Marcin Copik[1], Eiman Alnuaimi[2], Alok Kamatar[3], Valerie Hayot-Sasson[3], Alberto Madonna[4], Todd Gamblin[5], Kyle Chard[3,6], Ian Foster[3,6], Torsten Hoefler[1,4]

XaaS Containers: Performance-Portable Representation With Source and IR Containers

1 SPCL
2 ETH zürich
3 The University of Chicago
4 CSCS
5 Lawrence Livermore National Laboratory
6 Argonne National Laboratory

Marcin Copik[1], Eiman Alnuaimi[2], Alok Kamatar[3], Valerie Hayot-Sasson[3], Alberto Madonna[4], Todd Gamblin[5], Kyle Chard[3,6], Ian Foster[3,6], Torsten Hoefler[1,4]

XaaS Containers: Performance-Portable Representation With Source and IR Containers

XaaS: Acceleration as a Service to Enable Productive High-Performance Cloud Computing

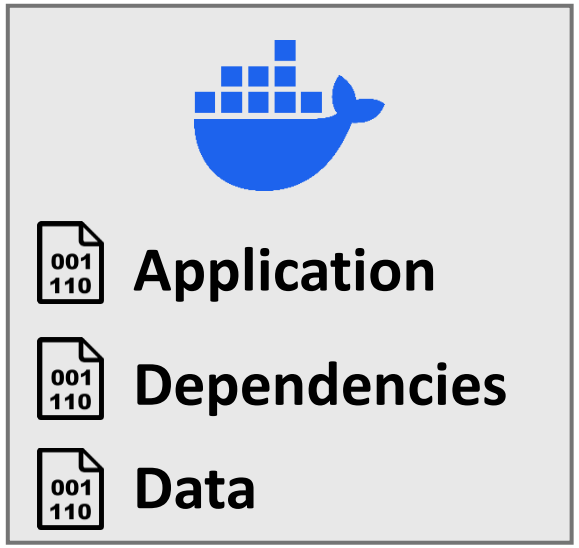
Torsten Hoefler and Marcin Copik, *ETH Zürich, Zürich, 8093, Switzerland*
Pete Beckman, *Argonne National Laboratory, Lemont, IL, 60439, USA*
Andrew Jones, *Microsoft, Redmond, WA, 98052, USA*
Ian Foster, *Argonne National Laboratory, Lemont, IL, 60439, USA*
Manish Parashar, *Utah University, Salt Lake City, UT, 84112, USA*
Daniel Reed, *Utah University, Salt Lake City, UT, 84117, USA*
Matthias Troyer, *Microsoft, Redmond, WA, 98052, USA*
Thomas Schulthess, *Swiss National Supercomputing Centre, Lugano, 6900, Switzerland*
Daniel Ernst, *Nvidia, Santa Clara, CA, 95051, USA*
Jack Dongarra, *University of Tennessee, Knoxville, TN, 37996, USA*

rich 3 THE UNIVERSITY OF CHICAGO

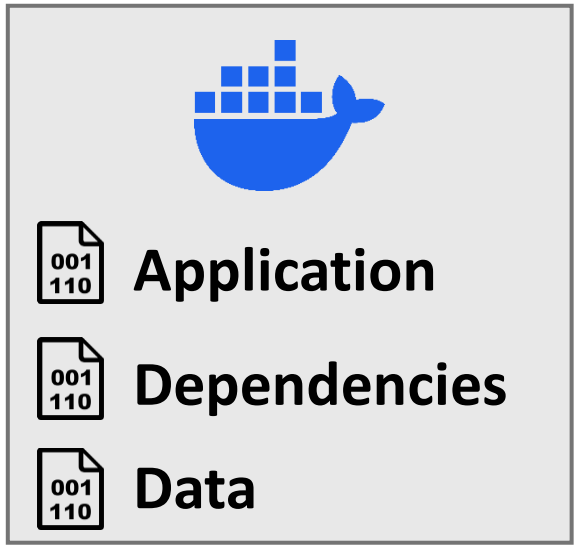
scs 5 Lawrence Livermore National Laboratory

e LABORATORY

Wonderful World of Containers



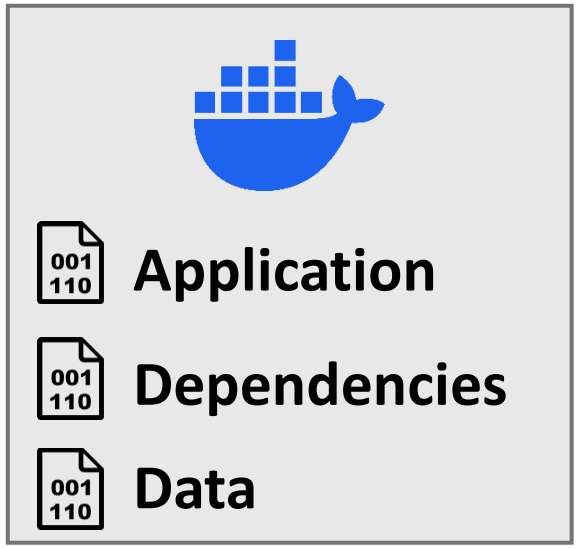
Wonderful World of Containers



docker run gromacs:2025.0

x64 & arm64

Wonderful World of Containers

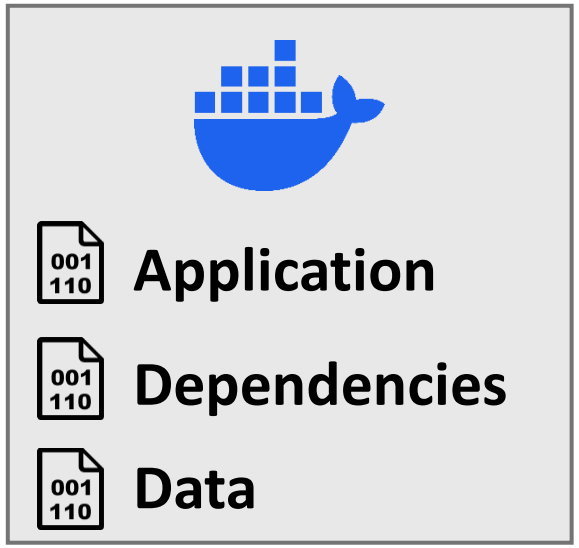


Will it work?

docker run gromacs:2025.0

x64 & arm64

Wonderful World of Containers



Will it work?

Will it be fast?

docker run gromacs:2025.0

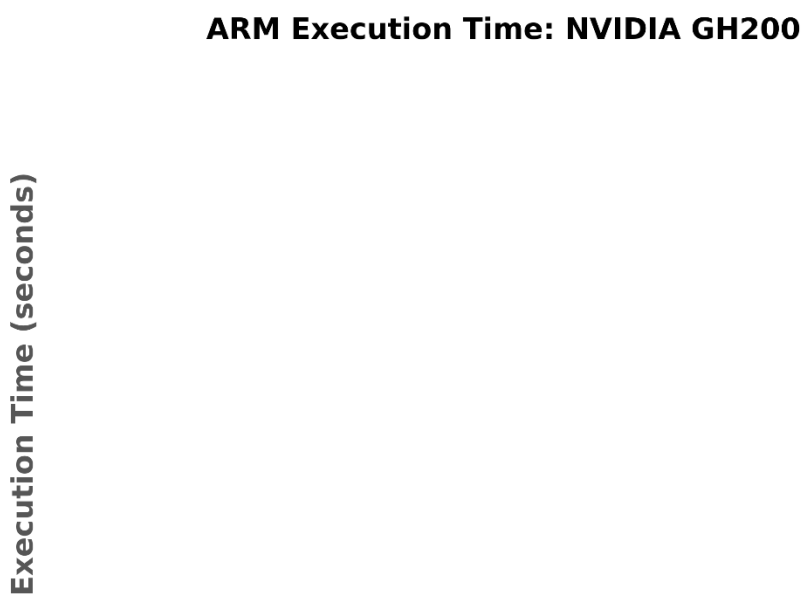
x64 & arm64

Containers: Portable, but not Performant

gromacs:2025.0, x64 & arm64

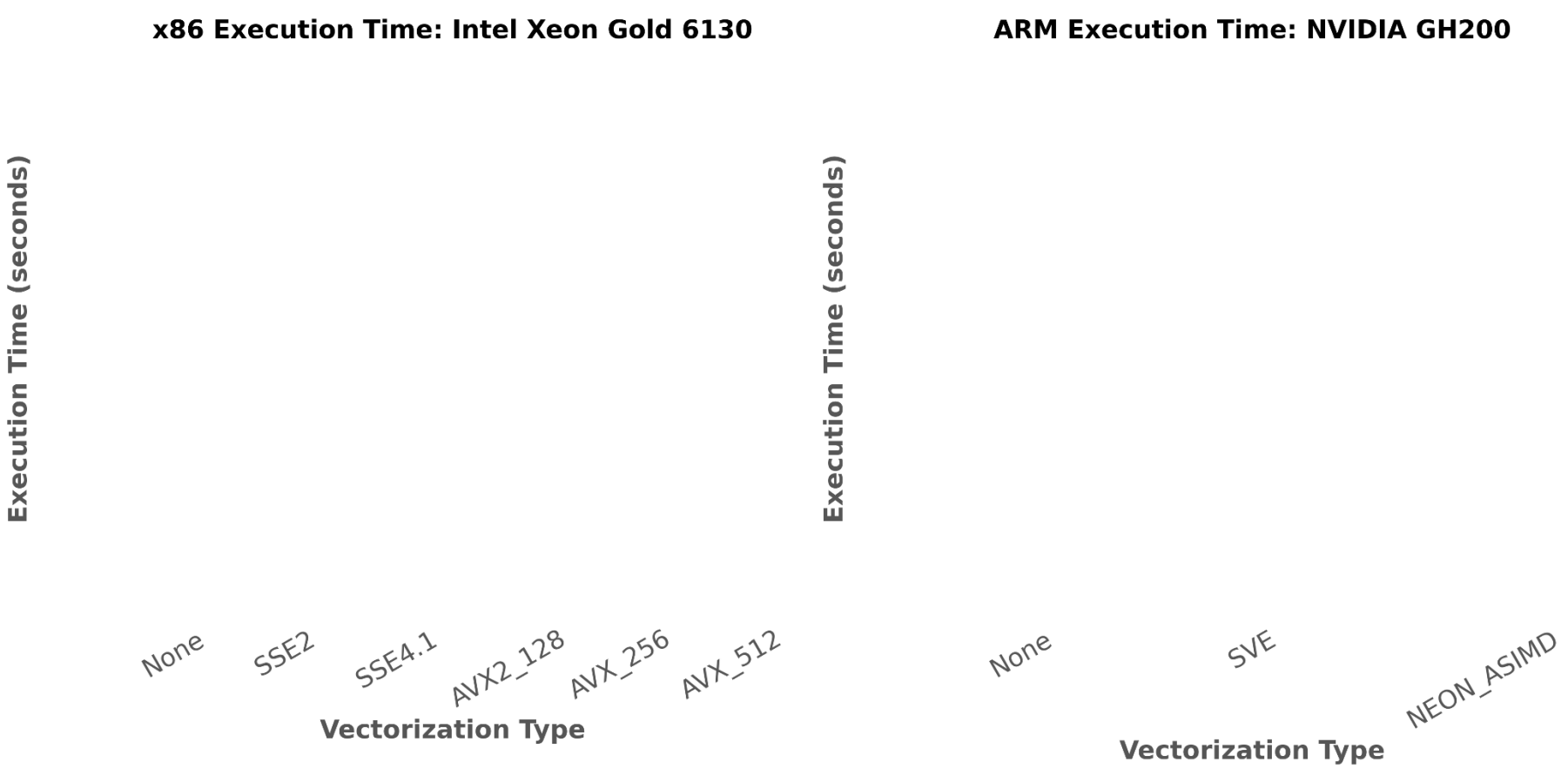
Containers: Portable, but not Performant

gromacs:2025.0, x64 & arm64



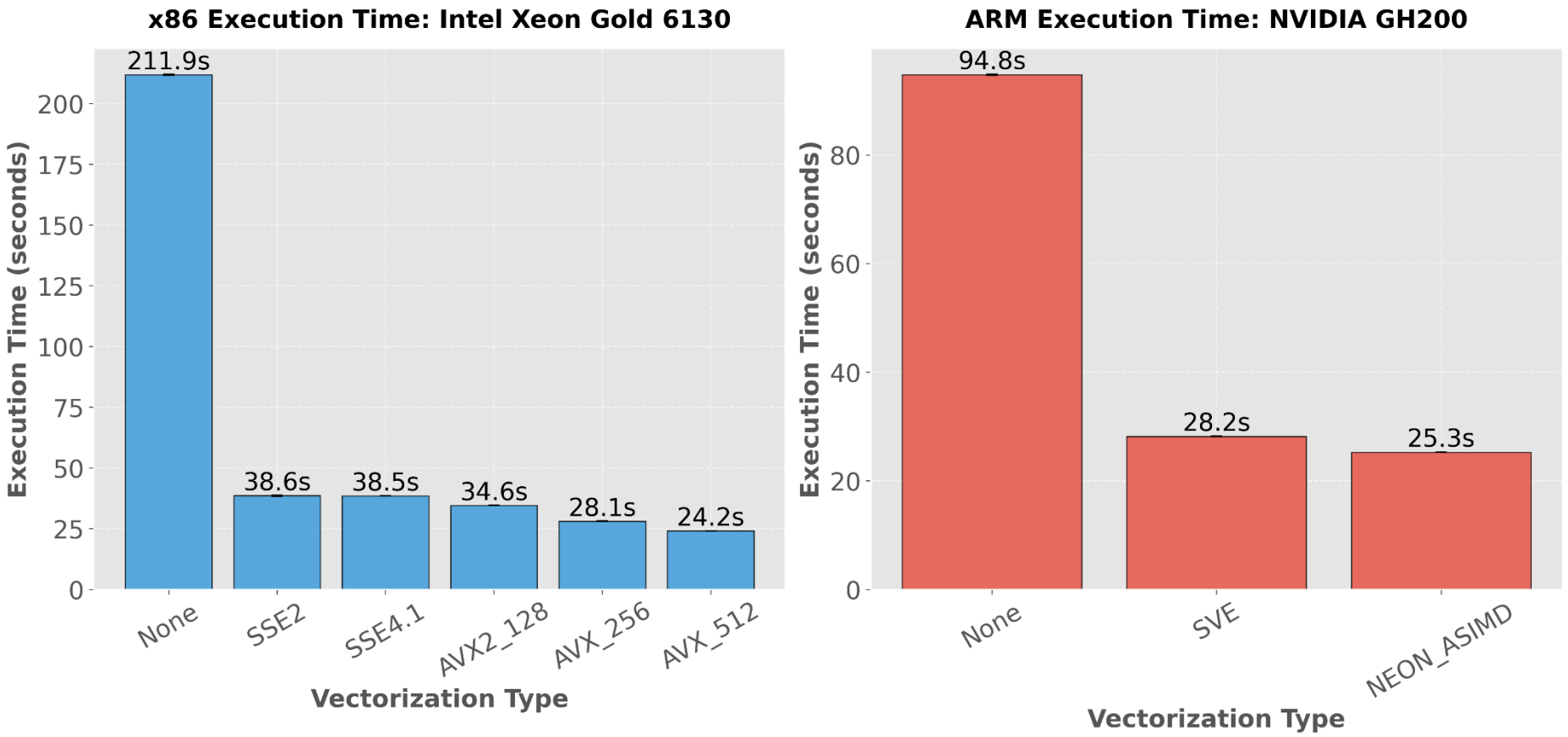
Containers: Portable, but not Performant

gromacs:2025.0, x64 & arm64



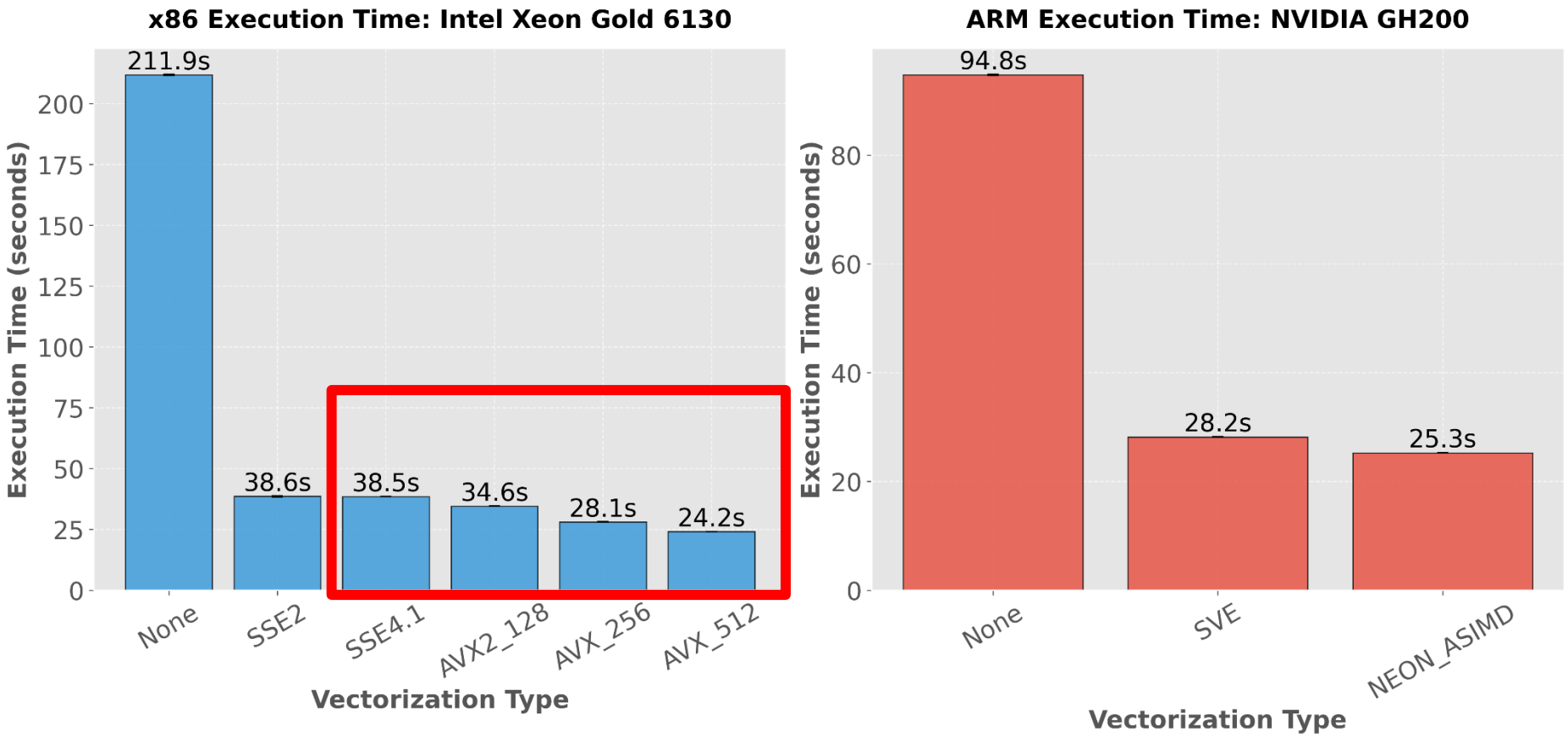
Containers: Portable, but not Performant

gromacs:2025.0, x64 & arm64



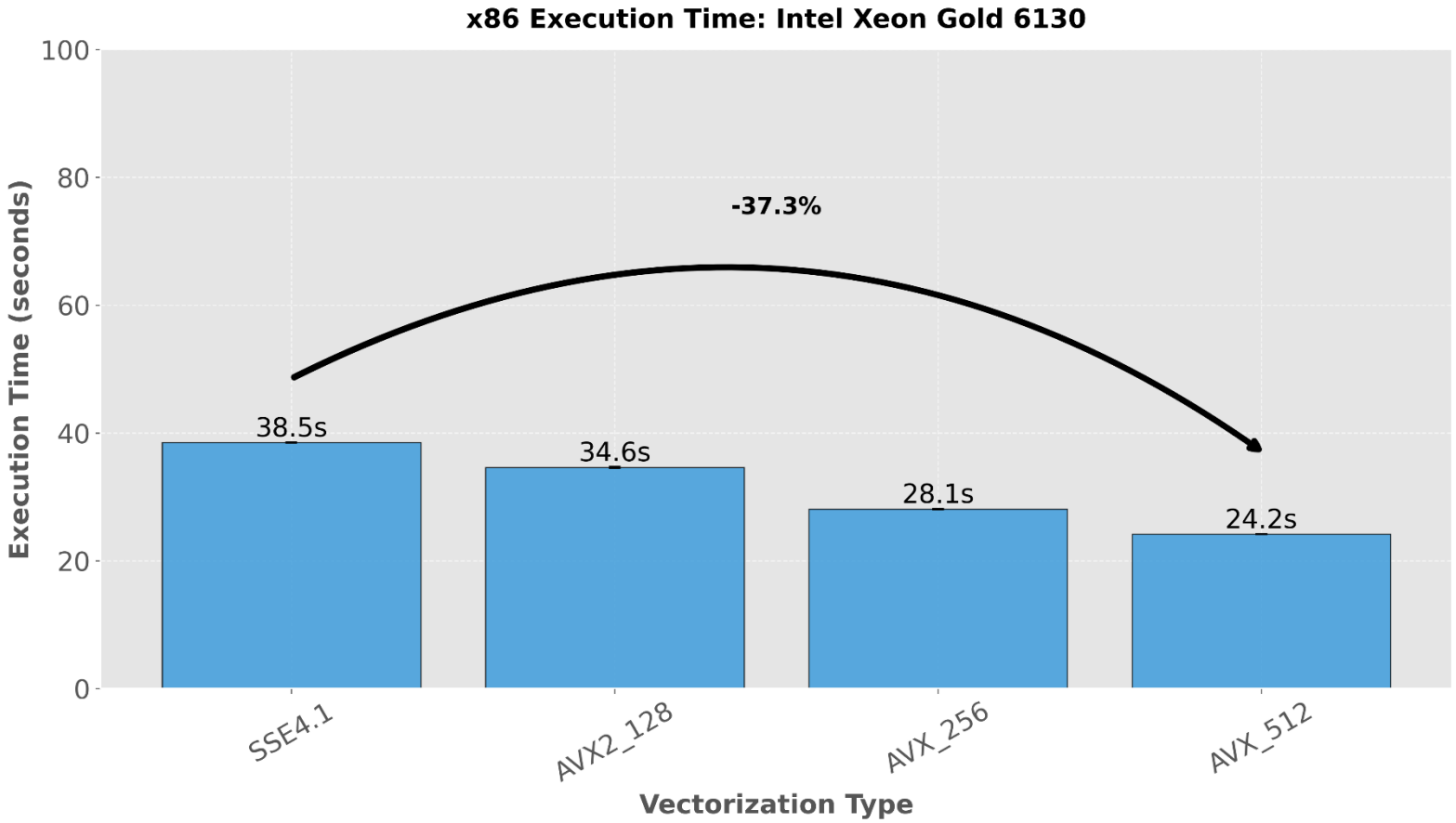
Containers: Portable, but not Performant

gromacs:2025.0, x64 & arm64

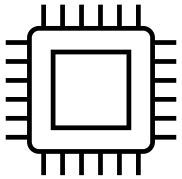


Containers: Portable, but not Performant

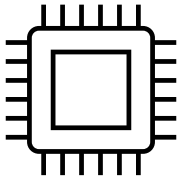
gromacs:2025.0, x64 & arm64



Just One More Container…



Just One More Container…



gromacs:2025.0

sse4.1

Dependencies , Data

gromacs:2025.0

avx-128

Dependencies , Data

gromacs:2025.0

avx-256

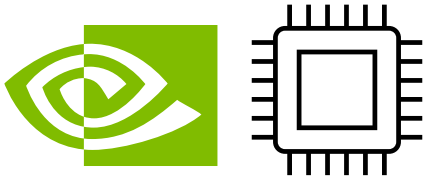
Dependencies , Data

gromacs:2025.0

avx-512

Dependencies , Data

Just One More Container…



gromacs:2025.0

sse4.1

Dependencies , Data

gromacs:2025.0

avx-128

Dependencies , Data

gromacs:2025.0

avx-256

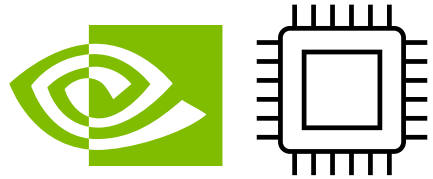
Dependencies , Data

gromacs:2025.0

avx-512

Dependencies , Data

Just One More Container…



gromacs:2025.0

sse4.1

Dependencies , Data

gromacs:2025.0

avx-128

Dependencies , Data

gromacs:2025.0

avx-256

Dependencies , Data

gromacs:2025.0

avx-512

Dependencies , Data

gromacs:2025.0

sse4.1-cuda

Dependencies , Data

gromacs:2025.0

avx-128-cuda

Dependencies , Data

gromacs:2025.0

avx-256-cuda

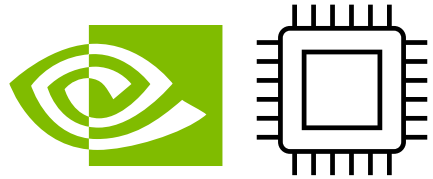
Dependencies , Data

gromacs:2025.0

avx-512-cuda

Dependencies , Data

Just One More Container…



gromacs:2025.0

sse4.1

gromacs:2025.0

sse4.1-cuda

001 110 Dependencies , Data

gromacs:2025.0

avx-128

gromacs:2025.0

avx-128-cuda

001 110 Dependencies , Data

gromacs:2025.0

avx-256

gromacs:2025.0

avx-256-cuda

001 110 Dependencies , Data

gromacs:2025.0

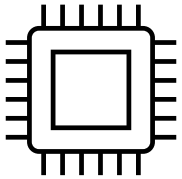
avx-512

gromacs:2025.0

avx-512-cuda

001 110 Dependencies , Data

Just One More Container…

gromacs:2025.0

sse4.1

gromacs:2025.0

sse4.1-cuda

001 110 Dependencies , Data

gromacs:2025.0

avx-128

gromacs:2025.0

avx-128-cuda

001 110 Dependencies , Data

gromacs:2025.0

avx-256

gromacs:2025.0

avx-256-cuda

001 110 Dependencies , Data

gromacs:2025.0

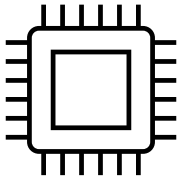
avx-512

gromacs:2025.0

avx-512-cuda

001 110 Dependencies , Data

Just One More Container…

gromacs:2025.0

sse4.1

gromacs:2025.0

sse4.1-cuda

gromacs:2025.0

sse4.1-rocm

001 110 Dependencies , Data

gromacs:2025.0

avx-128

gromacs:2025.0

avx-128-cuda

gromacs:2025.0

avx-128-rocm

001 110 Dependencies , Data

gromacs:2025.0

avx-256

gromacs:2025.0

avx-256-cuda

gromacs:2025.0

avx-256-rocm

001 110 Dependencies , Data

gromacs:2025.0

avx-512

gromacs:2025.0

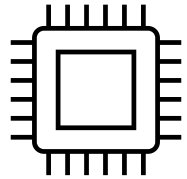
avx-512-cuda

gromacs:2025.0

avx-512-rocm

001 110 Dependencies , Data

Just One More Container…



gromacs:2025.0 **sse4.1**

gromacs:2025.0 **sse4.1-cuda**

gromacs:2025.0 **sse4.1-rocm**

001 110 Dependencies , Data

gromacs:2025.0 **avx-128**

gromacs:2025.0 **avx-128-cuda**

gromacs:2025.0 **avx-128-rocm**

001 110 Dependencies , Data

gromacs:2025.0 **avx-256**

gromacs:2025.0 **avx-256-cuda**

gromacs:2025.0 **avx-256-rocm**

001 110 Dependencies , Data

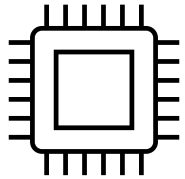
gromacs:2025.0 **avx-512**

gromacs:2025.0 **avx-512-cuda**

gromacs:2025.0 **avx-512-rocm**

001 110 Dependencies , Data

Just One More Container…



gromacs:2025.0 **sse4.1**

gromacs:2025.0 **sse4.1-cuda**

gromacs:2025.0 **sse4.1-rocm**

gromacs:2025.0 **sse4.1-sycl**

001 110 Dependencies , Data

gromacs:2025.0 **avx-128**

gromacs:2025.0 **avx-128-cuda**

gromacs:2025.0 **avx-128-rocm**

gromacs:2025.0 **avx-128-sycl**

001 110 Dependencies , Data

gromacs:2025.0 **avx-256**

gromacs:2025.0 **avx-256-cuda**

gromacs:2025.0 **avx-256-rocm**

gromacs:2025.0 **avx-256-sycl**

001 110 Dependencies , Data

gromacs:2025.0 **avx-512**

gromacs:2025.0 **avx-512-cuda**

gromacs:2025.0 **avx-512-rocm**

gromacs:2025.0 **avx-512-sycl**

001 110 Dependencies , Data

Just One More Container…

Which BLAS library should I use? Which FFT library?

gromacs:2025.0 sse4.1-cuda

gromacs:2025.0 avx-128-cuda

gromacs:2025.0 avx-256-cuda

gromacs:2025.0 avx-512-cuda

gromacs:2025.0 sse4.1-rocm

gromacs:2025.0 avx-128-rocm

gromacs:2025.0 avx-256-rocm

gromacs:2025.0 avx-512-rocm

gromacs:2025.0 sse4.1-sycl

001 110 Dependencies , Data

gromacs:2025.0 avx-128-sycl

001 110 Dependencies , Data

gromacs:2025.0 avx-256-sycl

001 110 Dependencies , Data

gromacs:2025.0 avx-512-sycl

001 110 Dependencies , Data

Just One More Container…

Which BLAS library should I use? Which FFT library?

Should we use OpenMP? Which MPI?

gromacs:2025.0

sse4.1-rocm

avx-128-rocm

avx-256-rocm

avx-512-rocm

gromacs:2025.0

sse4.1-sycl

avx-128-sycl

avx-256-sycl

avx-512-sycl

Dependencies , Data

Just One More Container…

Which BLAS library should I use? Which FFT library?

Should we use OpenMP? Which MPI?

What if container has CUDA 12.9, and my system driver is at 12.1?
For which CUDA architecture should we emit code?

HPC Specialization Points

HPC Specialization Points

Vectorization

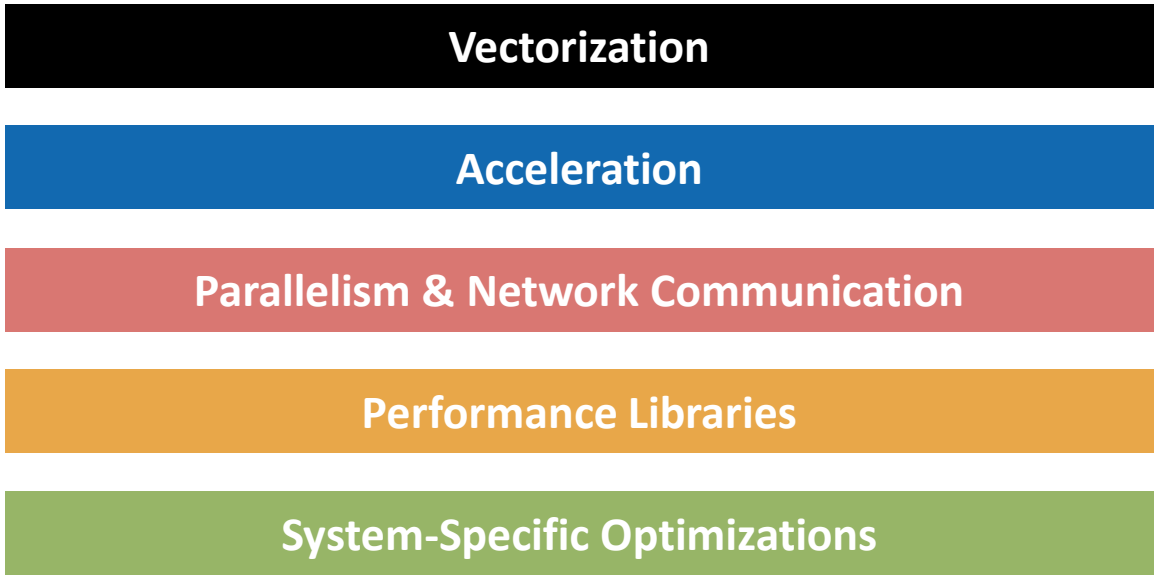
Acceleration

Parallelism & Network Communication

Performance Libraries

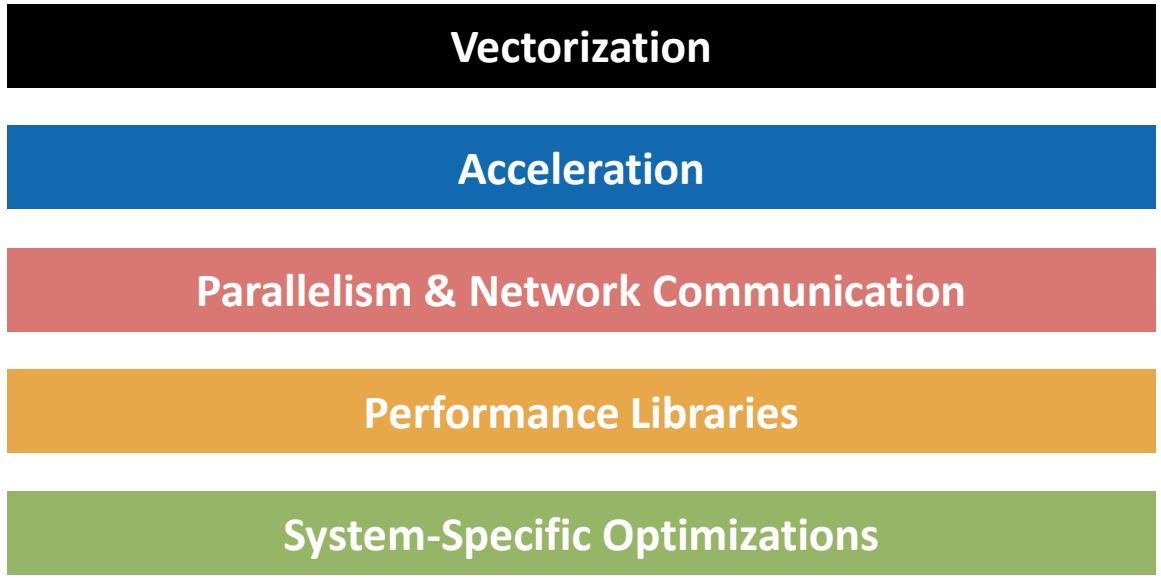
System-Specific Optimizations

HPC Specialization Points



Specialization Points

HPC Specialization Points



Specialization Points

Difficult to build one binary that supports all possible options!

HPC Specialization Points

Vectorization

Acceleration

Parallelism & Network Communication

Performance Libraries

System-Specific Optimizations

Specialization Points

WheelNext

Difficult to build one binary that supports all possible options!

HPC Specialization Points

Vectorization

Acceleration

Parallelism & Network Communication

Performance Libraries

System-Specific Optimizations

Specialization Points

Difficult to build one binary that supports all possible options!

WheelNext

blas-lapack-variant-provider

This is a plugin for the proposed wheel variants implementation that provides BLAS/LAPACK library selection.

This plugin always returns a fixed list of supported variants, and so can be used as a build-time plugin.

HPC Specialization Points

Vectorization

Acceleration

Parallelism & Network Communication

Performance Libraries

System-Specific Optimizations

Specialization Points

Difficult to build one binary that supports all possible options!

WheelNext

blas-lapack-variant-provider

This is a plugin for the proposed wheel variants implementation that provides BLAS/LAPACK library selection.

This plugin always returns a fixed list of supported variants, and so can be used as a build-time plugin.

HPC is not only Python!
We need a **general-purpose** solution.

How to Provide Portability for HPC Software?

How to Provide Portability for HPC Software?

Building Source

How to Provide Portability for HPC Software?

Building Source

Relinking Dependencies

How to Provide Portability for HPC Software?

Spack

EasyBuild

Building Source

NVIDIA PTX

(Re)lowering Application

SARUS

podman

Relinking Dependencies

How to Provide Portability for HPC Software?

Spack

EasyBuild

NVIDIA
PTX

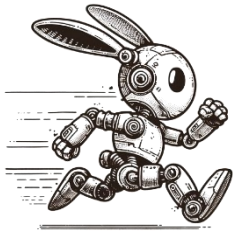
SARUS

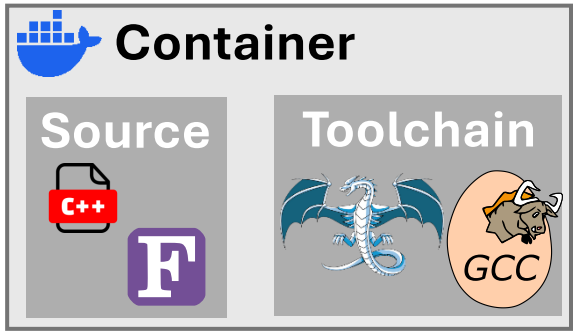
podman

Building Source

(Re)lowering Application

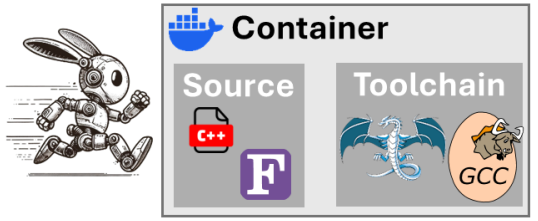
Relinking Dependencies





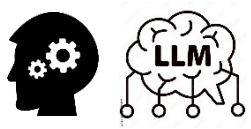
XaaS Source Container

Configuring Containers for HPC Specializations



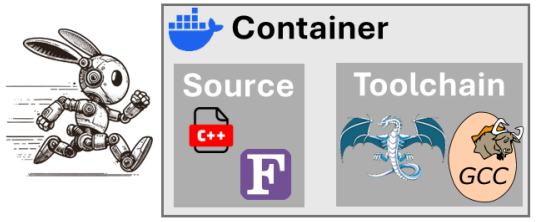
XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



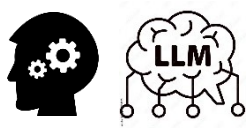
Specialization Discovery

Configuring Containers for HPC Specializations



XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



Specialization Discovery

GROMACS

In-Context Learning

Gemini 1.5 & 2.0

F1 Scores: 0.86 – 0.99

llama.cpp

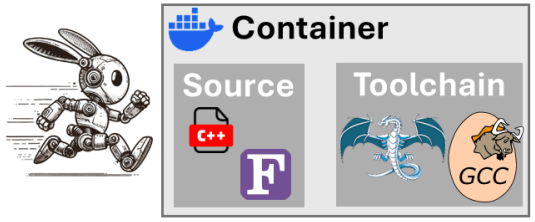
Generalization

GPT-o3

F1 Scores: 0.73 - 0.79

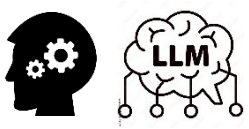
F1 Score: harmonic mean between **precision** (minimize false positives) and **recall** (minimize false negatives).

Configuring Containers for HPC Specializations



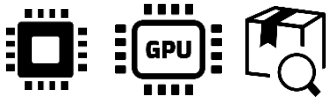
XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



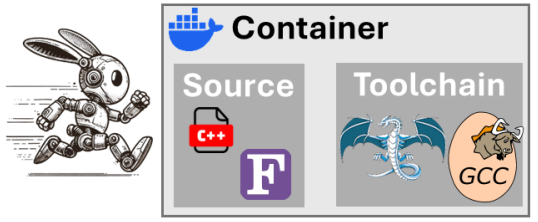
Specialization Discovery

```
{
 "CPU Info": {
   "Architecture": "x86_64",
   "Vectorization": [
    "avx512f", "avx", "avx2", "sse4_1"
   ]
 },
 "GPU Backends": {
   "CUDA": { "version": "12.1",
     "lib": ["/lib/libcuda.so.1"],
   }
 }
}
```



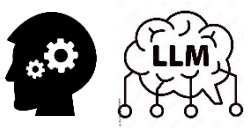
System Discovery

Configuring Containers for HPC Specializations



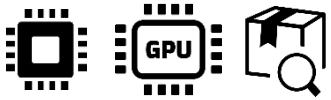
XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



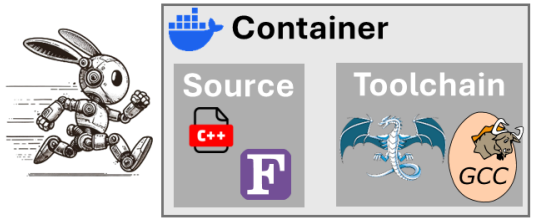
Specialization Discovery

```
{
 "CPU Info": {
   "Architecture": "x86_64",
   "Vectorization": [
    "avx512f", "avx", "avx2", "sse4_1"
   ]
 },
 "GPU Backends": {
   "CUDA": { "version": "12.1",
     "lib": ["/lib/libcuda.so.1"],
   }
 }
}
```



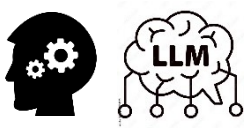
System Discovery

Configuring Containers for HPC Specializations



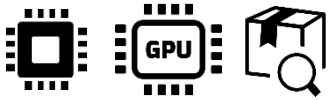
XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



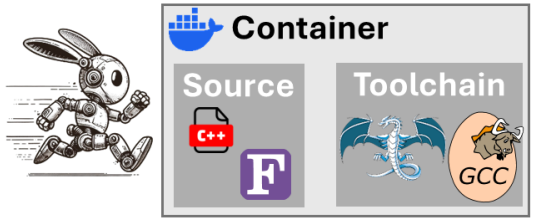
Specialization Discovery

```
{
 "CPU Info": {
   "Architecture": "x86_64",
   "Vectorization": [
    "avx512f", "avx", "avx2", "sse4_1"
   ]
 },
 "GPU Backends": {
   "CUDA": { "version": "12.1",
     "lib": ["/lib/libcuda.so.1"],
   }
 }
}
```



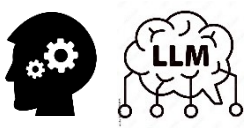
System Discovery

Configuring Containers for HPC Specializations



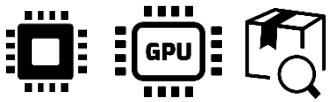
XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



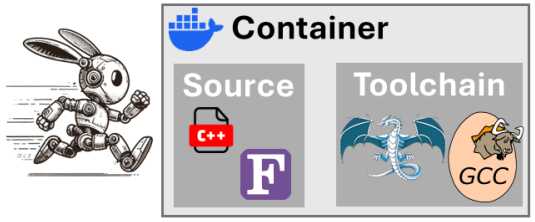
Specialization Discovery

```
{
 "CPU Info": {
   "Architecture": "x86_64",
   "Vectorization": [
    "avx512f", "avx", "avx2", "sse4_1"
   ]
 },
 "GPU Backends": {
   "CUDA": { "version": "12.1",
     "lib": ["/lib/libcuda.so.1"],
   }
 }
}
```



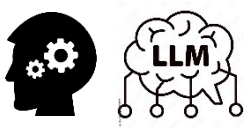
System Discovery

Configuring Containers for HPC Specializations



XaaS Source Container

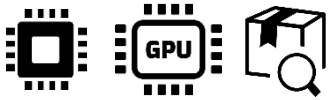
```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



Specialization Discovery

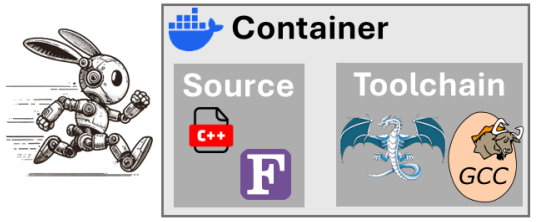
```
{
 "CPU Info": {
   "Architecture": "x86_64",
   "Vectorization": [
    "avx512f", "avx", "avx2", "sse4_1"
   ]
 },
 "GPU Backends": {
   "CUDA": { "version": "12.1",
     "lib": ["/lib/libcuda.so.1"],
   }
 }
}
```

```
"vectorization_flags": {
  "SSE4.1": "-DGMX_SIMD=SSE4.1",
  "AVX_512": "-DGMX_SIMD=AVX_512",
  "AVX2_256": "-DGMX_SIMD=AVX2_256"
},
```



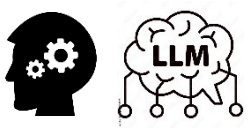
System Discovery

Configuring Containers for HPC Specializations



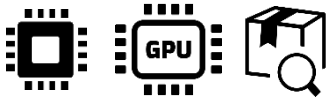
XaaS Source Container

```
{
  "gpu_build": {"value": true, "build_flag": "-DGMX_GPU"},
  "gpu_backends": {
    "CUDA": {"min_version": "12.1", "flag": "-DGMX_GPU=CUDA"},
    "HIP": {"min_version": "5.4.3", "flag": "-DGMX_GPU=HIP"}
  },
  "vectorization": {
    "None": {"flag": "-DGMX_SIMD=None"},
    "SSE4.1": {"flag": "-DGMX_SIMD=SSE4.1"},
    "AVX2_256": {"flag": "-DGMX_SIMD=AVX2_256"},
    "AVX_512": {"flag": "-DGMX_SIMD=AVX_512"},
    "ARM_NEON_ASIMD": {"flag": "-DGMX_SIMD=ARM_NEON_ASIMD"}
}}
```



Specialization Discovery

```
{
 "CPU Info": {
   "Architecture": "x86_64",
   "Vectorization": [
    "avx512f", "avx", "avx2", "sse4_1"
   ]
 },
 "GPU Backends": {
   "CUDA": { "version": "12.1",
     "lib": ["/lib/libcuda.so.1"],
   }
 }
}
```

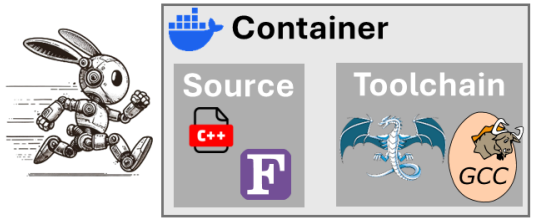


System Discovery

```
{
  "common_specialization": {
    "vectorization_flags": {
      "SSE4.1": "-DGMX_SIMD=SSE4.1",
      "AVX_512": "-DGMX_SIMD=AVX_512",
      "AVX2_256": "-DGMX_SIMD=AVX2_256"
    },
    "gpu_backends": {
      "CUDA": { "version": "12.1",
        "flag": "-DGMX_GPU=CUDA",
      },
    }}
}
```

Feature Intersection

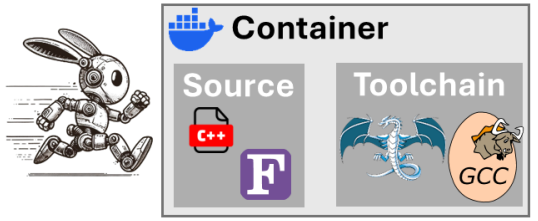
XaaS Source Containers Deployment



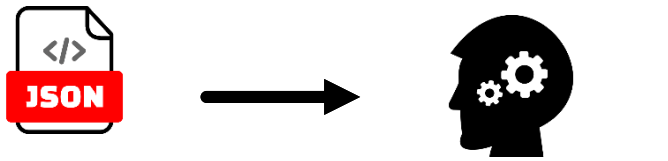
XaaS Source Container

Feature Intersection

XaaS Source Containers Deployment



XaaS Source Container

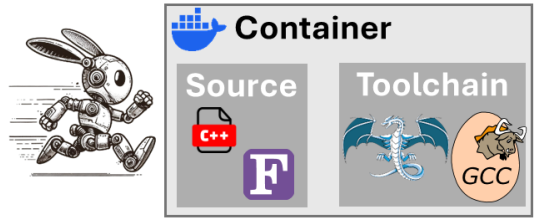


Feature Intersection

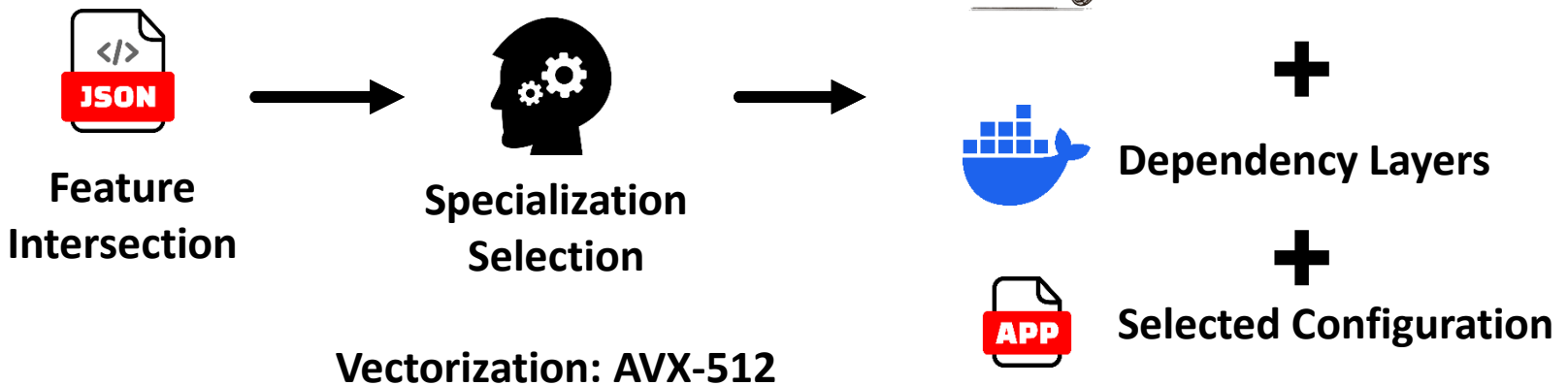
Specialization Selection

Vectorization: AVX-512
GPU: CUDA
...

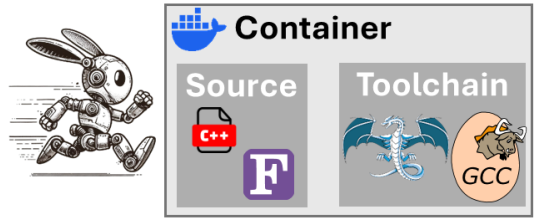
XaaS Source Containers Deployment



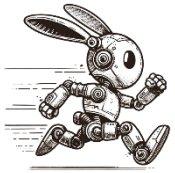
XaaS Source Container



XaaS Source Containers Deployment



XaaS Source Container



XaaS Source Container
Source Code + Dev Tools

Dependency Layers

Selected Configuration

Feature Intersection

Specialization Selection

Vectorization: AVX-512
GPU: CUDA
...

Deployed XaaS Container

How to Provide Portability for HPC Software?

Spack

EasyBuild

NVIDIA
PTX

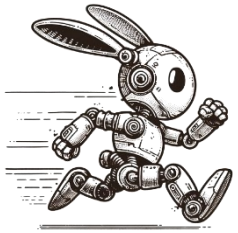
SARUS

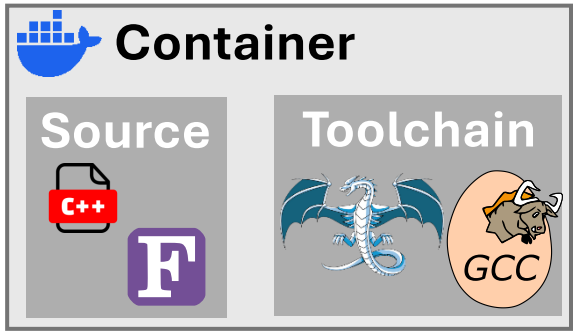
podman

Building
Source

(Re)lowering
Application

Relinking
Dependencies





XaaS Source Container

How to Provide Portability for HPC Software?

Spack

EasyBuild

NVIDIA
PTX

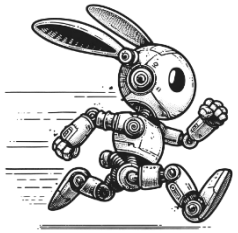
SARUS

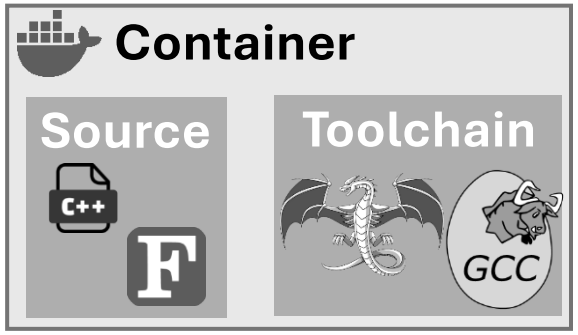
podman

Building
Source

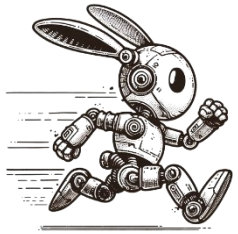
(Re)lowering
Application

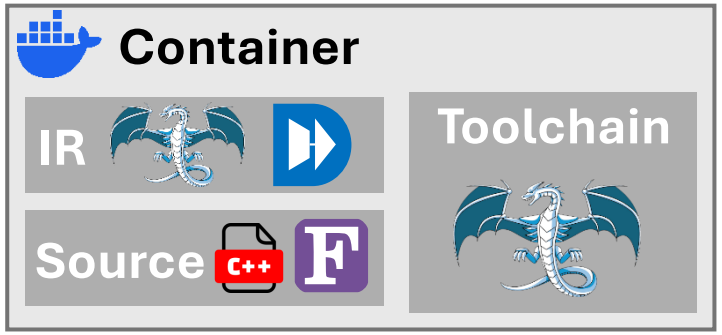
Relinking
Dependencies





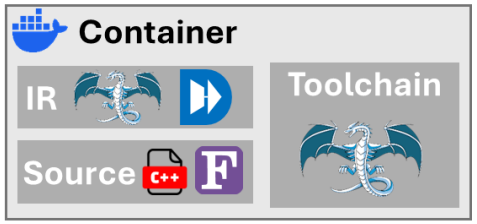
XaaS Source Container





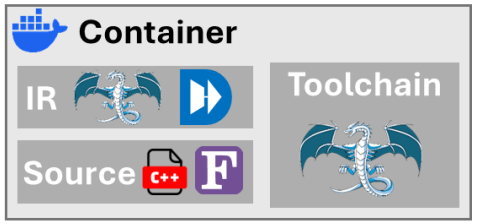
XaaS IR Container

XaaS IR Containers: Eliminating Redundancy

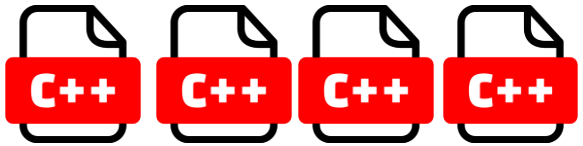


XaaS IR Container

XaaS IR Containers: Eliminating Redundancy



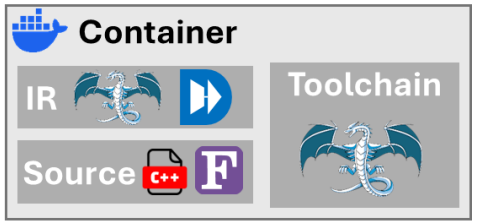
XaaS IR Container



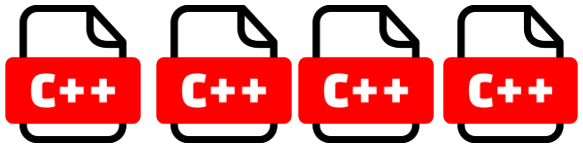
Config № 1

AVX-256, no GPU, OpenMP

XaaS IR Containers: Eliminating Redundancy

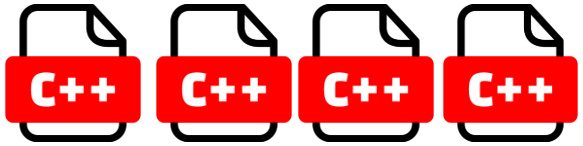


XaaS IR Container



Config № 1

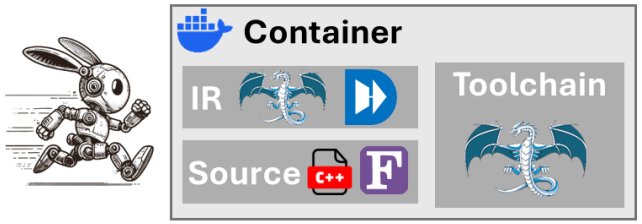
AVX-256, no GPU, OpenMP



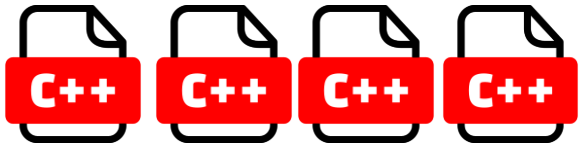
Config № 2

AVX-512, CUDA, MPI

XaaS IR Containers: Eliminating Redundancy

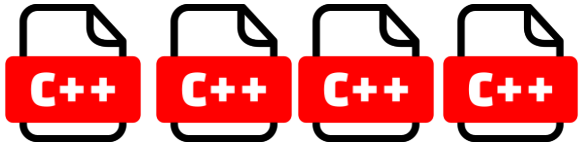


XaaS IR Container



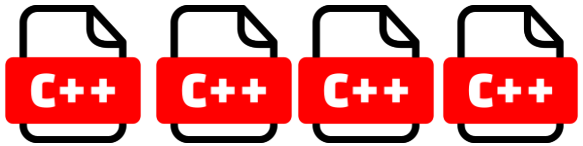
Config № 1

AVX-256, no GPU, OpenMP



Config № 2

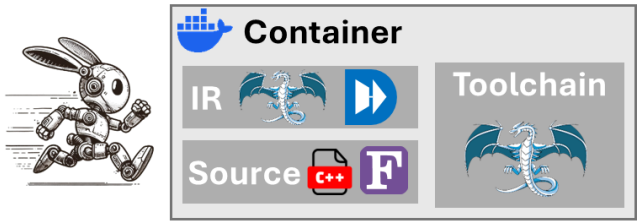
AVX-512, CUDA, MPI



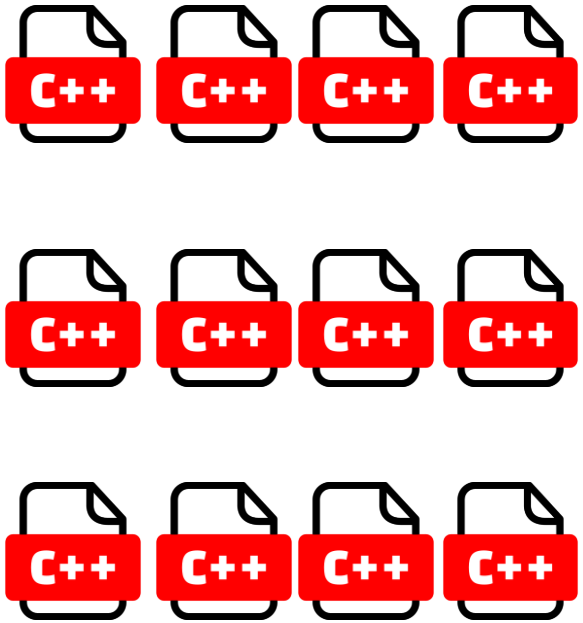
Config № 3

AVX-256, CUDA, MPI

XaaS IR Containers: Eliminating Redundancy



XaaS IR Container



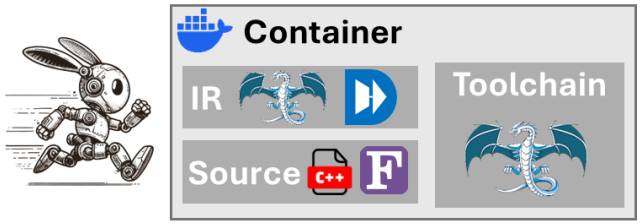
Config № 1
AVX-256, no GPU, OpenMP

Config № 2
AVX-512, CUDA, MPI

Config № 3
AVX-256, CUDA, MPI

- Compile-time definitions.
- Include paths in build directories.
- Vectorization and architecture flags.
- OpenMP flags.

XaaS IR Containers: Eliminating Redundancy



XaaS IR Container

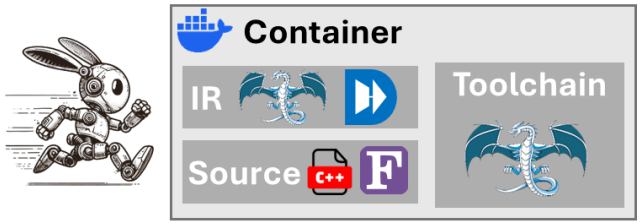
Config № 1
AVX-256, no GPU, OpenMP

Config № 2
AVX-512, CUDA, MPI

Config № 3
AVX-256, CUDA, MPI

- Compile-time definitions.
- Include paths in build directories.
- Vectorization and architecture flags.
- OpenMP flags.

XaaS IR Containers: Eliminating Redundancy



XaaS IR Container

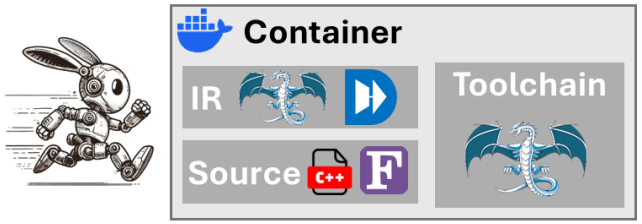
Config № 1
AVX-256, no GPU, OpenMP

Config № 2
AVX-512, CUDA, MPI

Config № 3
AVX-256, CUDA, MPI

- Compile-time definitions.
- Include paths in build directories.
- Vectorization and architecture flags.
- OpenMP flags.

XaaS IR Containers: Eliminating Redundancy



XaaS IR Container

Config № 1
AVX-256, no GPU, OpenMP

Config № 2
AVX-512, CUDA, MPI

Config № 3
AVX-256, CUDA, MPI

- Compile-time definitions.
- Include paths in build directories.
- Vectorization and architecture flags.
- OpenMP flags.

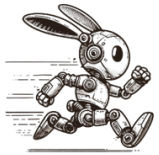
Example 1: GROMACS, CPU Vectorization (5)

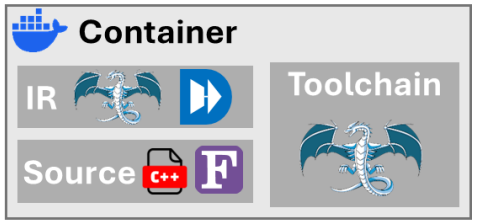
5 Build Configurations: 8710 compiled files
IR Container: 2695 IRs -> **69% reduction**

Example 2: GROMACS, CUDA + CPU Vectorization (2)

4 Build Configurations: 7052 compiled files
IR Container: 2694 IRs -> **62% reduction**

XaaS IR Containers: Build & Deploy

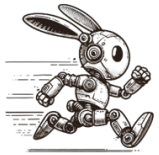


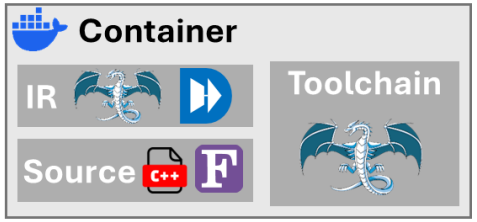


XaaS IR Container

Container Build

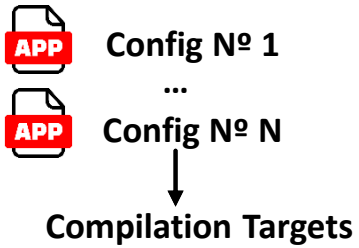
XaaS IR Containers: Build & Deploy



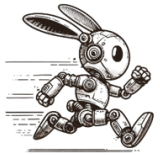


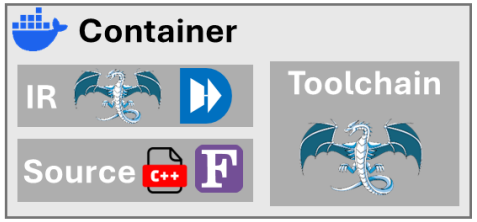
XaaS IR Container

Container Build



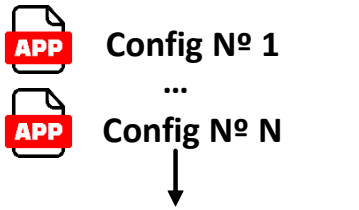
XaaS IR Containers: Build & Deploy



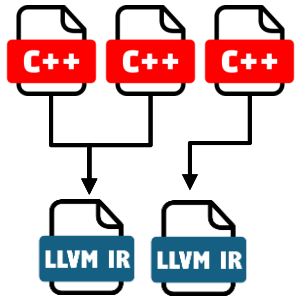


XaaS IR Container

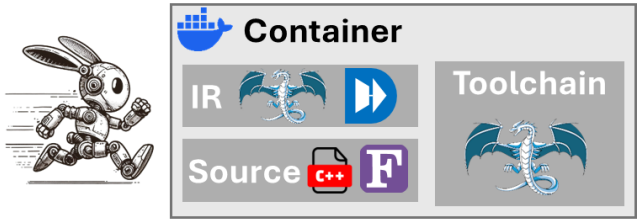
Container Build



- Preprocessing
- OpenMP Detection
- Vectorization
- Building IRs

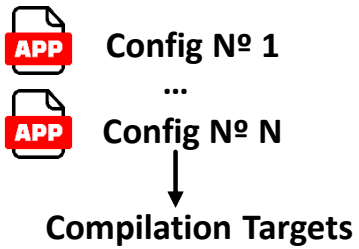


XaaS IR Containers: Build & Deploy

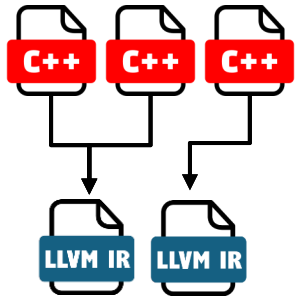


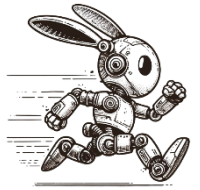
XaaS IR Container

Container Build

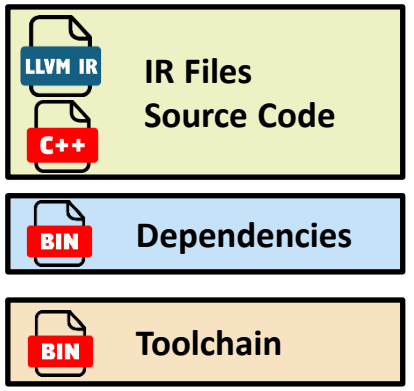


- Preprocessing
- OpenMP Detection
- Vectorization
- Building IRs

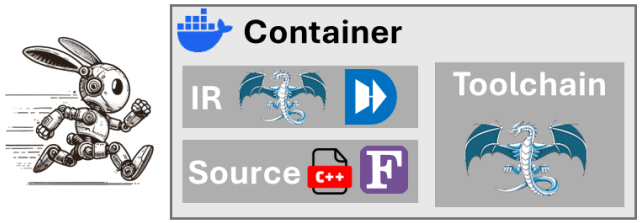




IR Container

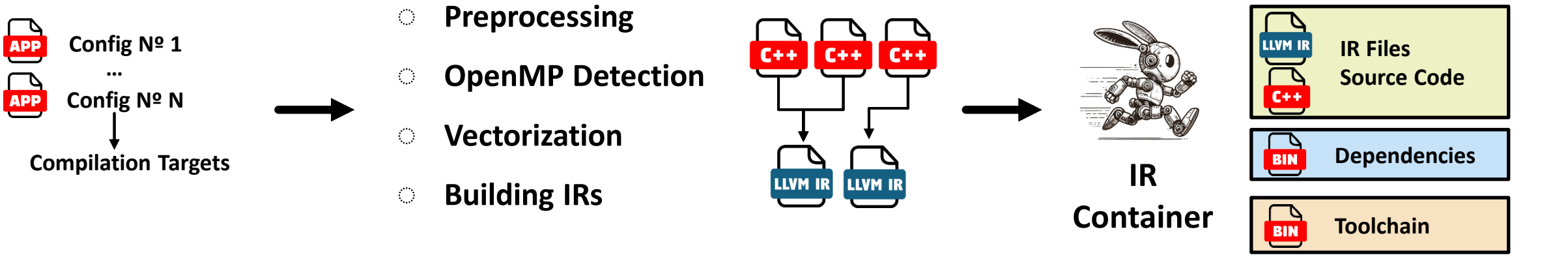


XaaS IR Containers: Build & Deploy



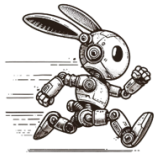
XaaS IR Container

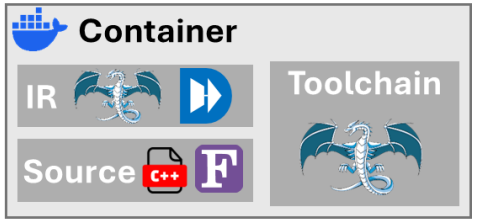
Container Build



Container Deployment

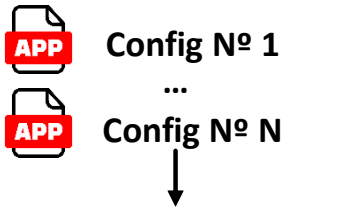
XaaS IR Containers: Build & Deploy



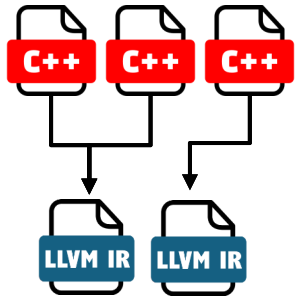


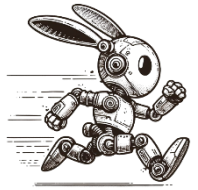
XaaS IR Container

Container Build

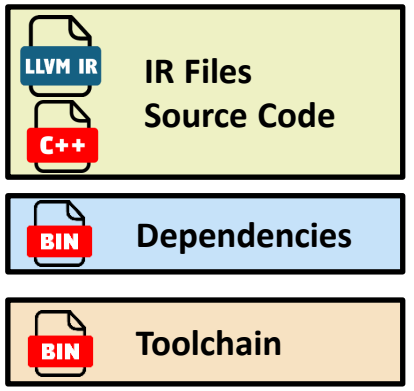


- Preprocessing
- OpenMP Detection
- Vectorization
- Building IRs

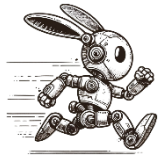




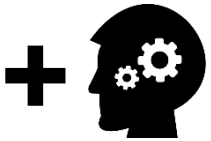
IR Container



Container Deployment

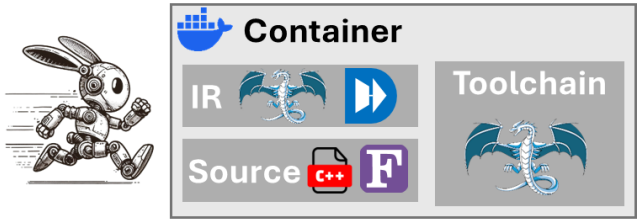


XaaS
IR Container



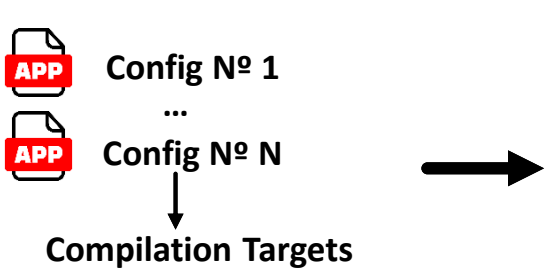
Specialization
Selection

XaaS IR Containers: Build & Deploy

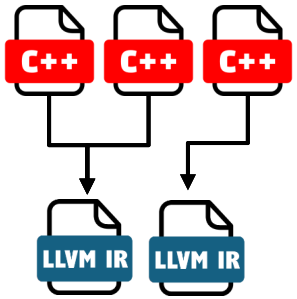


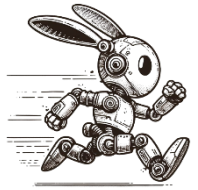
XaaS IR Container

Container Build

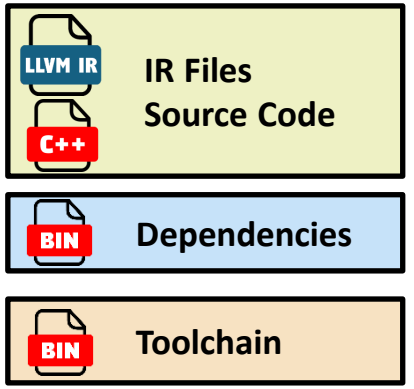


- Preprocessing
- OpenMP Detection
- Vectorization
- Building IRs





IR Container

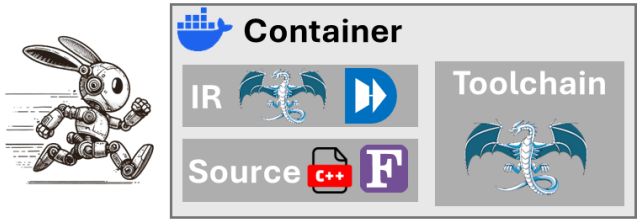


Container Deployment

Specialization Selection

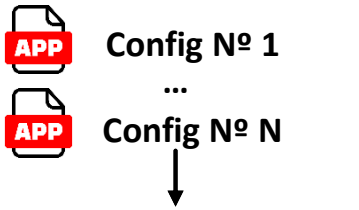
LLVM Vectorization, Lowering

XaaS IR Containers: Build & Deploy

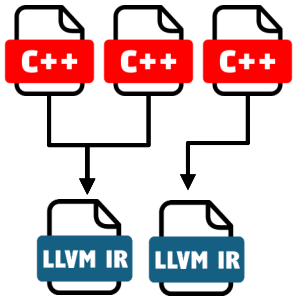


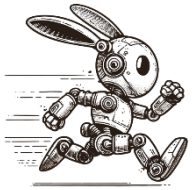
XaaS IR Container

Container Build

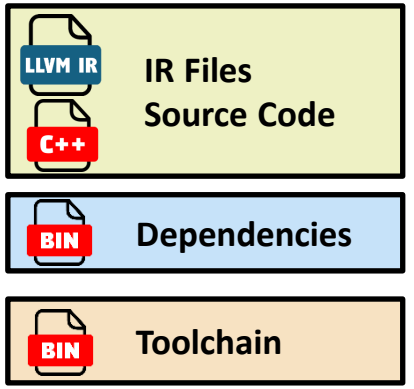


- Preprocessing
- OpenMP Detection
- Vectorization
- Building IRs





IR Container



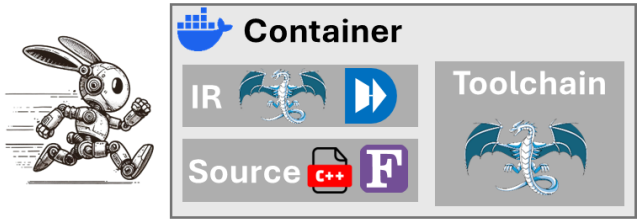
Container Deployment

Specialization
Selection

LLVM Vectorization,
Lowering

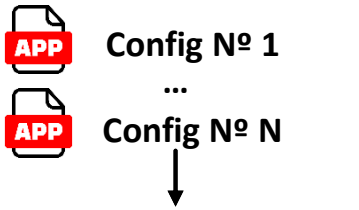
Linking,
Installation

XaaS IR Containers: Build & Deploy

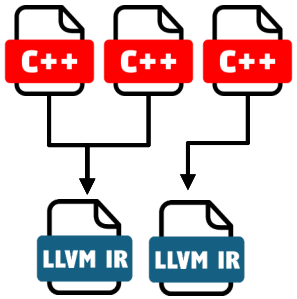


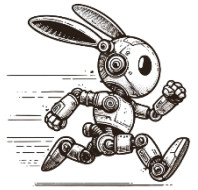
XaaS IR Container

Container Build

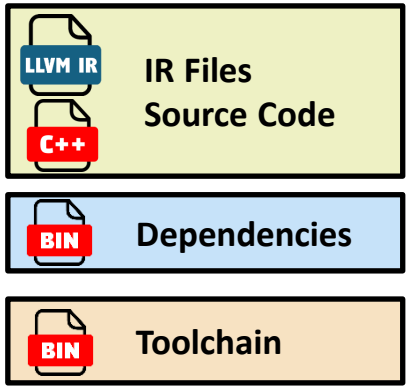


- Preprocessing
- OpenMP Detection
- Vectorization
- Building IRs

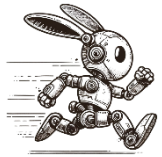




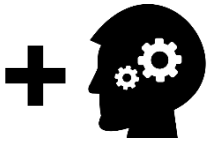
IR Container



Container Deployment



XaaS IR Container



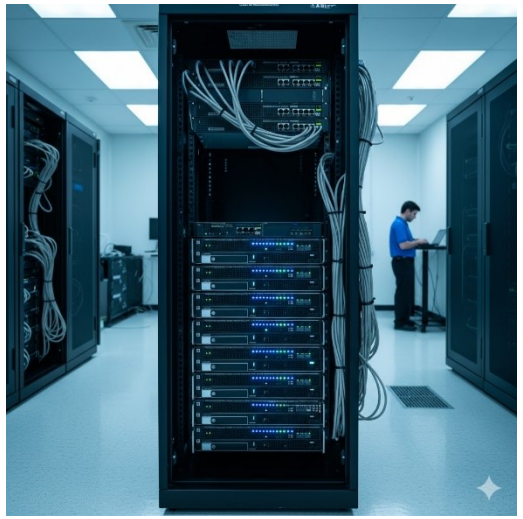
Specialization Selection

LLVM Vectorization, Lowering

Linking, Installation

Deployed IR Container

CSCS Ault



Intel/AMD CPU, NVIDIA GPU

CSCS Alps

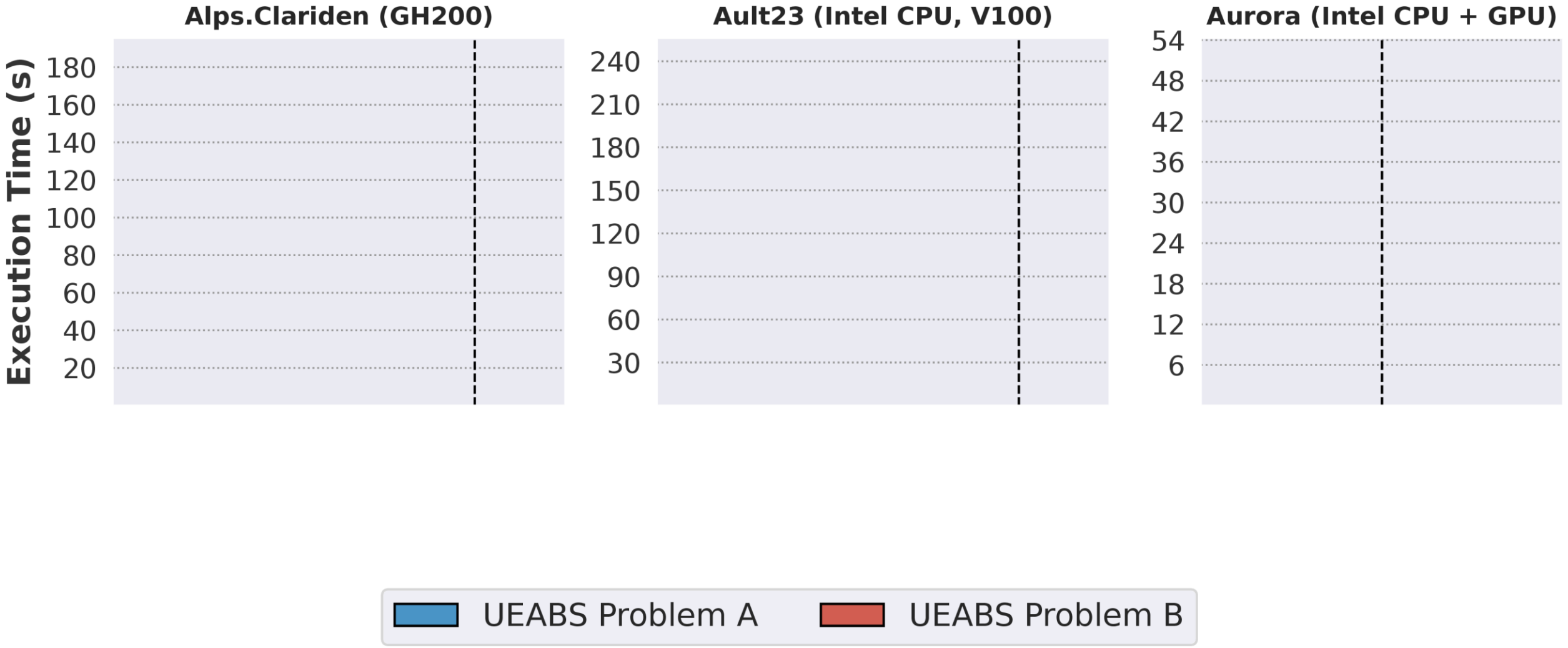
ARM CPU, NVIDIA GPU

ALCF Aurora

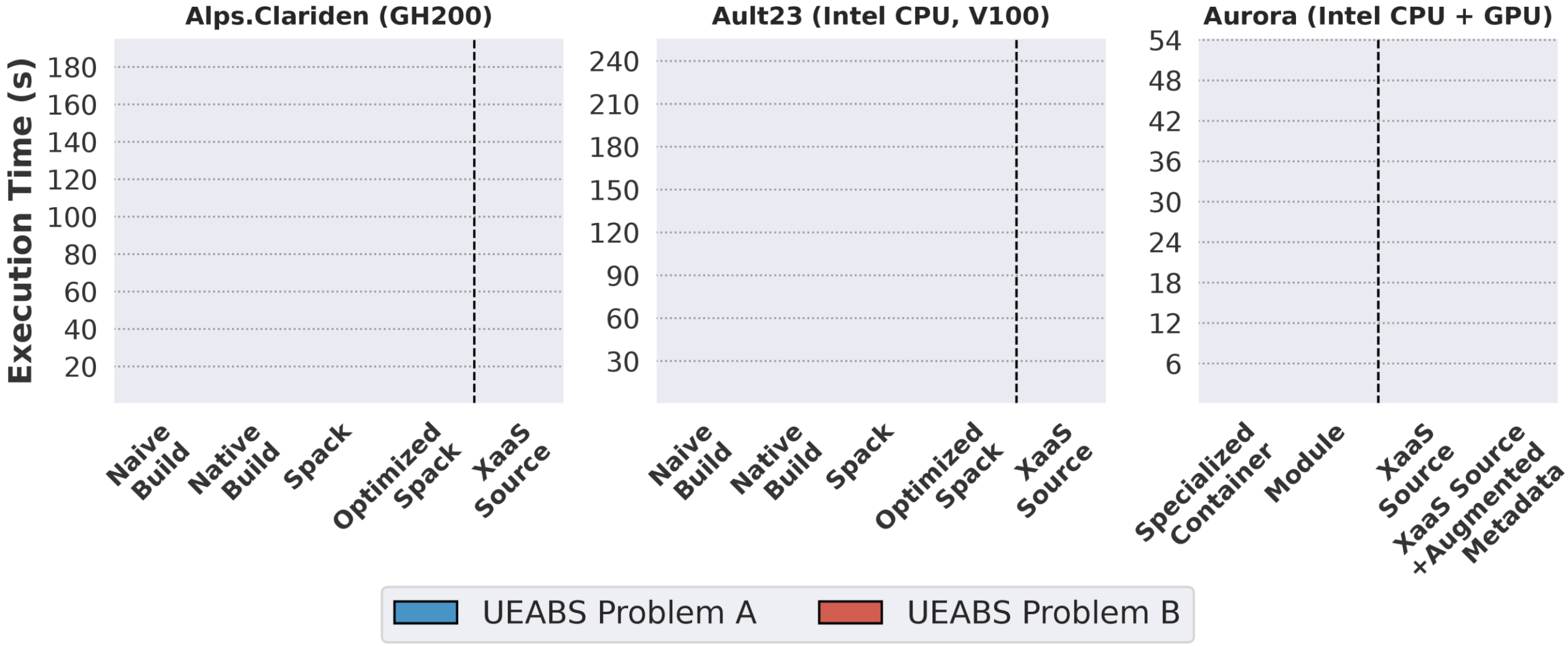
Intel CPU, Intel GPU

Can XaaS Source and IR Containers provide performance portability?

Evaluation – GROMACS in XaaS Source Containers

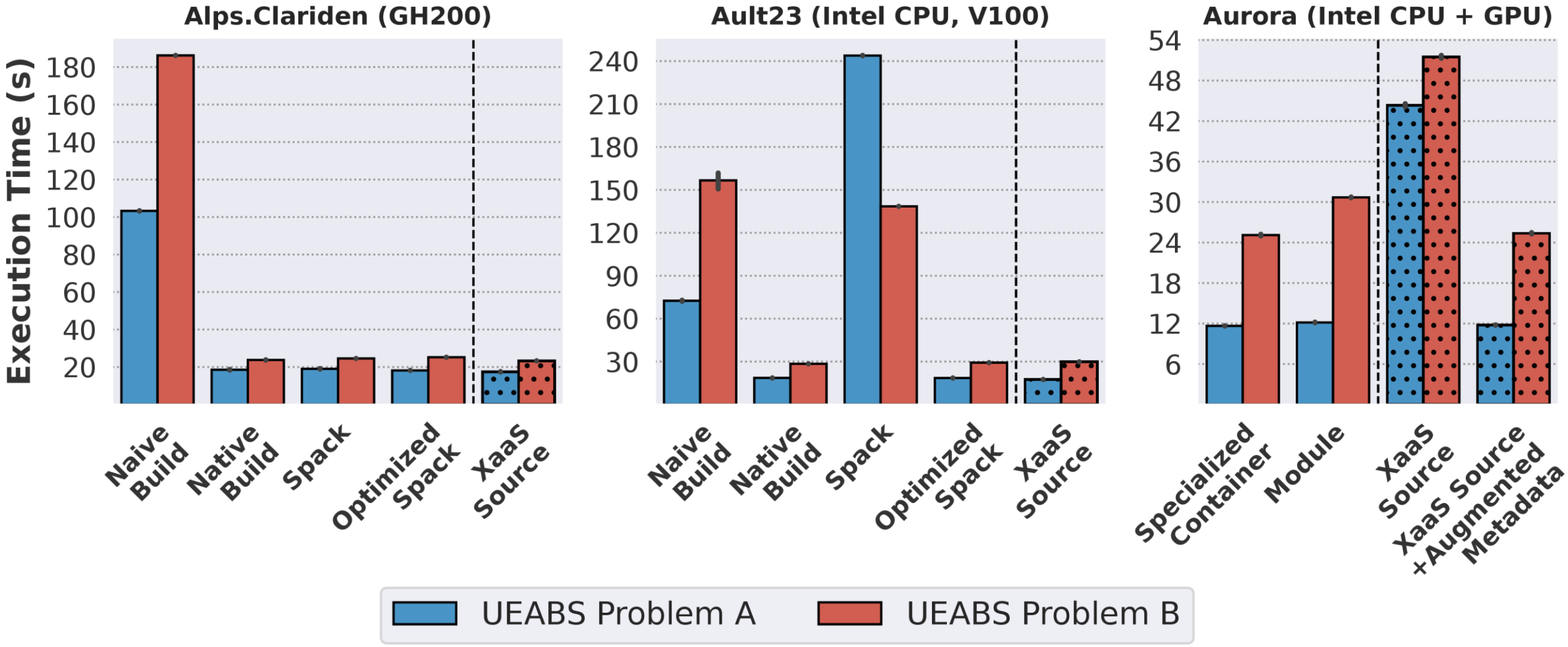


Evaluation – GROMACS in XaaS Source Containers



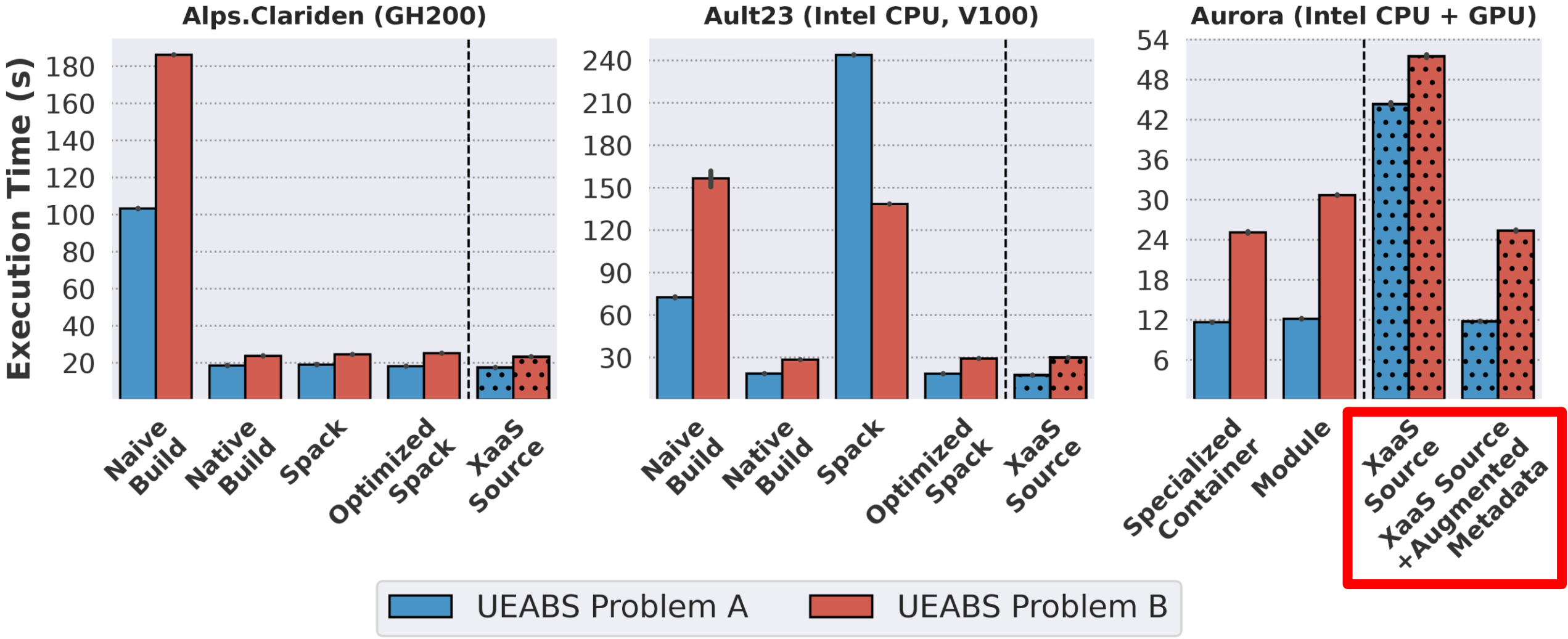
Baselines (depending on availability): “naïve” build, specialized native builds, specialized containers, system modules, Spack.

Evaluation – GROMACS in XaaS Source Containers



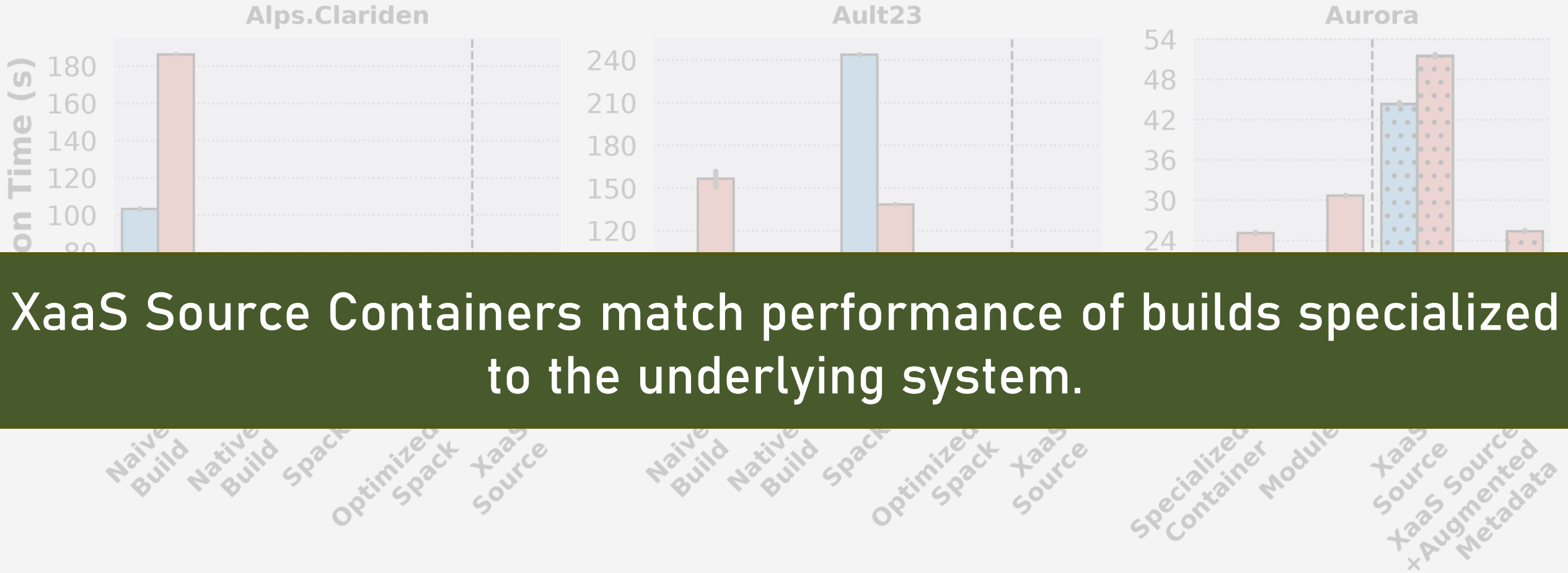
Baselines (depending on availability): “naïve” build, specialized native builds, specialized containers, system modules, Spack.

Evaluation – GROMACS in XaaS Source Containers



Baselines (depending on availability): "naïve" build, specialized native builds, specialized containers, system modules, Spack.

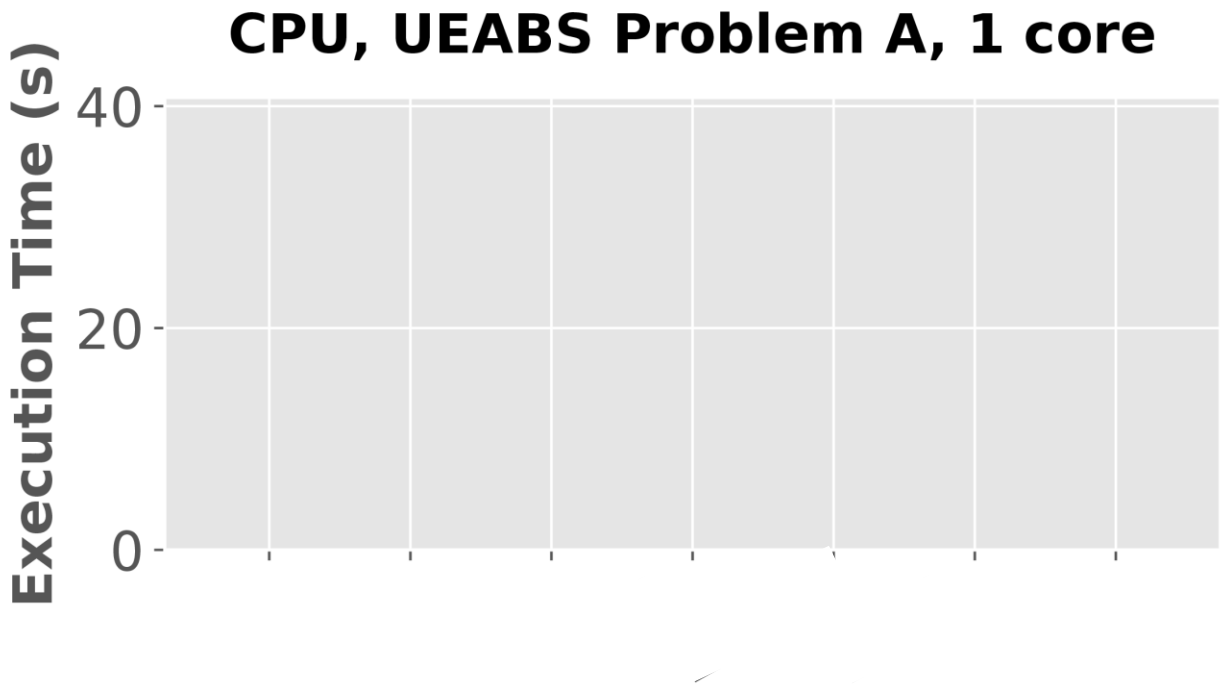
Evaluation – GROMACS in XaaS Source Containers

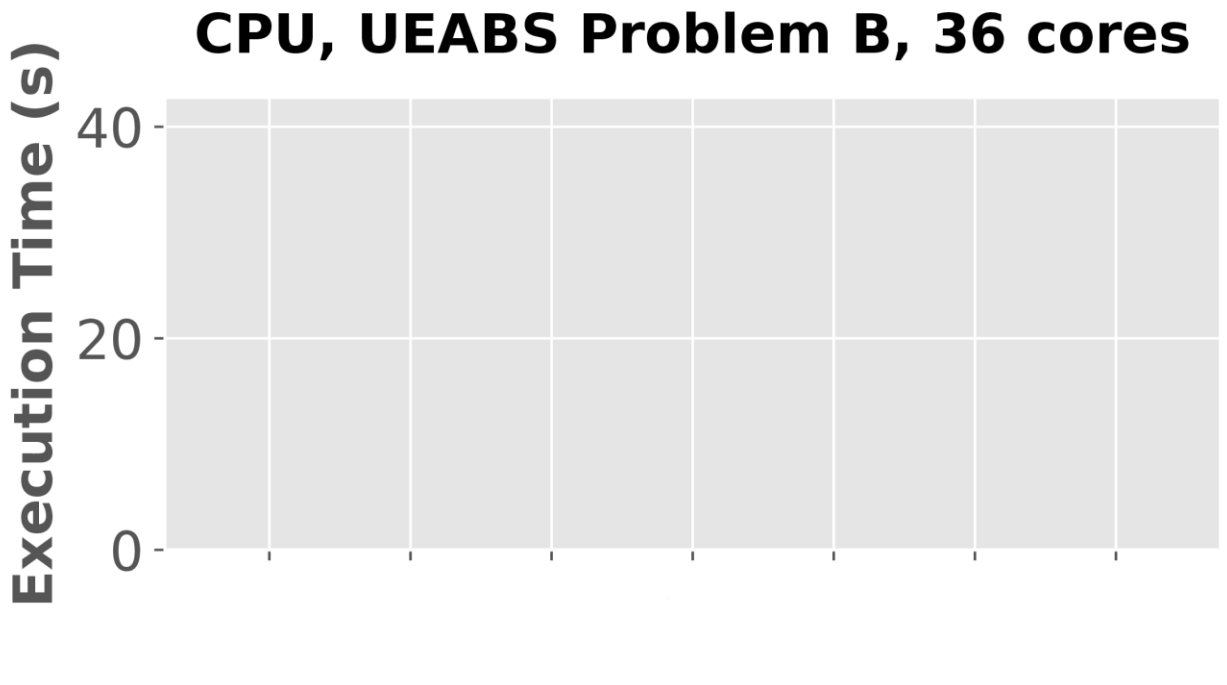


XaaS Source Containers match performance of builds specialized to the underlying system.

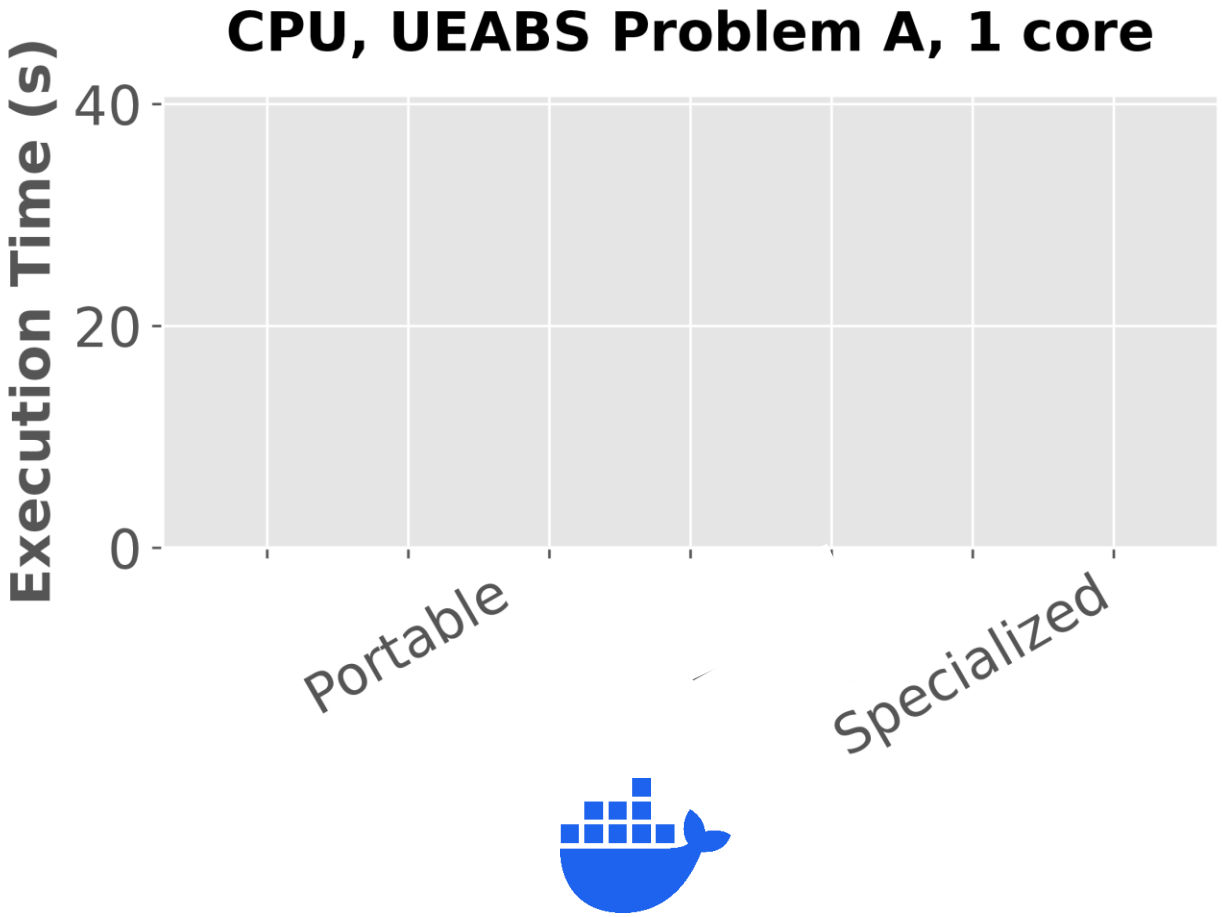
Baselines (depending on availability): “naïve” build, specialized native builds, specialized containers, system modules, Spack.

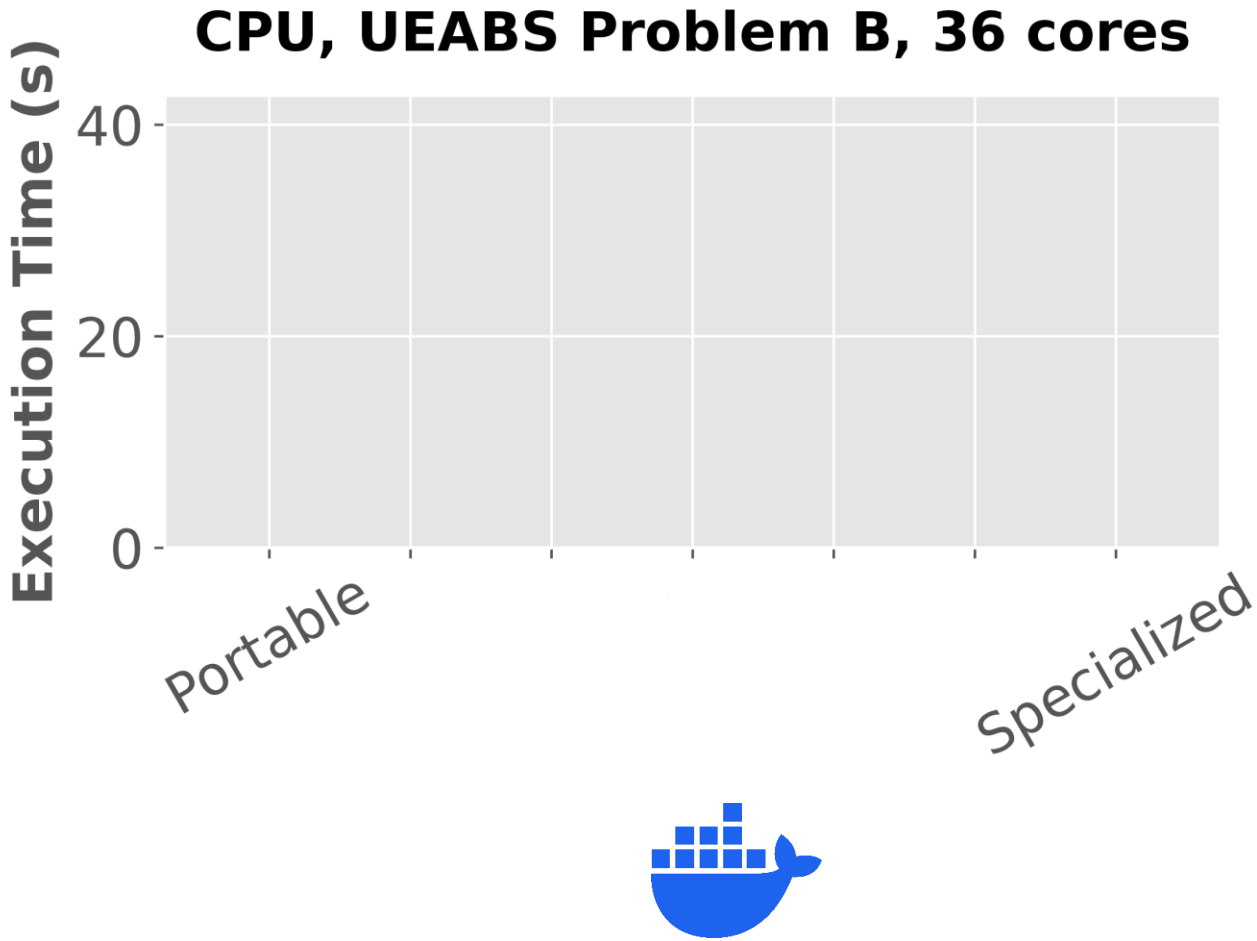
Evaluation – XaaS IR Containers with CPU Vectorization





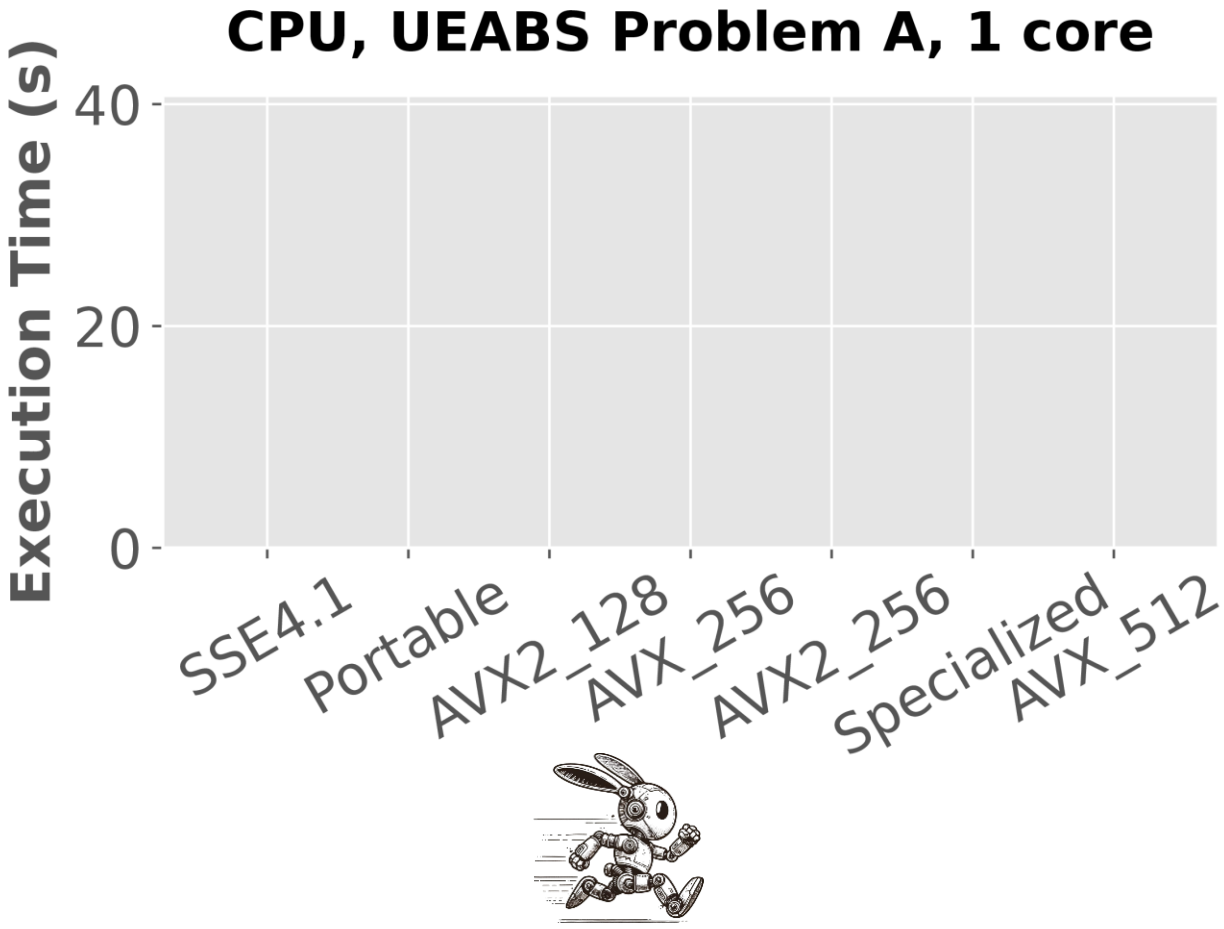
Evaluation – XaaS IR Containers with CPU Vectorization

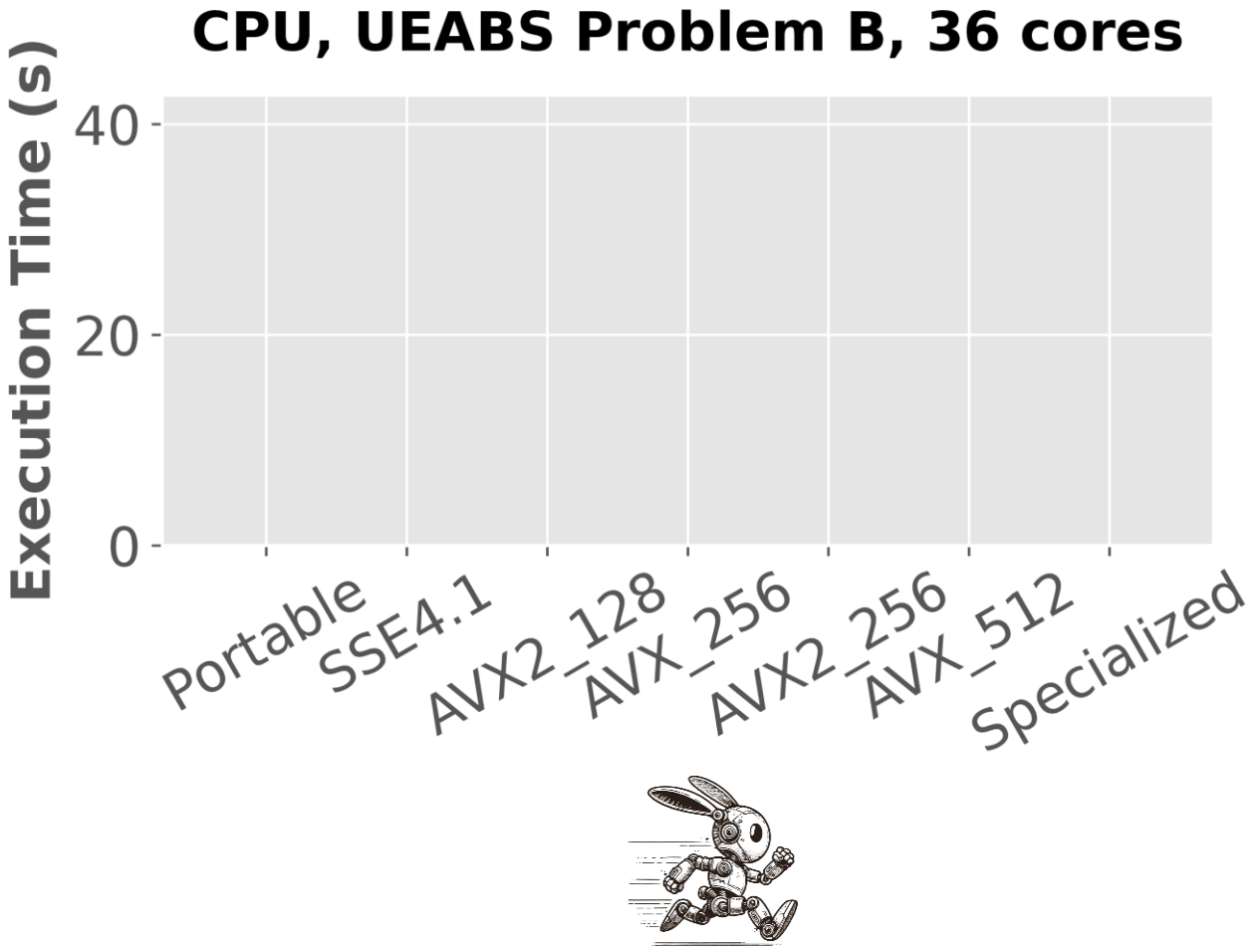




Baselines: portable (SSE4.1) and specialized (AVX-512) Docker containers.

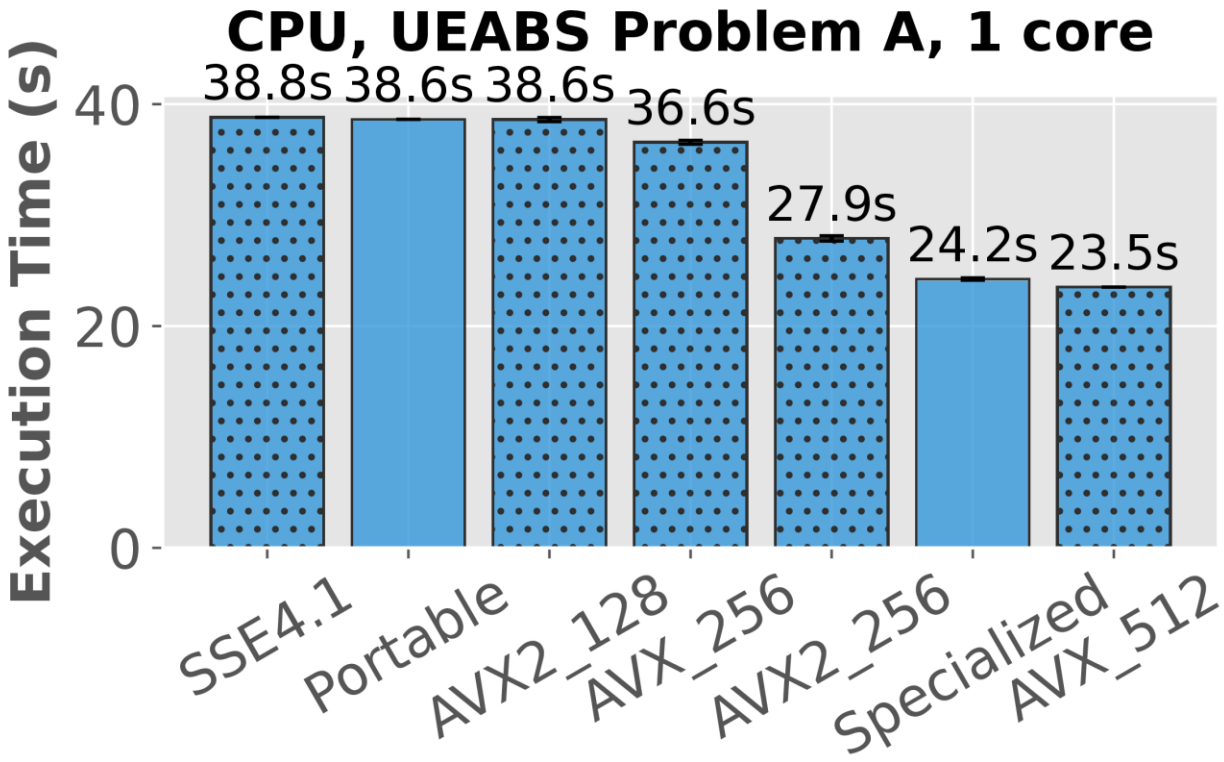
Evaluation – XaaS IR Containers with CPU Vectorization

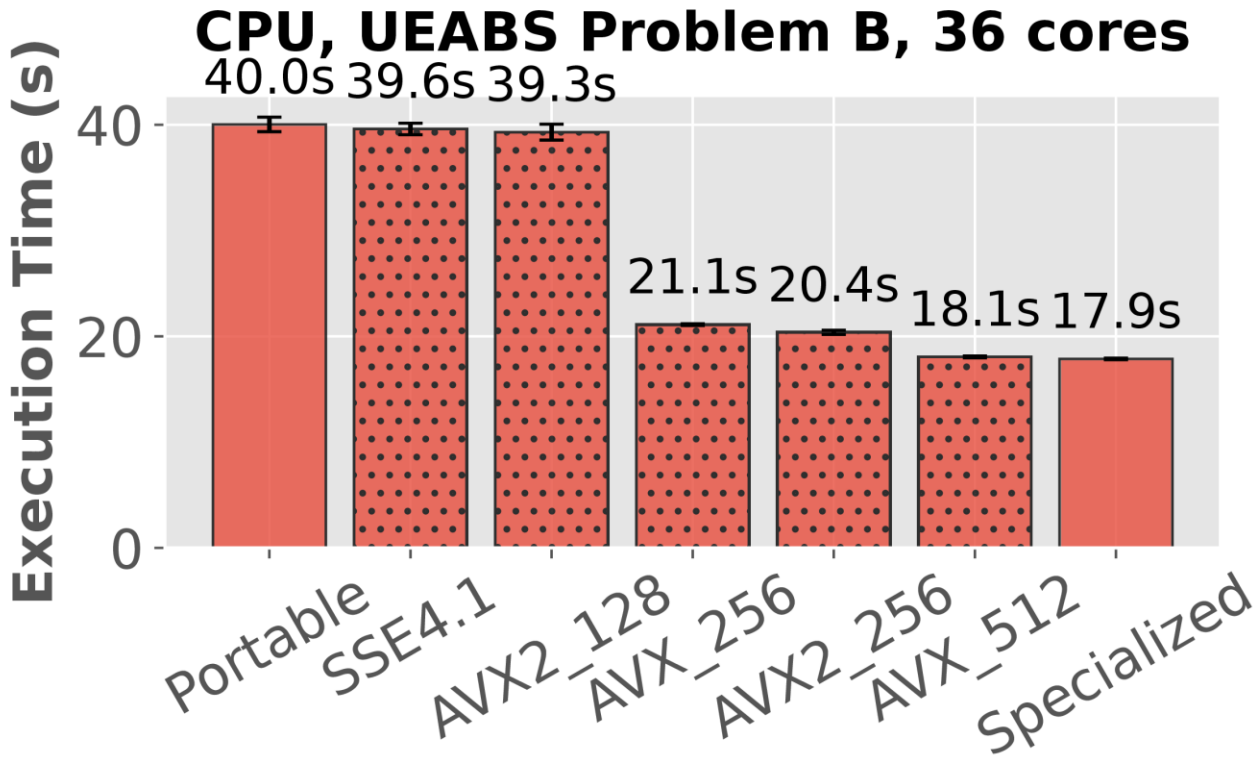




Baselines: portable (SSE4.1) and specialized (AVX-512) Docker containers.

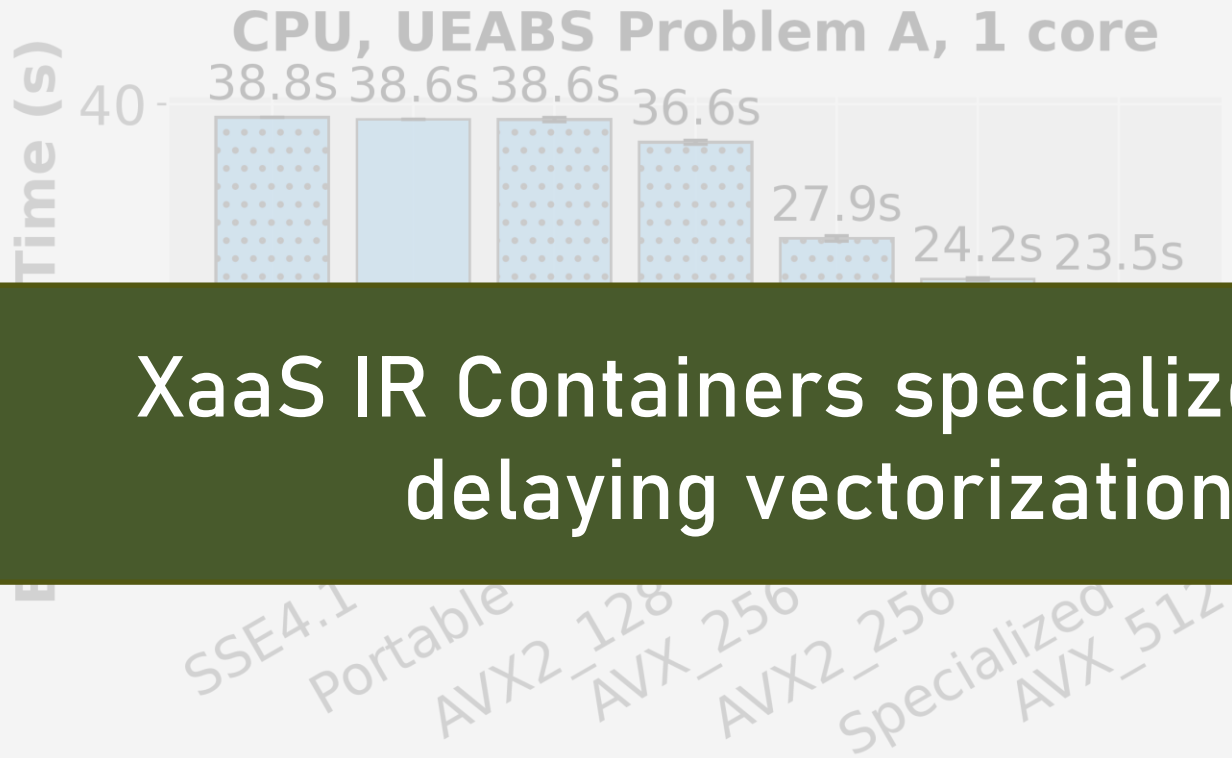
Evaluation – XaaS IR Containers with CPU Vectorization

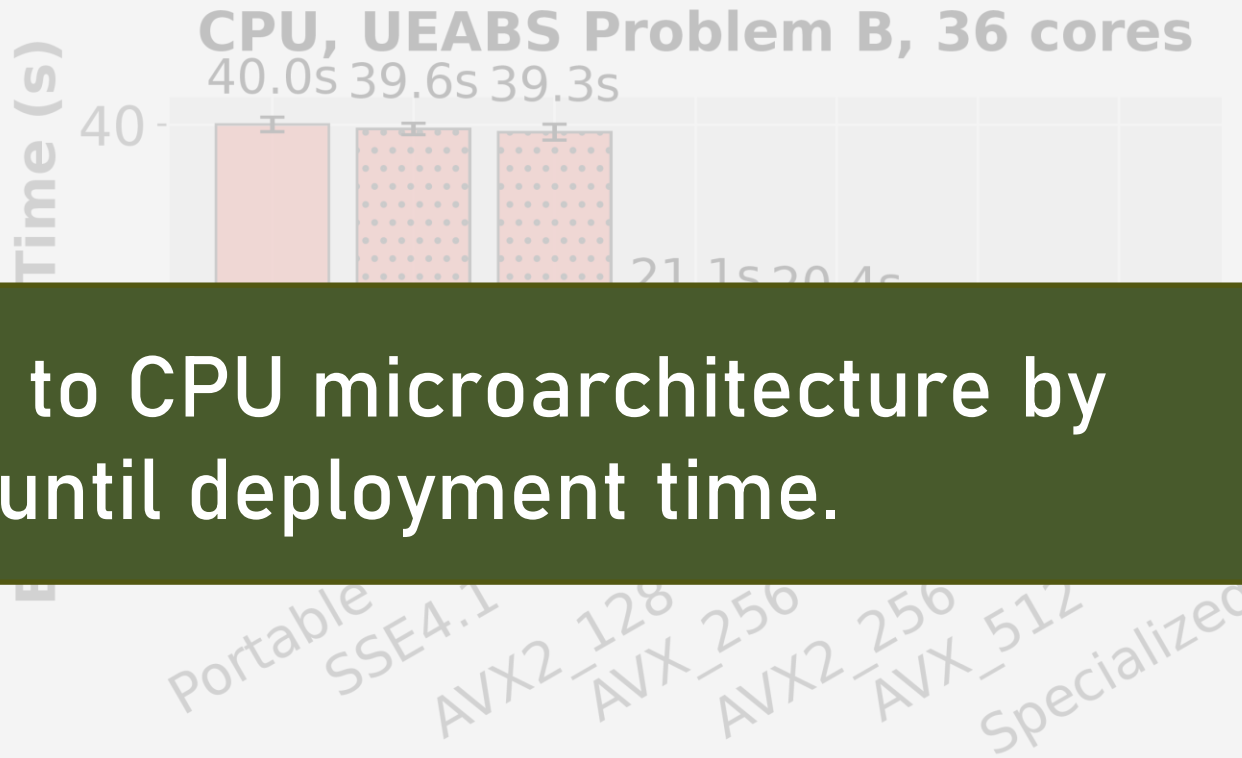




Baselines: portable (SSE4.1) and specialized (AVX-512) Docker containers.

Evaluation – XaaS IR Containers with CPU Vectorization

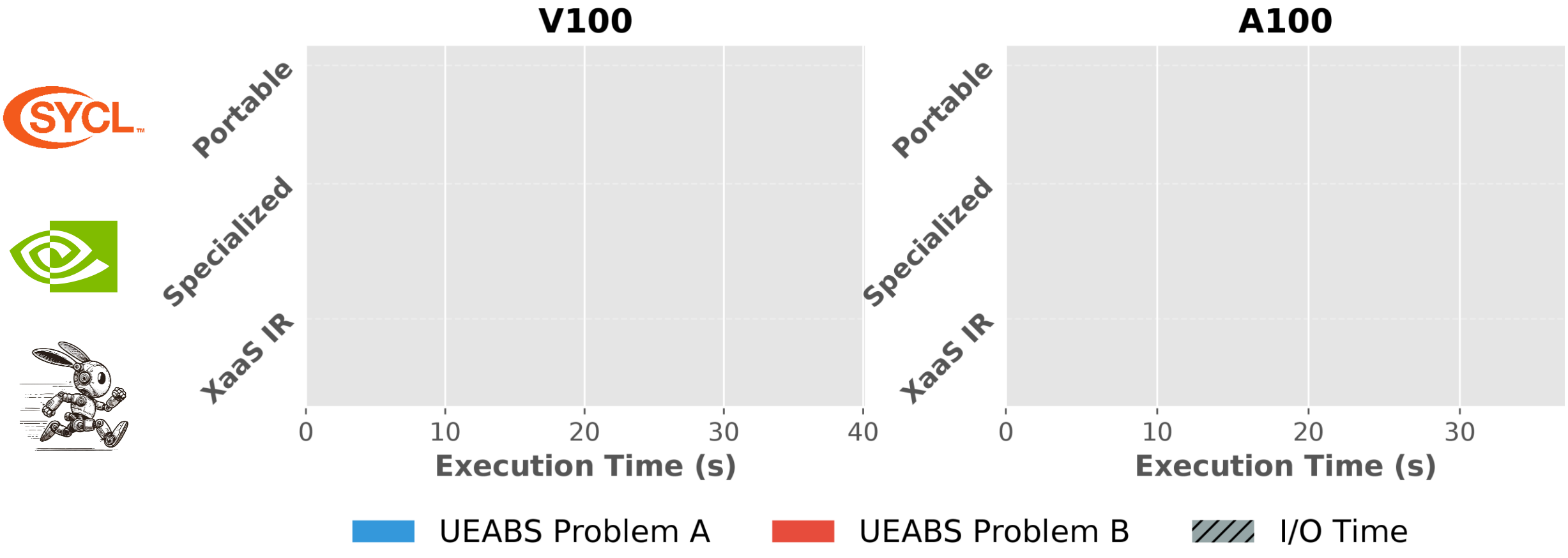




XaaS IR Containers specialize to CPU microarchitecture by delaying vectorization until deployment time.

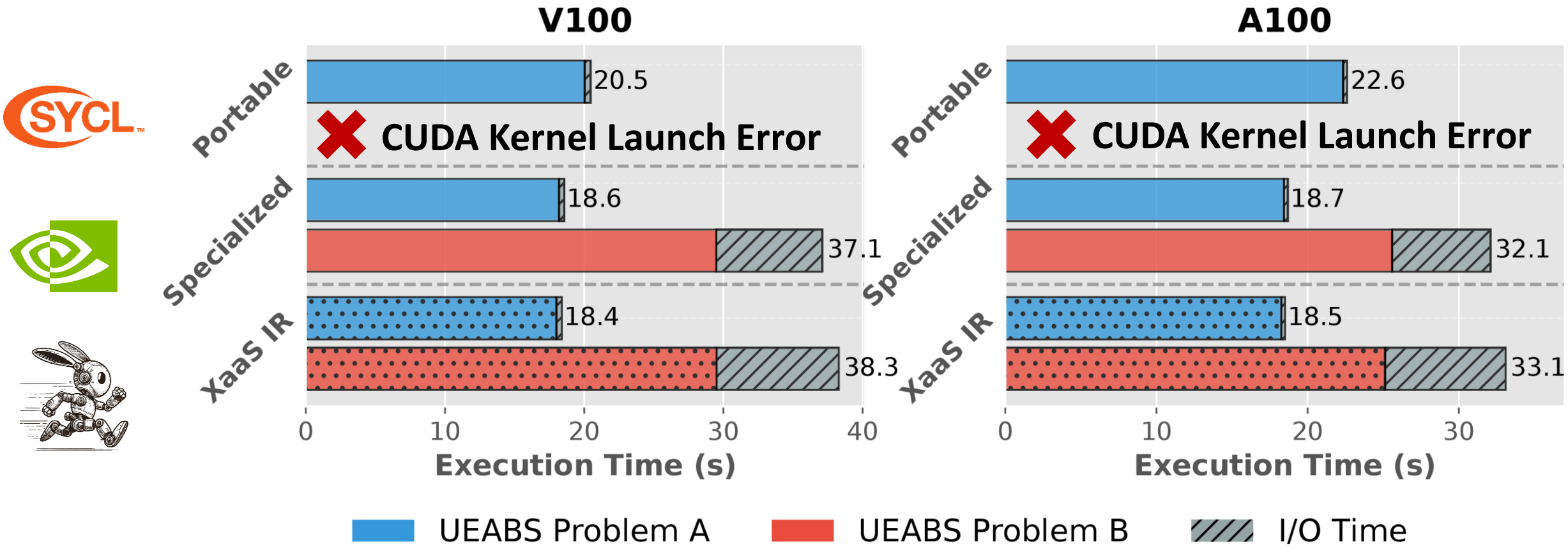
Baselines: portable (SSE4.1) and specialized (AVX-512) Docker containers.

Evaluation – XaaS IR Containers with CUDA Support



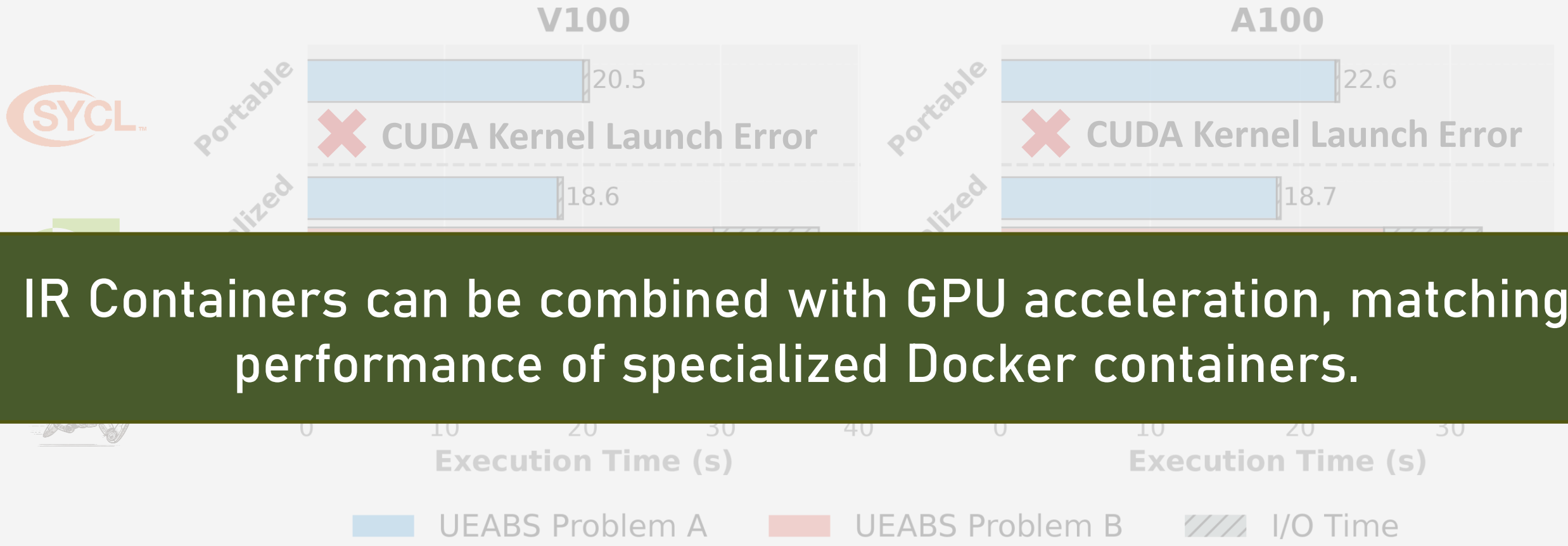
Baselines: specialized Docker containers with CUDA, AVX-512 (V100) or AVX-256 (A100).
Portable containers with SYCL + CUDA, compiled with Intel's icpx compiler.

Evaluation – XaaS IR Containers with CUDA Support



Baselines: specialized Docker containers with CUDA, AVX-512 (V100) or AVX-256 (A100).
Portable containers with SYCL + CUDA, compiled with Intel's icpx compiler.

Evaluation – XaaS IR Containers with CUDA Support



IR Containers can be combined with GPU acceleration, matching performance of specialized Docker containers.

Baselines: specialized Docker containers with CUDA, AVX-512 (V100) or AVX-256 (A100).
Portable containers with SYCL + CUDA, compiled with Intel's icpx compiler.

Conclusions

More of SPCL's research:

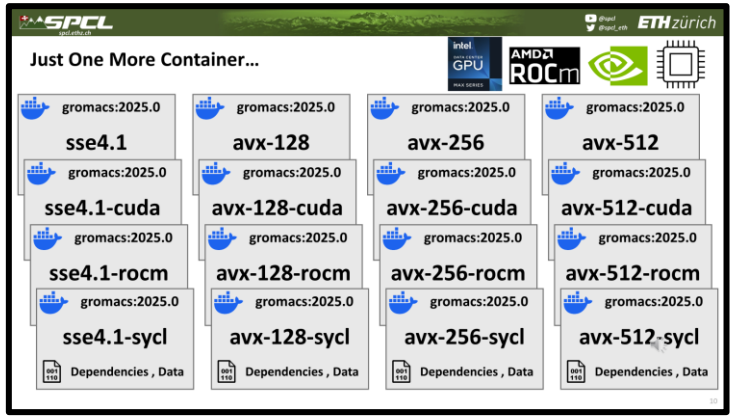
youtube.com/@spcl — 240+ Talks

twitter.com/spcl_eth — 1.7K+ Followers

github.com/spcl — 7.2K+ Stars

… or spcl.ethz.ch

Conclusions



More of SPCL's research:

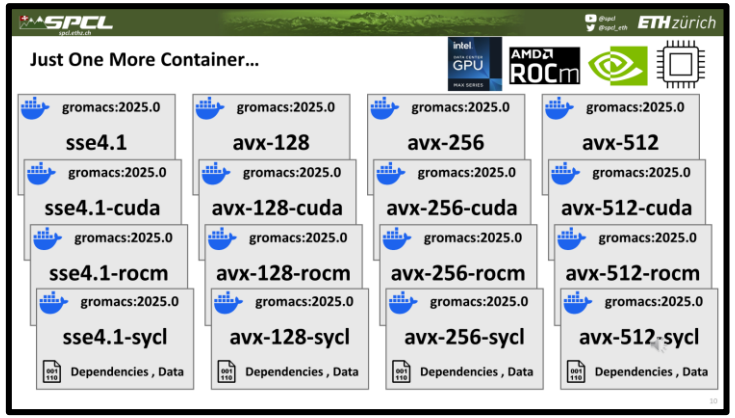
youtube.com/@spcl — 240+ Talks

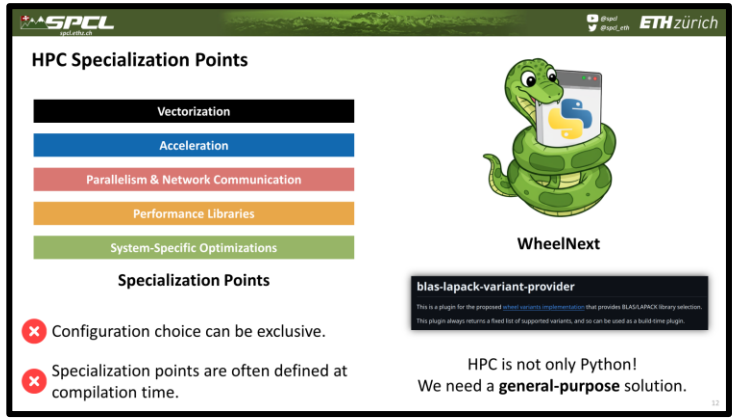
twitter.com/spcl_eth — 1.7K+ Followers

github.com/spcl — 7.2K+ Stars

… or spcl.ethz.ch

Conclusions





More of SPCL's research:

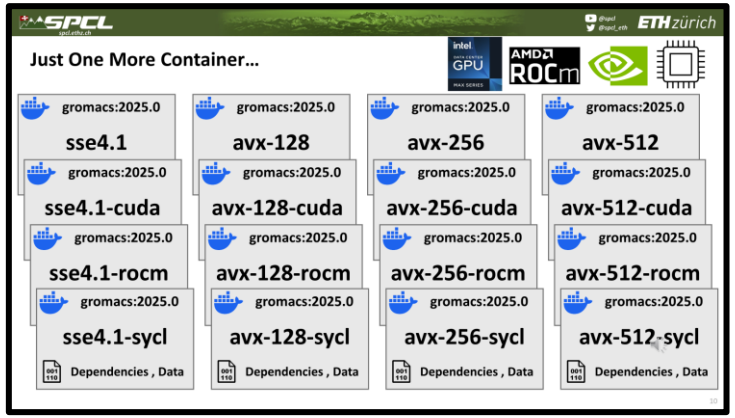
youtube.com/@spcl — 240+ Talks

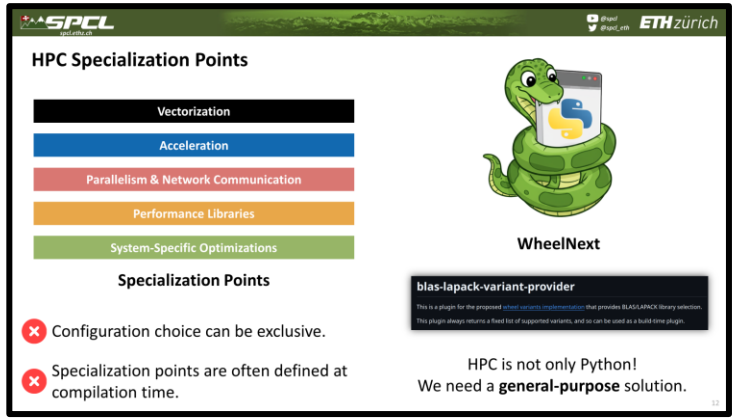
twitter.com/spcl_eth — 1.7K+ Followers

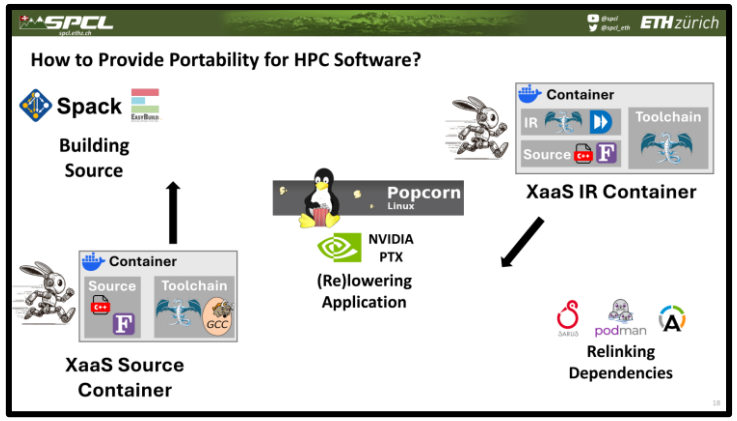
github.com/spcl — 7.2K+ Stars

… or spcl.ethz.ch

Conclusions







More of SPCL's research:

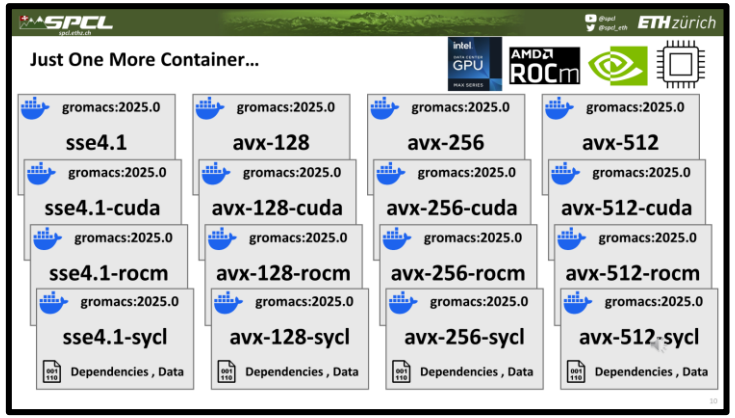
youtube.com/@spcl — 240+ Talks

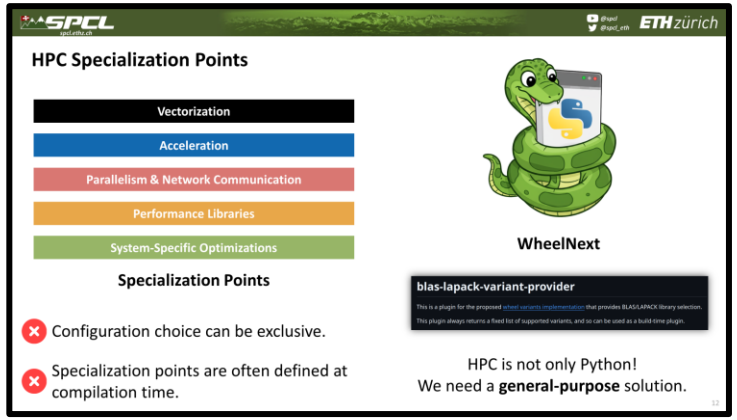
twitter.com/spcl_eth — 1.7K+ Followers

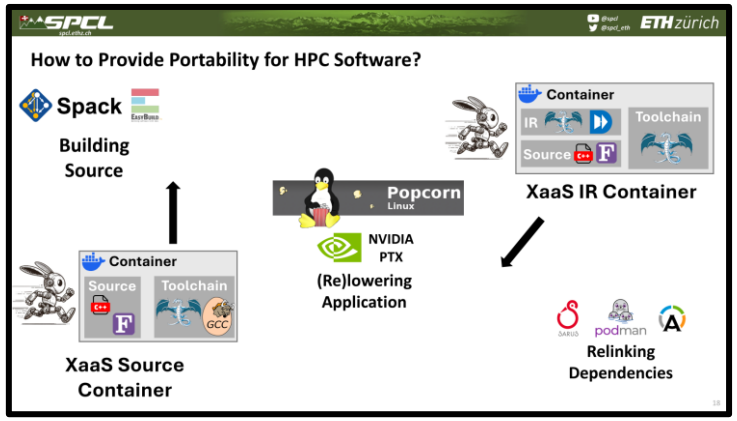
github.com/spcl — 7.2K+ Stars

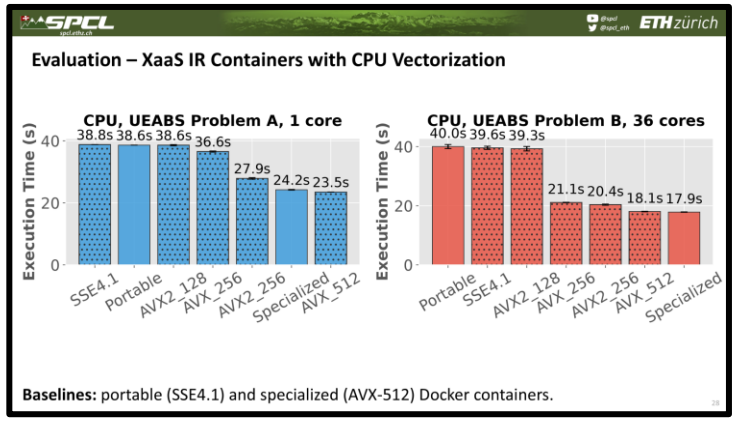
… or spcl.ethz.ch

Conclusions









More of SPCL's research:

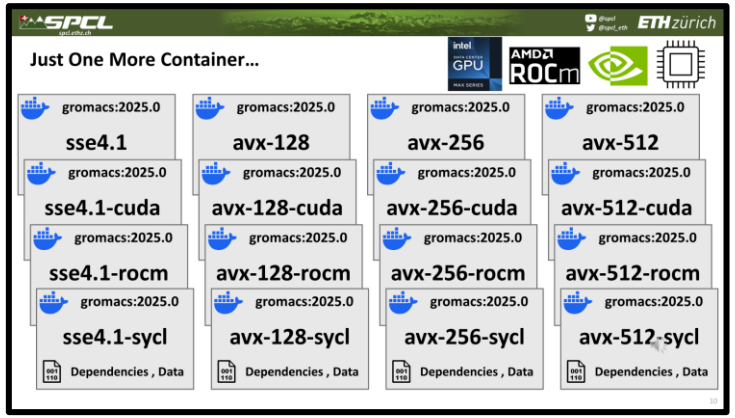
youtube.com/@spcl — 240+ Talks

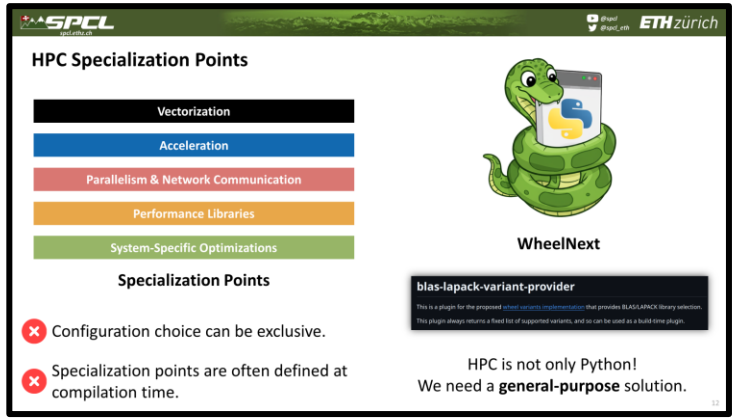
twitter.com/spcl_eth — 1.7K+ Followers

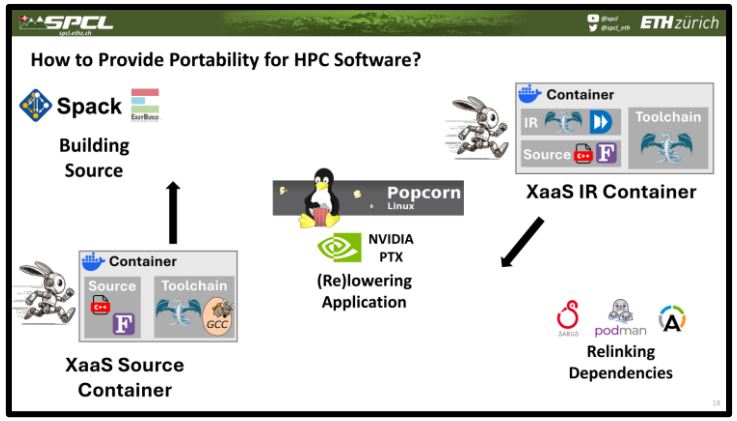
github.com/spcl — 7.2K+ Stars

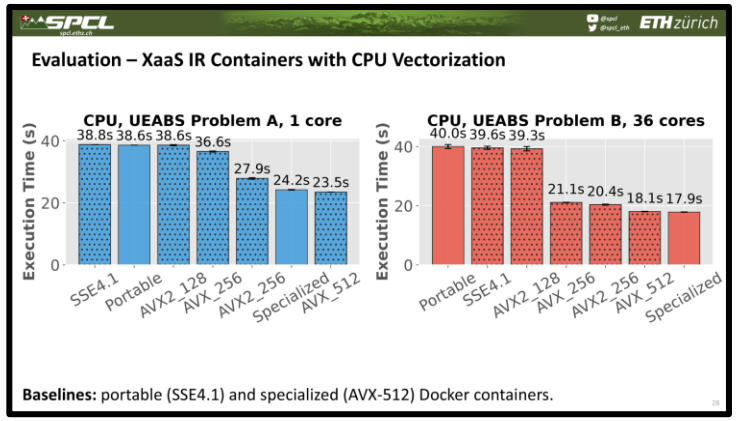
… or spcl.ethz.ch

Conclusions









More of SPCL's research:

youtube.com/@spcl — 240+ Talks

twitter.com/spcl_eth — 1.7K+ Followers

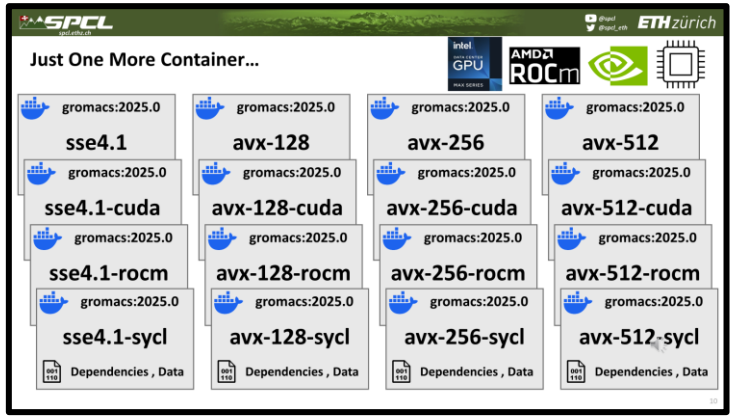
github.com/spcl — 7.2K+ Stars

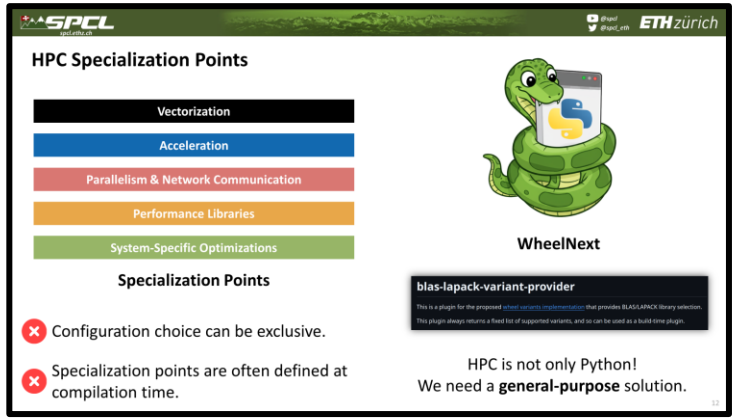
… or spcl.ethz.ch

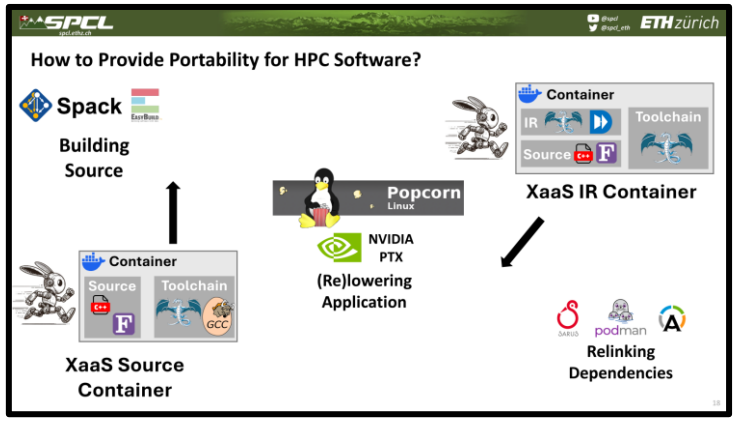
GitHub

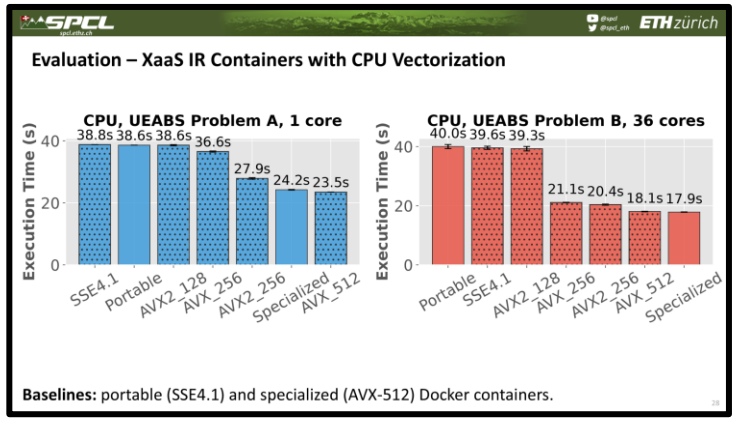
spcl/XaaS-Containers

Conclusions









More of SPCL's research:

youtube.com/@spcl — 240+ Talks

twitter.com/spcl_eth — 1.7K+ Followers

github.com/spcl — 7.2K+ Stars

… or spcl.ethz.ch

GitHub

Artifact

spcl/XaaS-Containers