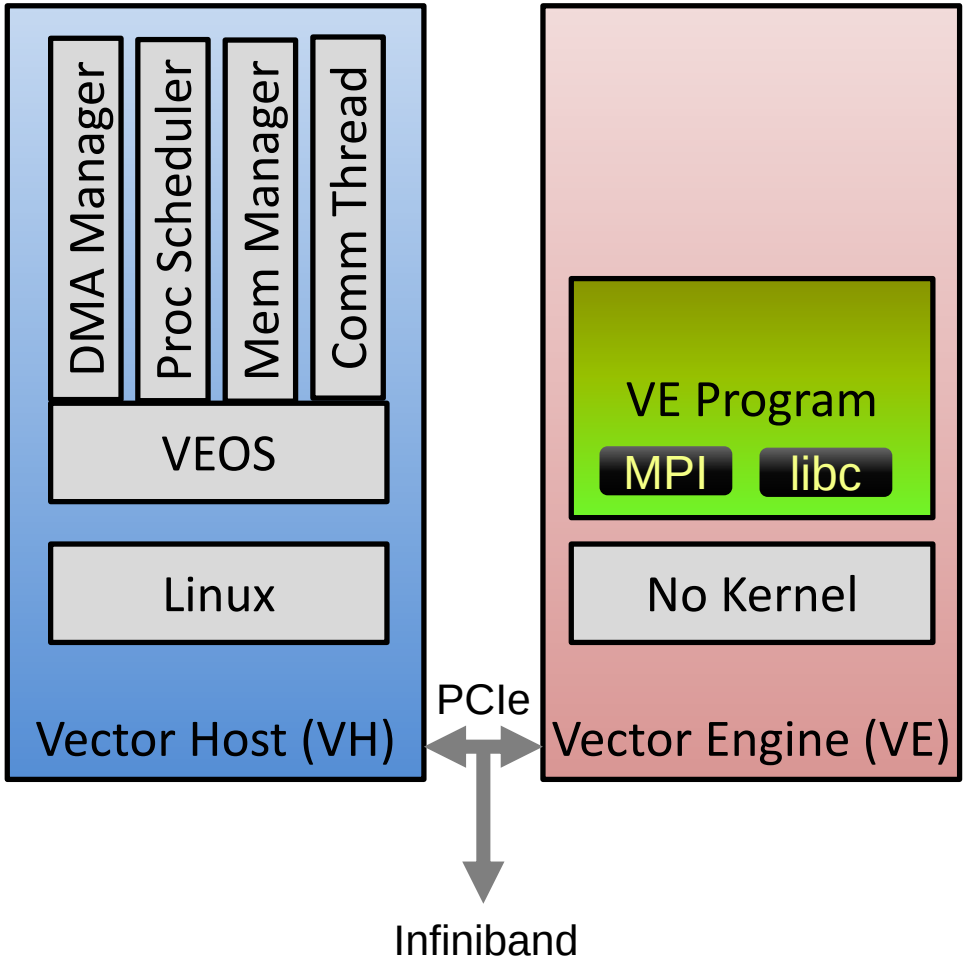
VEOS / Installation

VEOS

- VEOS is running on the host (Linux environment)
- Provides OS, Linux compatible and HPC functionality for VE programs
- Installation via yum repositories

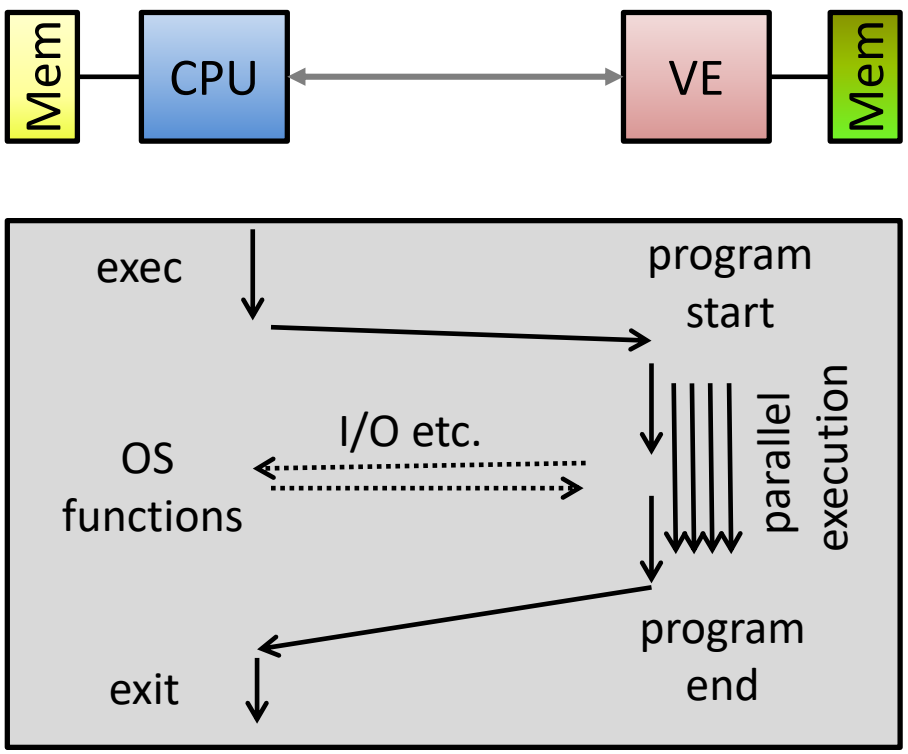
Compiler

- C/C++ and Fortran compiler available in different versions (ncc, nfort)
- Very picky, compatible flags available
- Covers different tools (gdb, top, nm, etc.)
- Limited modules support

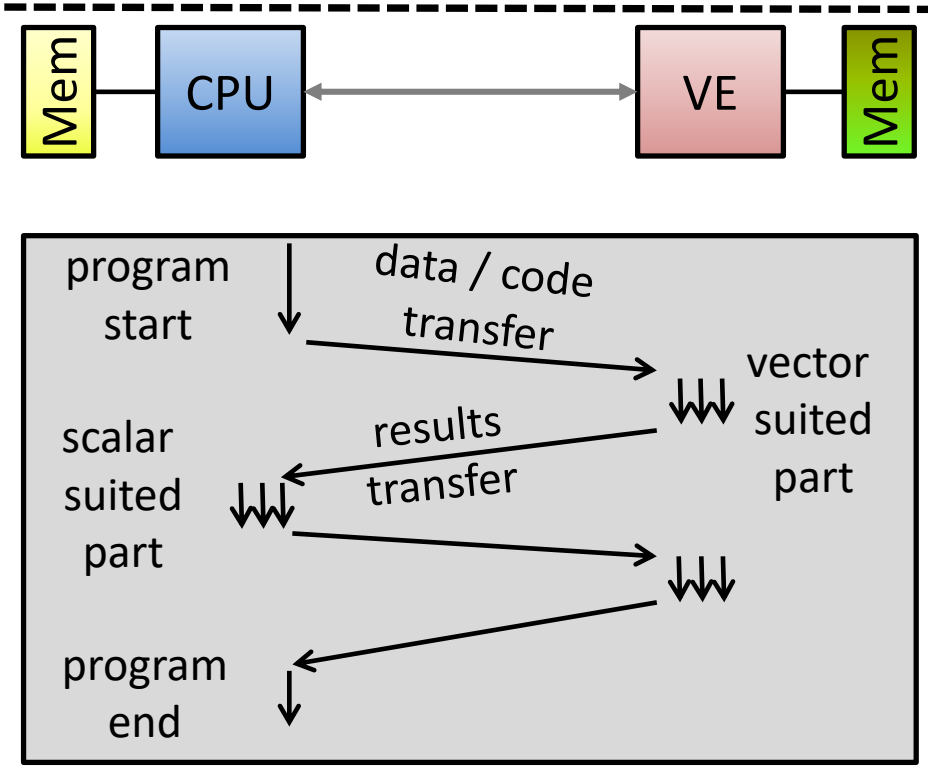


Aurora Execution Models

- Supports native and (limited) offloaded execution
- “Reverse Offloading”: VHCall (limited Fortran support)



Native Execution



Offloaded Execution

Software Packages / MPI

- MPI
  - NEC MPI available (proprietary, MPI-3.1)
  - Hybrid MPI (VE + VH) possible, no experiences in Aachen yet
  - MPI communication happens on VE (in contrast to system calls)

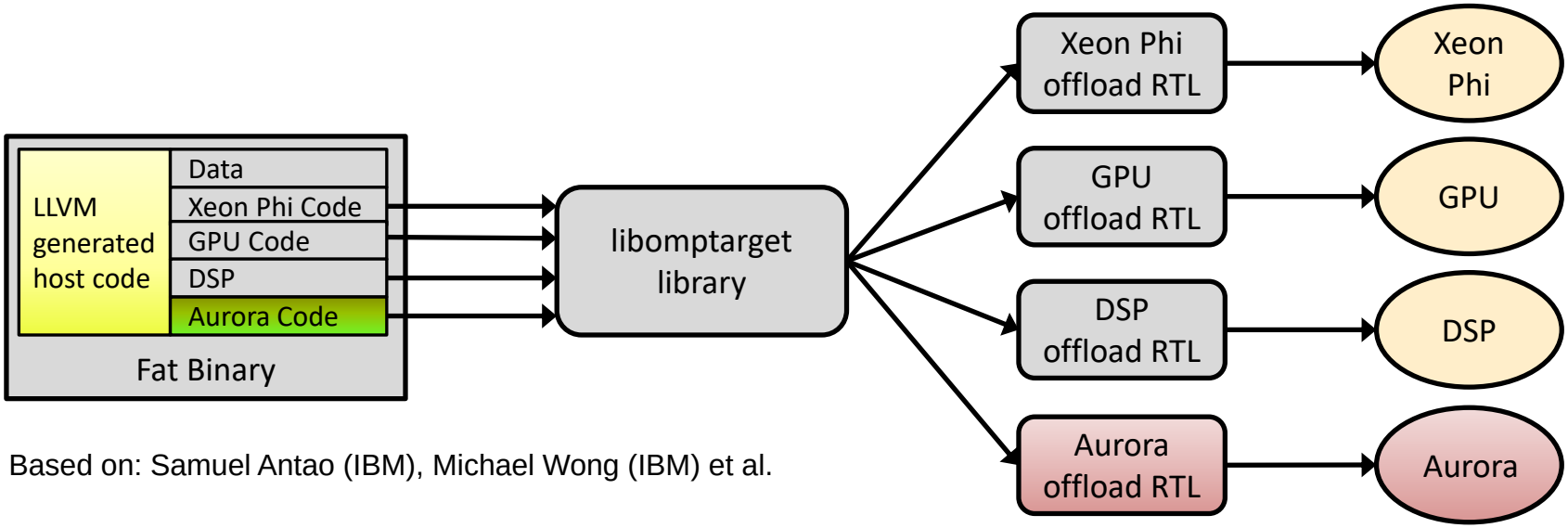
- Numerical libraries
  - BLAS, SBLAS, LAPACK, SCALAPACK, ASL, Heterosolver

SX-Aurora TSUBASA in Aachen

- A300-8: 8 VEs in a single system
- Accessible only for selected users
- Used as dialog system
  - No scheduling right now
  - Slurm plugin available in general
  - Developing, testing, performance analysis
- Although it is a quite specific system, the effort for installation / maintenance is reasonable
- Use cases in Aachen
  - Research / development of an LLVM-based offloading infrastructure
  - ML experiences (bachelor thesis)
  - Porting / Testing of applications
    - QE: Promising application
    - GROMACS: Issues with boost compilation
    - XNS (GRMES dominated): Works, performance is ok, but less than 2 Intel Xeon
    - CIAO: Work in progress
    - CP2K: Work in progress
    - LAMMPS: Some hotspots much faster than on Intel CPU, very expensive initialization

OpenMP Offloading

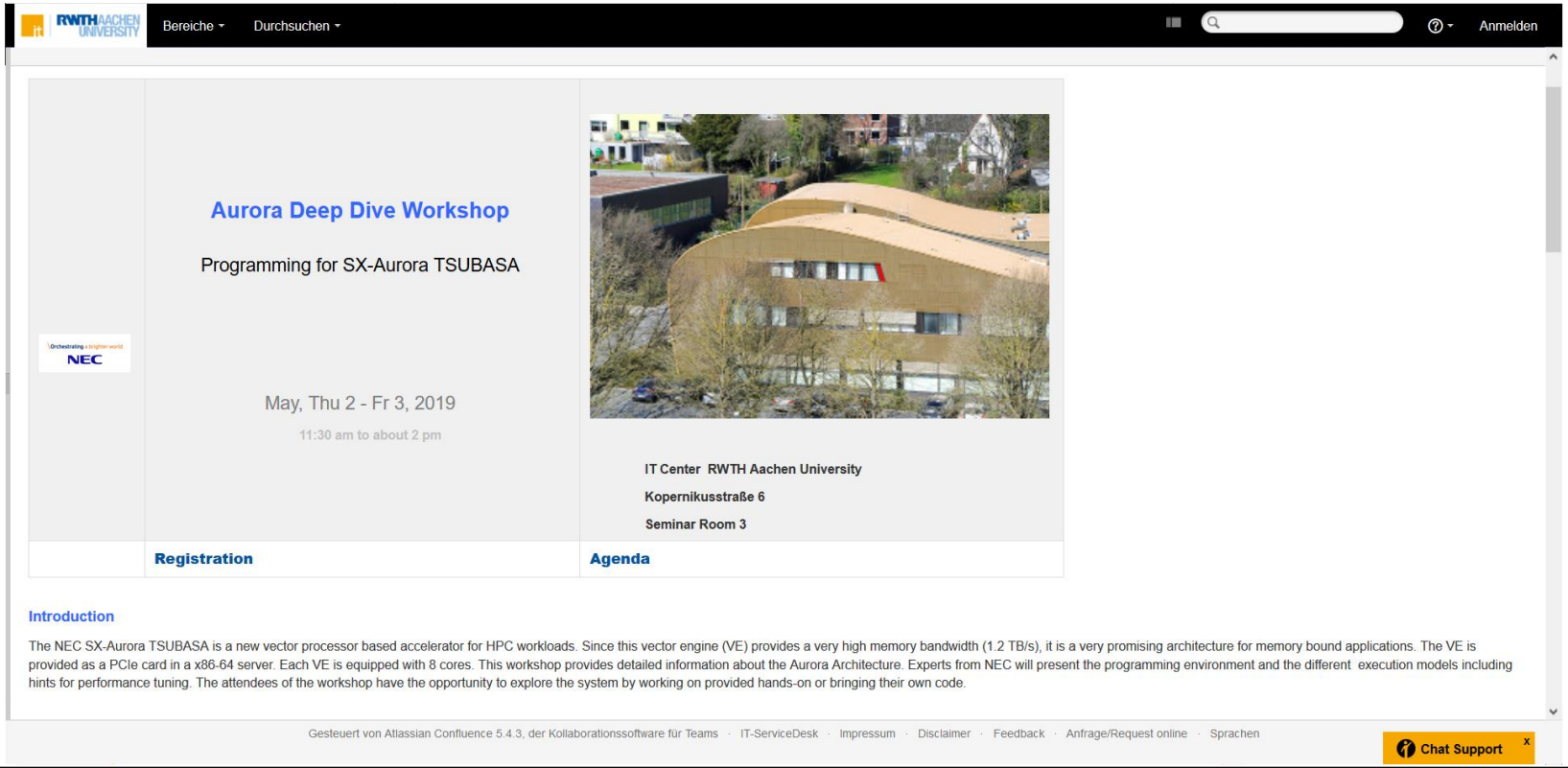
- Not officially supported (yet)
- RWTH Aachen implemented a prototype based on LLVM/Clang and VEO
- Source-2-source transformation for target device code
- `$ clang -fopenmp -fopenmp-targets=`**`aurora-nec-veort-unknown`** `input.c`



Based on: Samuel Antao (IBM), Michael Wong (IBM) et al.

2nd Auroa Deep Dive Workshop

- NEC and RWTH Aachen University organizing a workshop for SX-Aurora
- Please contact me, if you interested



Documentation

- Man Pages
  - Very limited man pages ☹
- Compiler and library documents
  - www.hpc.nec/documents
- Aurora forum
  - www.hpc.nec/forums
- NEC Blog
  - https://sx-aurora.github.io
- RWTH OpenMP Offloading
  - https://rwth-hpc.github.io/sx-aurora-offloading/
  - https://github.com/RWTH-HPC