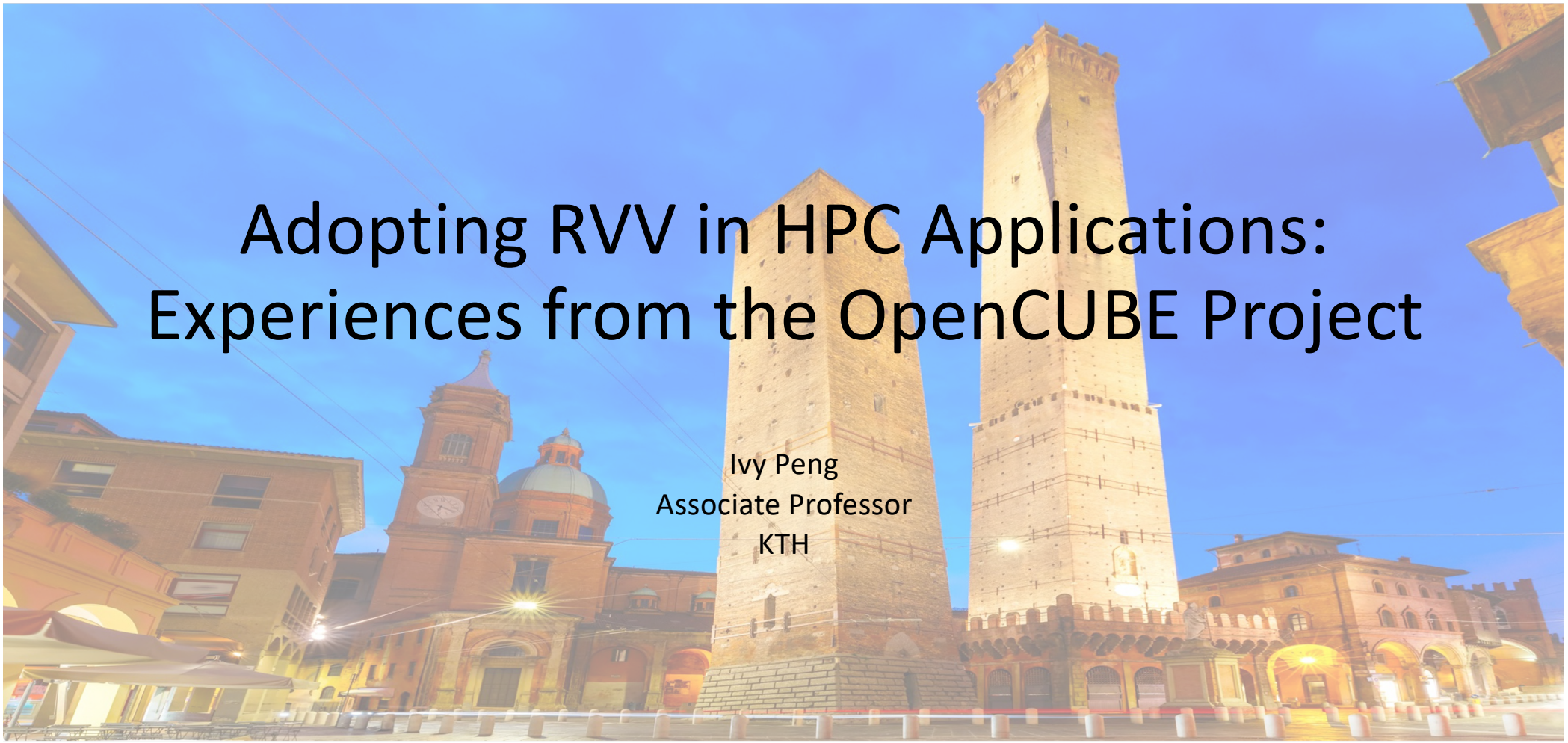


Adopting RVV in HPC Applications: Experiences from the OpenCUBE Project

Ivy Peng
Associate Professor
KTH

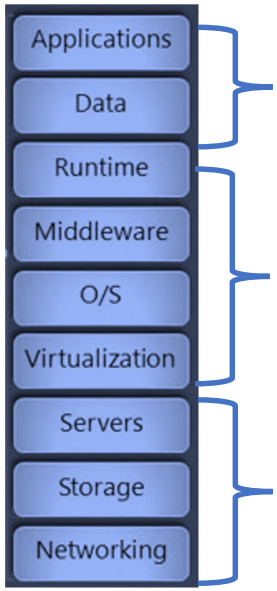
Funded by the European Union
OpenCUBE
Open-Source Cloud-Based Services on EPI Systems (GA 101092984)

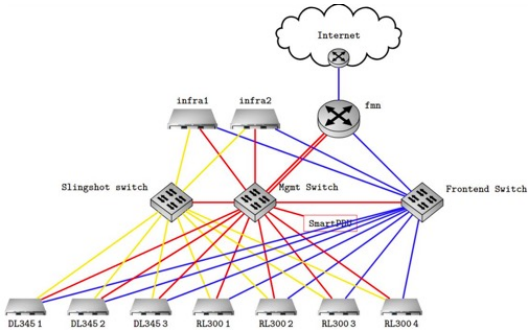
June 12, 2026, Bologna

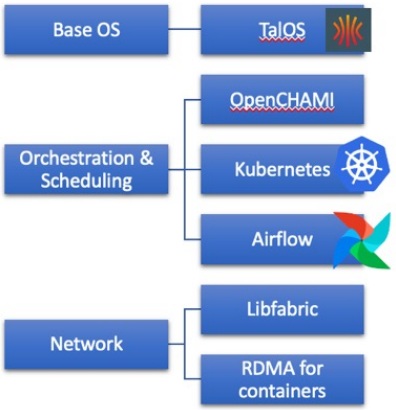
Funded by the European Union
OpenCUBE

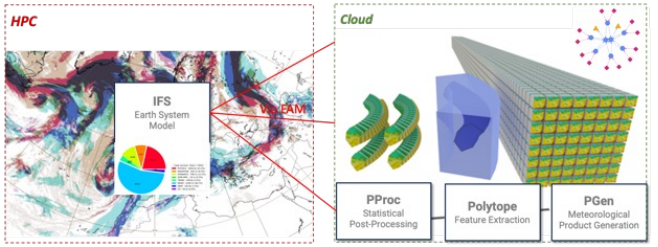
The OpenCUBE Project

- Aim to deliver the first **production-ready** blueprint based on ARM host and RISC-V accelerator for HPC and Cloud workloads:
  - A reference **hardware** implementation
  - A validated **software** stack
  - A set of important pilot **applications**







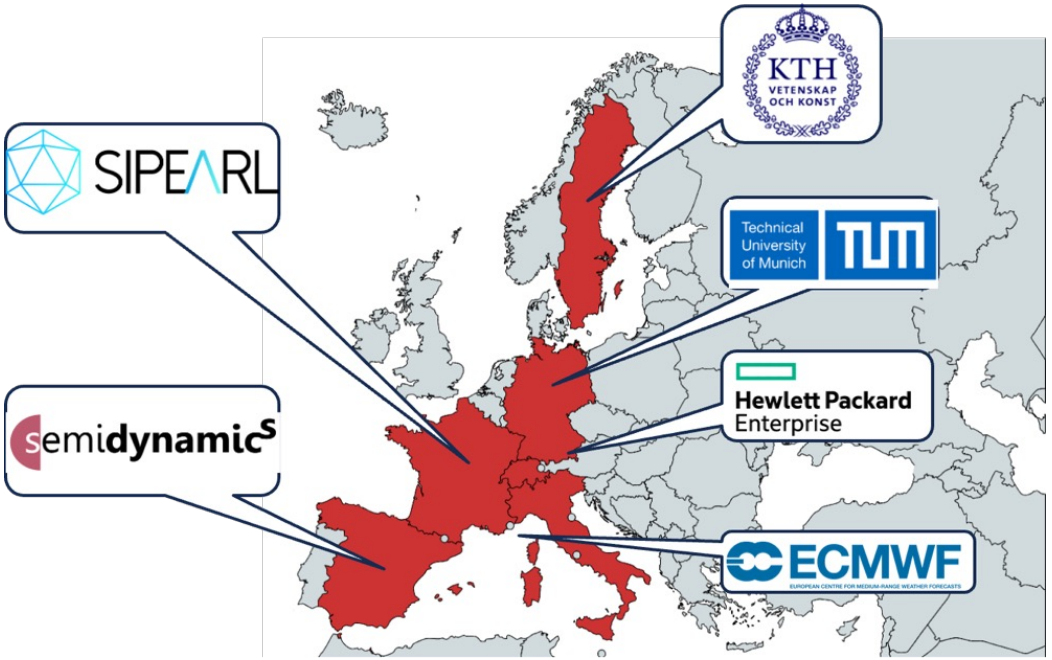


Funded by the European Union

OpenCUBE

The OpenCUBE Consortium

- Coordinator: KTH (Sweden)



https://horizon-opencube.eu/

Funded by the European Union
OpenCUBE

The OpenCUBE Hardware Platform

- Need a testbed for developing and evaluating use cases at scale

Installation at HPE Grenoble Data Center

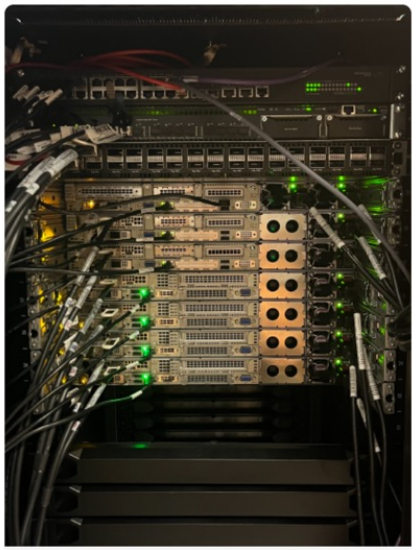
Funded by the European Union

OpenCUBE

The OpenCUBE Hardware Platform

OpenCUBE Pilot Platform

Installed at HPE Grenoble DC, France

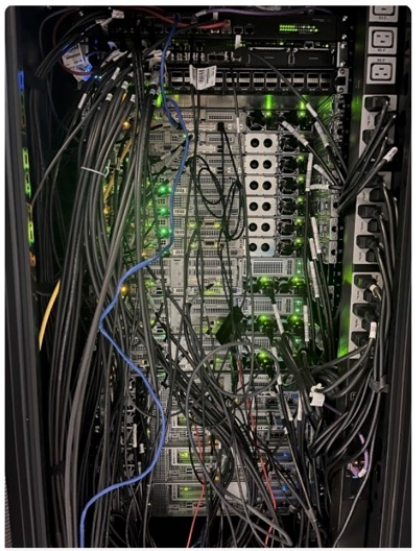


Phase I installation

Consists of 4x RL300 compute nodes with Arm Ampere processors, 2 Infra nodes, Xillinx Alveo U55C FPGA, Slingshot.

Storage Upgrade

Additional installation of three DL345 storage nodes

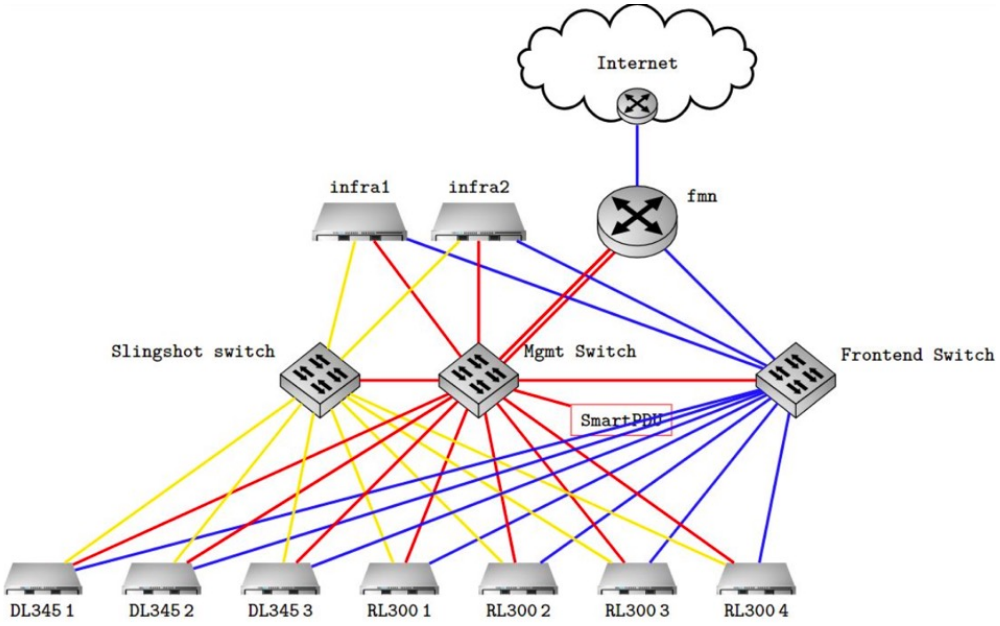


Phase II installation

Upgrade to 8 nodes with 4x ARM AmpereOne servers added

Access to OpenCUBE Platform

- The prototype platform emulates two compute units:
  - ARM Ampere/AmpereOne for Rhea CPU
  - Use FPGA board for Semidynamics Atrevido® RISC-V vector processor




```
 _____  _____  _____  __   __  _____  __   __  _____  _____
/\  __ \/\  == \/\  ___\/\ "-.\ \/\  ___\/\ \/\ \/\  == \/\  ___\
\ \ \/\ \ \  _-/\ \  __\\ \ \-.  \ \ \__ \ \ \_\ \ \  __<\ \  __\
 \ \_____\ \_\   \ \_____\ \_\\"\_\ \_____\ \_____\ \_____\ \_____\
  \/_____/\/_/    \/_____/\/_/ \/_/\/_____/\/_____/\/_____/\/_____/

WELCOME on login.pt.horizon-opencube.eu

System Information:
  Manufacturer: HPE
  Product Name: ProLiant DL325 Gen10 Plus v2
  Processor(s):  1 * AMD EPYC 7443P 24-Core Processor
  Memory:       16 * SK Hynix DDR4-3200 SDRAM (32 GB, ECC, Reg.)

Summary:
  Number of cores:   1 * 24 cores =  24 cores (48 threads)
  Amount of memory:  8 * 32 GB    = 256 GB

Have a lot of fun…
```

```
# cat /proc/cpuinfo
processor       : 0
hart            : 0
isa             : rv64imafdcv
mmu             : sv39
uarch           : epi,avispado

#
```

Funded by the European Union
OpenCUBE

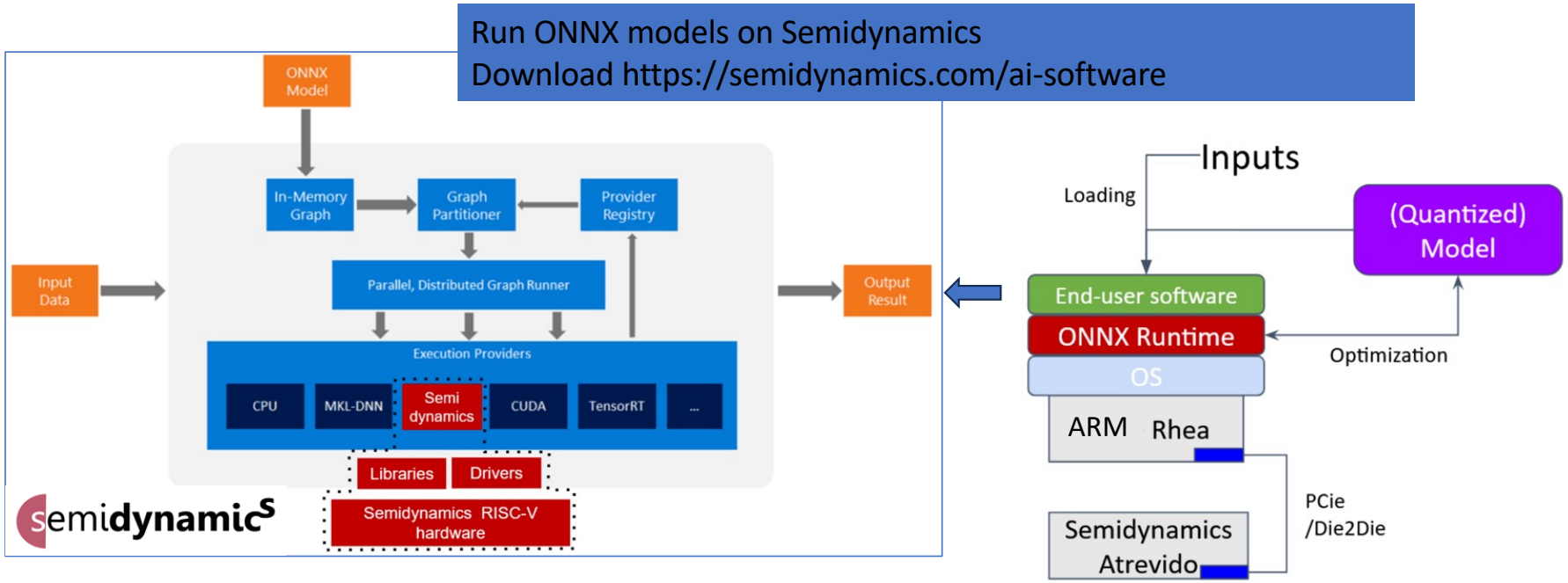
Adopting RVV in HPC Applications

- The barrier is a balance between efforts, portability, and performance
- Machine Learning workloads:
  - ML primitives, e.g., XNNPACK
  - ONNX runtime support
  - …
- Scientific applications:
  - Accelerated math libraries, .e.g., BLAS
  - Compiler support: auto-vectorization
  - …

Funded by the European Union
OpenCUBE

RISC-V ONNX Runtime

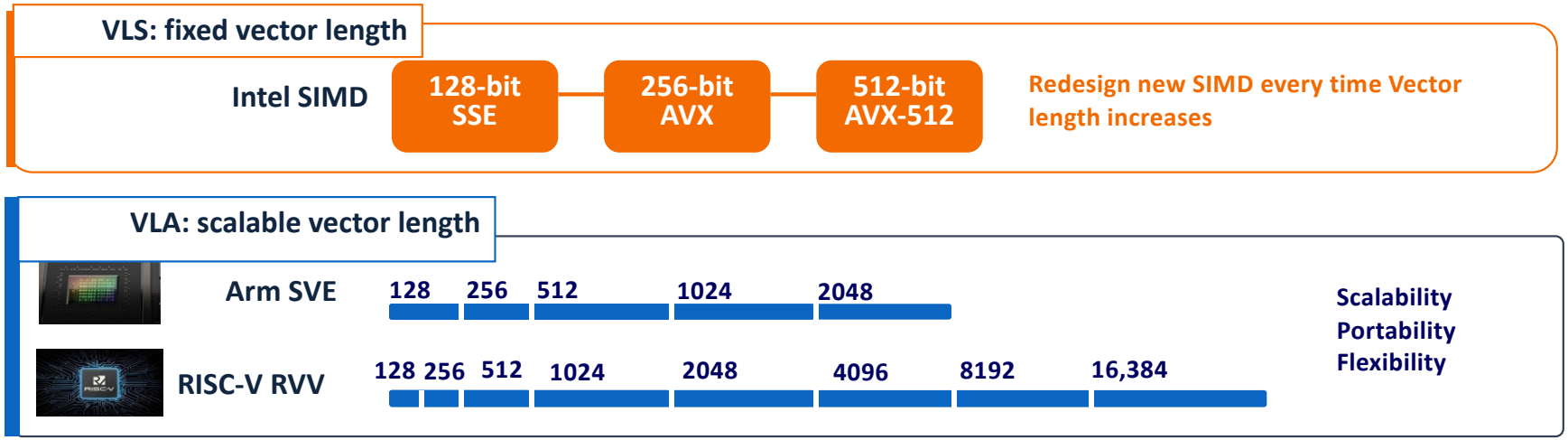
- To utilize **Semidynamics** RVV in OpenCUBE platform:
  - an Execution Provider that uses RVV to optimize the execution of ORT operators
  - an Accelerator Provider to offload computation from ARM host to Semidynamics RVV



Funded by the European Union
OpenCUBE

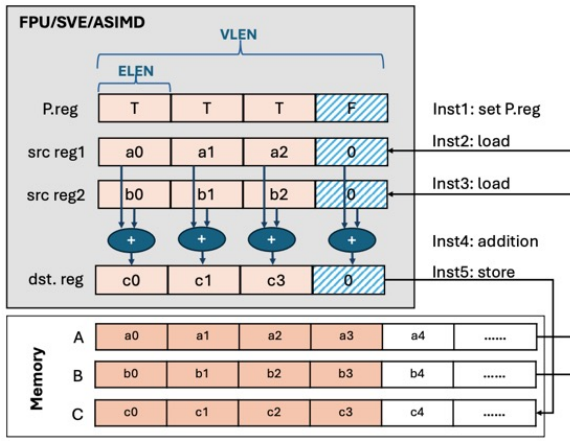
Compiler support for VLA

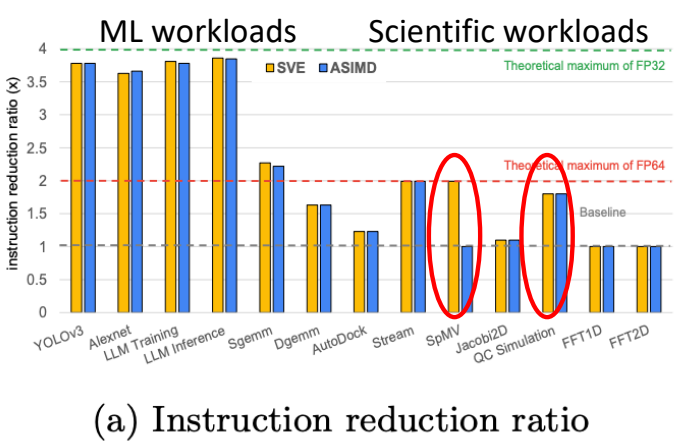
- We compare compiler's support for SVE and RVV for HPC applications
- Both ARM SVE and RISC-V Vector extension support VLA programming
  - vector length is scalable. a vector instruction operates on active vector length
  - In contrast to fixed vector length (VLS): compact ISA, portability



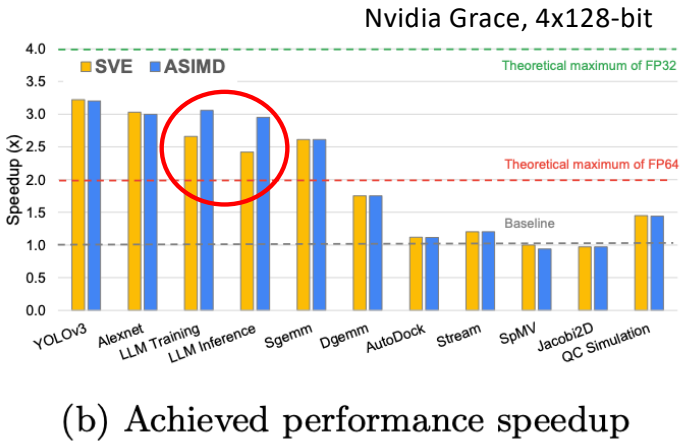
Auto-vectorization for ARM SVE

- GCC toggles between VLA and VLS: -march=armv8-a+sve | armv8-a+asimd
- LLVM: -scalable-vectorization=on|off -march=armv8-a+sve | armv8-a+asimd





(a) Instruction reduction ratio



(b) Achieved performance speedup

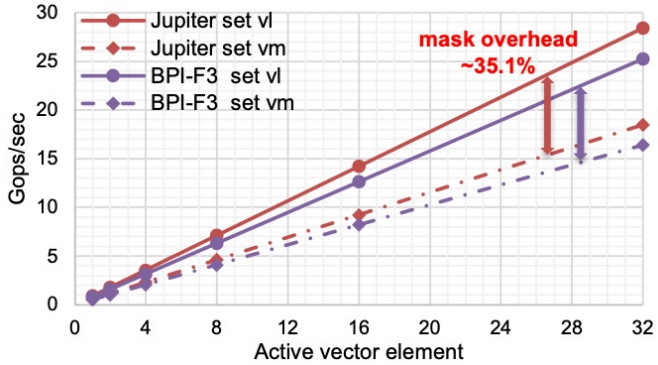
[1]Shi, R., Schieffer, G., Gokhale, M., Lin, P. H., Patel, H., & Peng, I. ARM SVE Unleashed: Performance and Insights Across HPC Applications on Nvidia Grace. EuroPar'25

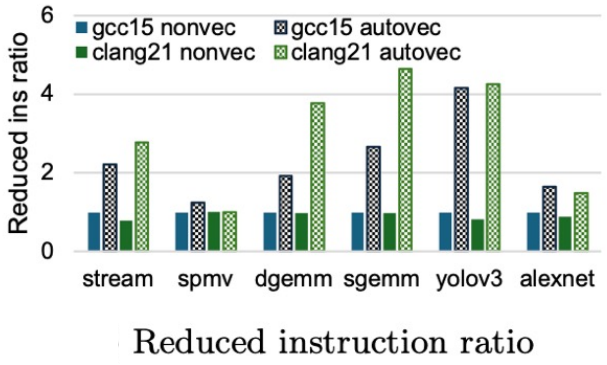
Funded by the European Union
OpenCUBE

Auto-vectorization for RVV

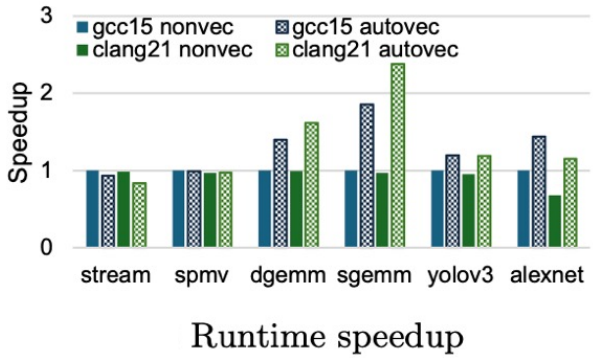
- GCC 14 and LLVM 19 support RVV 1.0. RVV-specific development recently.
  - We compare GCC15 v.s LLVM21 in a set of HPC applications on Spacemit(R) X60 CPUs with 2 × 256-bit RVV1.0
- Before using counter-derived metrics for analyzing utilization efficiency, need to calibrate the counters
  - Used an assembly benchmark suite
  - 7 Counters: retired instruction, vector ins., vector ld/st ins., FP ld/st ins., and FP ins

Two ways of setting tailing elements: setvl v.s. mask





Reduced instruction ratio



Runtime speedup

[1]Shi, R., Gokhale, M., Lin, P., Teruel, X., & Peng I. Closer in the Gap: Towards Portable Performance on RISC-V Vector Processors. To appear at EuroPar'26

Funded by the European Union
OpenCUBE

Performance Counter Calibration on RVV

Table 1: Perf event statistics for different assembly instructions.

Assembly bench.	Ref #ins.	Retired ins.	Vec. ld ins.	Vec. st ins.	Vec. ins.	FP ld ins.	FP st ins.	FP ins.
flw	1.28×10^{10}	1.31×10^{10}	16	16	55	$\mathbf{1.28 \times 10^{10}}$	696	1.28×10^{10}
lw	1.28×10^{10}	1.31×10^{10}	16	16	53	736	855	1613
vle.vv	1.28×10^{10}	1.31×10^{10}	$\mathbf{1.28 \times 10^{10}}$	17	$\mathbf{1.28 \times 10^{10}}$	662	738	2443
fsw	1.28×10^{10}	1.31×10^{10}	16	16	55	601	$\mathbf{1.28 \times 10^{10}}$	1.28×10^{10}
sw	1.28×10^{10}	1.31×10^{10}	16	16	53	736	855	1613
vse.vv	1.28×10^{10}	1.32×10^{10}	1108	$\mathbf{1.28 \times 10^{10}}$	$\mathbf{1.28 \times 10^{10}}$	662	738	2443
vfadd.vv	1.28×10^{10}	1.30×10^{10}	124	17	$\mathbf{1.28 \times 10^{10}}$	652	726	2.56×10^{10}
vmacc.vv	1.28×10^{10}	1.30×10^{10}	144	17	$\mathbf{1.28 \times 10^{10}}$	632	723	2.56×10^{10}
fadd	1.28×10^{10}	1.30×10^{10}	16	16	6.40×10^{9}	770	856	1.28×10^{10}
fmadd	1.28×10^{10}	1.30×10^{10}	16	16	6.40×10^{9}	770	856	1.28×10^{10}

7 Counters: retired instruction, vector ins., vector ld/st ins., FP ld/st ins., and FP ins

[1]Shi, R., Gokhale, M., Lin, P., Teruel, X., & Peng I. Closer in the Gap: Towards Portable Performance on RISC-V Vector Processors. To appear at EuroPar'26

Funded by the European Union
OpenCUBE

A Production Quantum Simulator

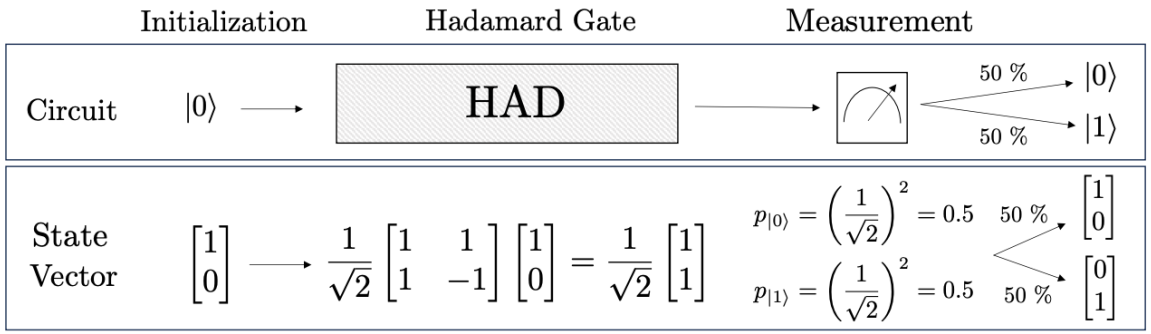
- Qsim is a Schrödinger full state-vector simulator by Google
  - stores all amplitudes of an n-qubit quantum state and updates them when gates are applied.
  - Quantum gate U: a unitary matrix that updates selected amplitude groups.
  - Apply gate: several iteration of matrix-vector multiplication

State-vector representation

$|\psi\rangle = [\, c_0, c_1, c_2, \ldots, c_{2^{n-1}} \,]^T$

Unitary Matrix for k-qubit gate

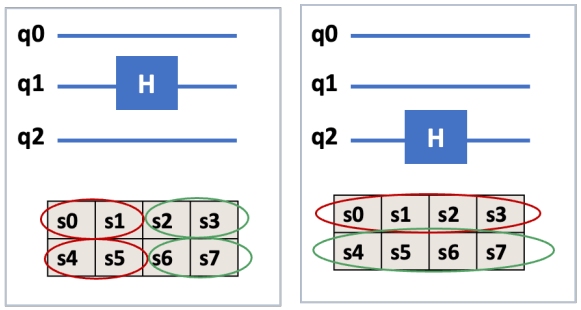
$U \in \mathbb{C}^{(2^k \times 2^k)}, \quad U^\dagger U = I$

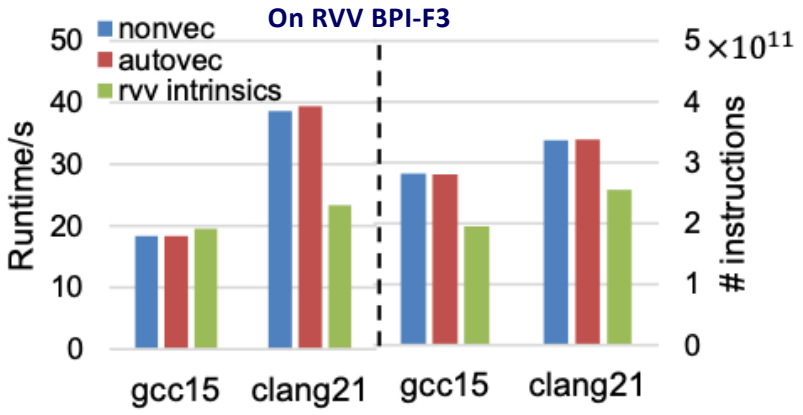


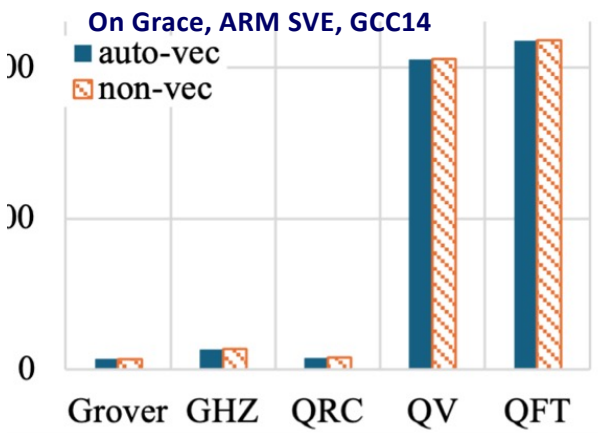
Faj, J., Peng, I., Wahlgren, J., & Markidis, S. (2023). Quantum Computer Simulations at Warp Speed: Assessing the Impact of GPU AccelerationLinks to an external site

A Production Quantum Simulator

- Auto-vectorization by the compiler is insufficient
  - no reduction in instructions or runtime
  - interleaved memory access: long-latency interleaved ld2
  - irregular computation for lower qubit





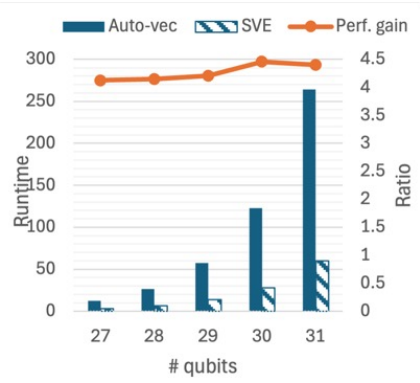


[1]Shi, R., Schieffer, G., Lin, P. H., Gokhale, M., Herten, A., & Peng, I. High-performance Vector-length Agnostic Quantum Circuit Simulations on ARM Processors. IEEE IPDPS 2026.

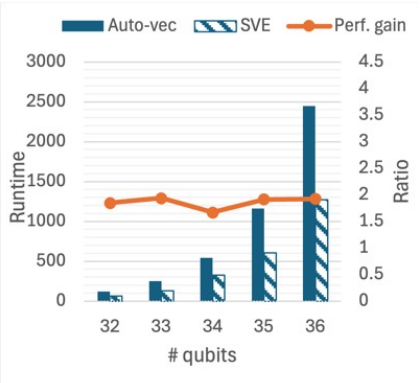
Funded by the European Union
OpenCUBE

A Production Quantum Simulator

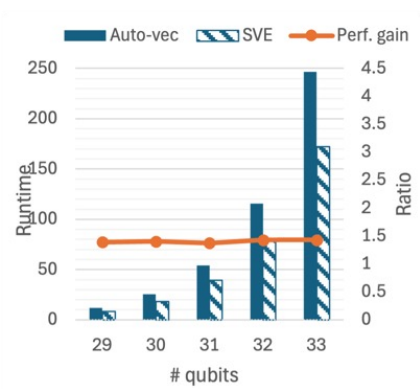
- We move on to a VLA specific design[1]
  - Optimizations include in-place memory layout adjustment, gate fusion
  - Good speedup in SVE intrinsics implementation on 3 ARM systems
  - However, instructions exploded in RVV intrinsics implementation on RVV testbed



A64FX

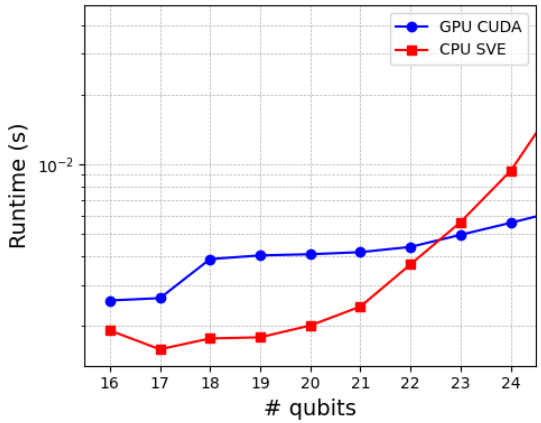


Grace



Graviton

ARM SVE version outperform H100 for < 22 qubits



[1]Shi, R., Schieffer, G., Lin, P. H., Gokhale, M., Herten, A., & Peng, I. High-performance Vector-length Agnostic Quantum Circuit Simulations on ARM Processors. IEEE IPDPS 2026.

Funded by the European Union
OpenCUBE

Takeaways

- OpenCUBE aims to provide a production-ready platform with ARM processors + RVV accelerator

- Adopting RVV in HPC applications can reuse codesign and optimization for ARM SVE
  - Irregular memory access may result in long-latency instructions
  - Irregular computation may result in mask predication instructions

- HPC applications need accurate and reliable performance counters
  - Still lagging on RVV hardware

- Compiler support for RVV is advancing fast but is still less mature than ARM SVE

https://horizon-opencube.eu/