Lightweight Isolation for HPC Applications

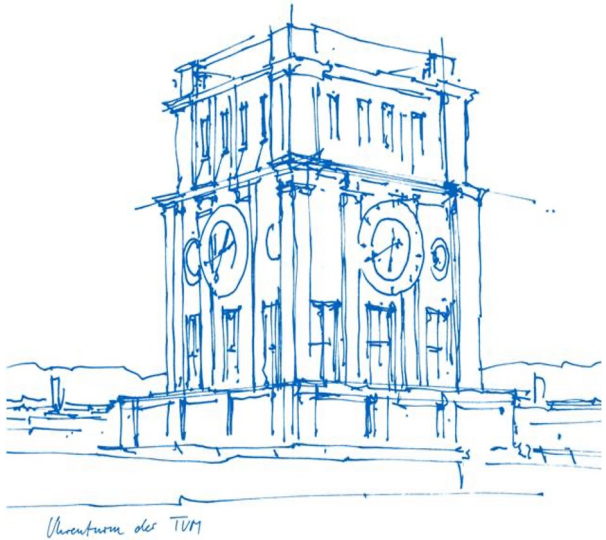
CANOPIE HPC Workshop@2023

Mohak Chadha: mohak.chadha@tum.de

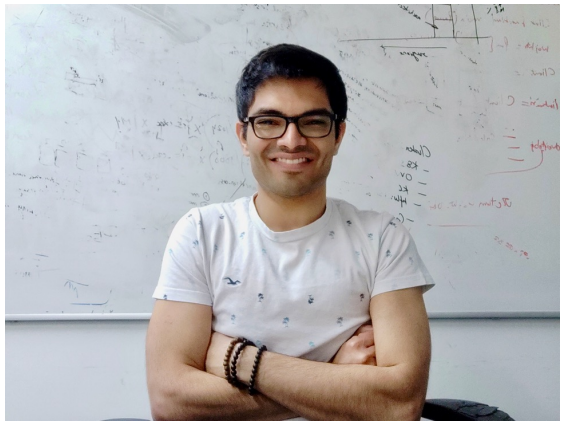
Chair of Computer Architecture and Parallel Systems (CAPS)

Technical University of Munich

Germany



About Me



Final-year PhD candidate at TUM

Focusing on: *Serverless Computing*

Domains: Cloud Computing, High Performance Computing, Parallel Computing, Systems for ML

Website:

SCAN ME

Table of Contents

Rise of Containers in HPC

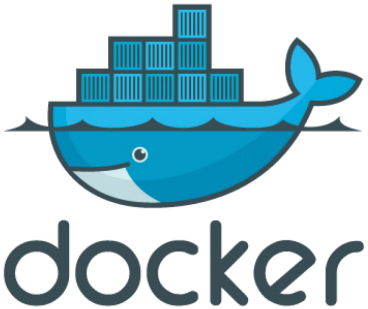
[SC'17]

[PloS one'17]

[ISC'19]

[J. Phys.'17]

[Redhat, ISC'19]



Why containers in HPC?

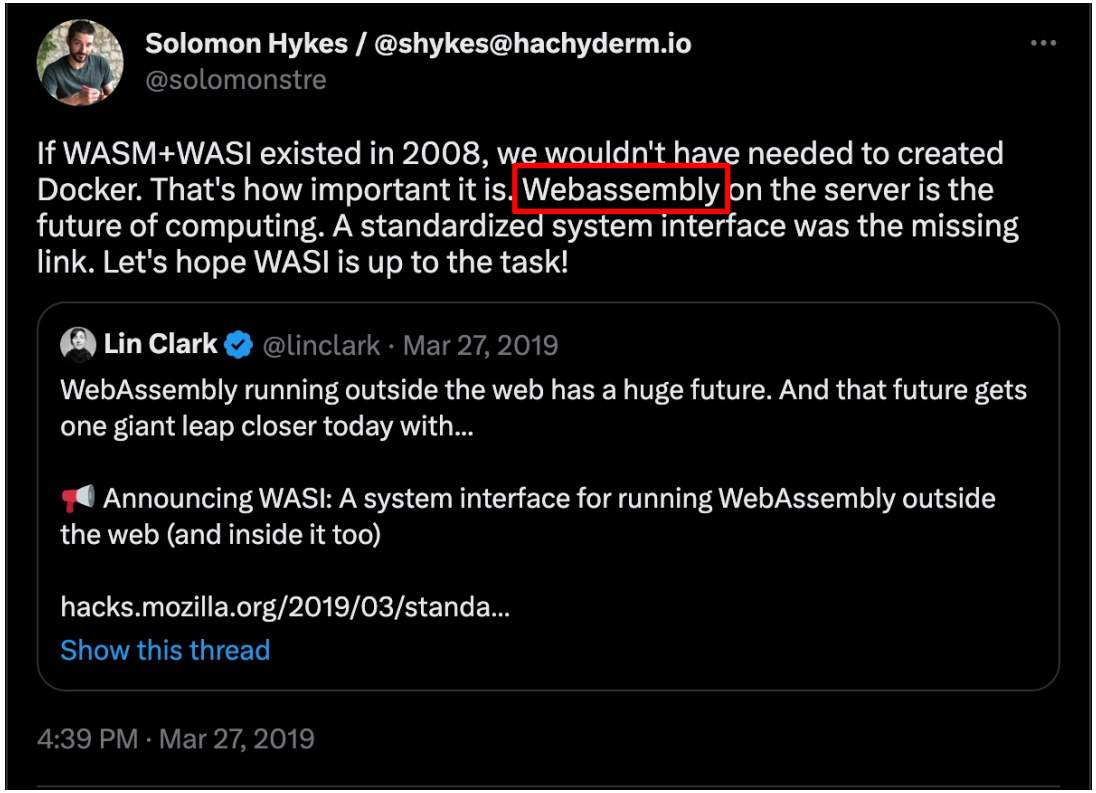
Enabling custom user-defined software stacks

Challenges in Container-based HPC application Development

1. Root privileges for running containers
2. Increasing heterogeneity of HPC nodes
3. Requirement for special networking libraries or compilers
4. Building high-performant application container images.

Only, 8% of the total jobs at NERSC use containers [2018]

Alternative to containers?

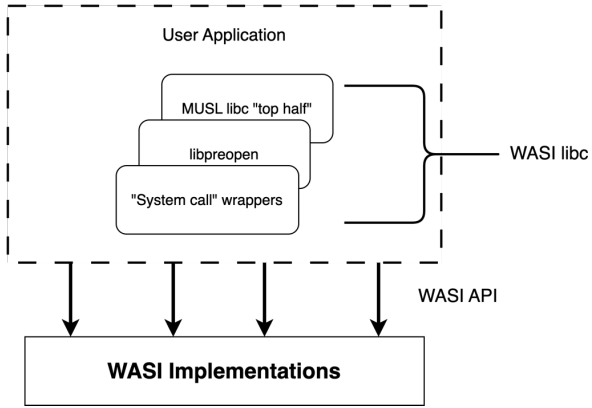


Introduction: WebAssembly (Wasm)

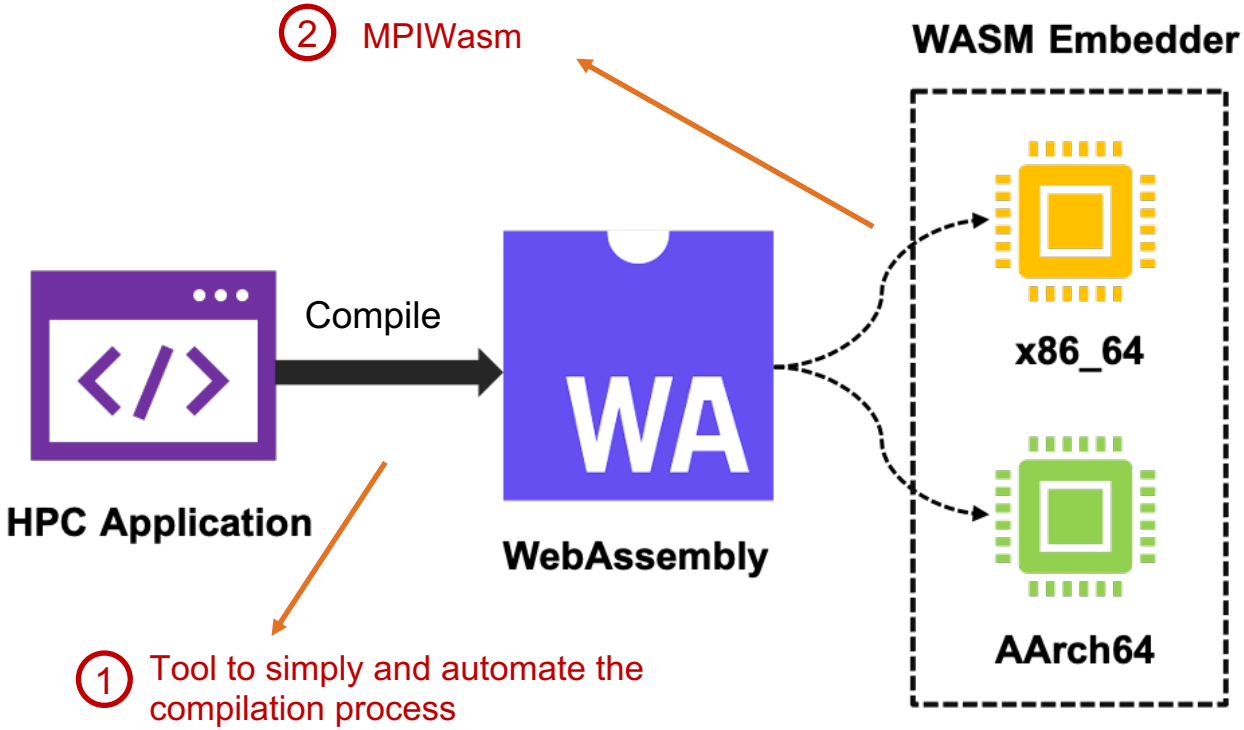
- Binary format, with alternative human-readable text representation
- Virtual ISA
- Linear 32-bit memory space
- Lightweight userspace isolation mechanism
- Import/export system for granting capabilities

Introduction: WASI: Wasm System Interface

- Standardized non-Web system-oriented API for Wasm
- Capability-oriented
- Portable
- Custom libc implementation integrated into WASI-SDK



What we did?


2
MPIWasm
WASM Embedder
Compile
x86_64
HPC Application
WebAssembly
AArch64
1
Tool to simply and automate the compilation process

MPIWasm

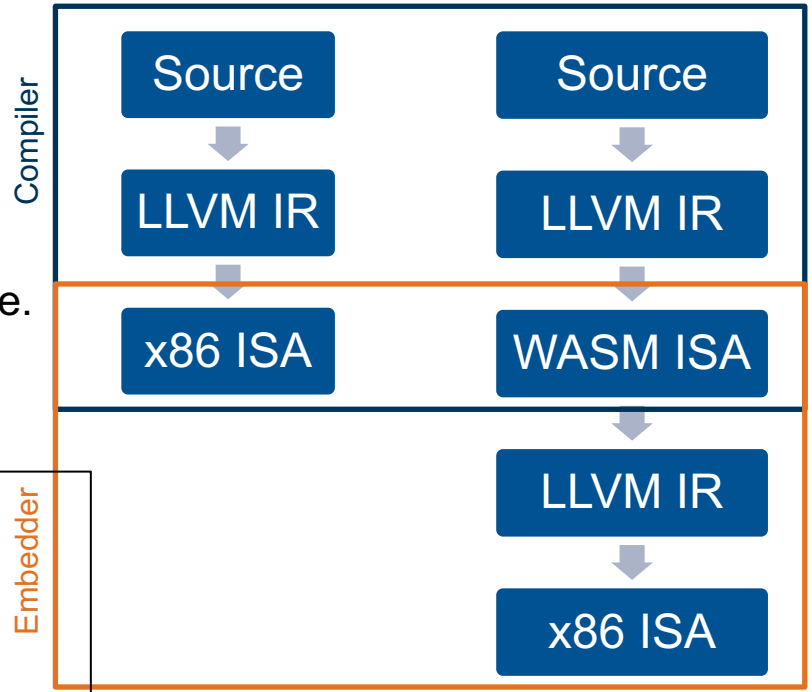
- Extends Wasmer.
- Support for C/C++ applications conforming to MPI-2.2 standard.
- Support for both x86_64 and aarch64 processors.
- High performance execution of MPI-based Wasm modules.
- Low-overhead for MPI calls through zero-copy memory operations.
- Support for high-performance network interconnects.

Executing Wasm Code with High-Performance

- AoT Compilation
- Caching mechanism for generated machine code.

```
$ tree ./cache
./cache
   de5afe5d4f24cf986e1fe1c3e614304a4d860f14d6794e0e99a97ff38887cafe
   e025e7872b4c6b4d852cfc475472f7ab2bc7f67654a31597a0e1076978f939d1
0 directories, 2 files

$ file cache/de5afe5d4f24cf986e1fe1c3e614304a4d860f14d6794e0e99a97...
ELF 64-bit LSB shared object, x86-64, version 1 (SYSV), dynamically linked, not stripped
```



Native (left) vs. WASM (right) code generation flow for LLVM-based compiler and MPIWasm.

Experimental Evaluation

- **128** nodes of HPC system
- One node of AWS Graviton2 processor
- Comparison with standardized HPC benchmarks

CPU Model	AWS Graviton2 (Neoverse-N1)
CPU Cores	32
CPU Frequency	2.50 GHz
CPU Turbo Frequency	3.00 GHz
CPU L1 Cache	2 MB
CPU L2 Cache	32 MB
CPU L3 Cache	32 MB
Memory	64 GB

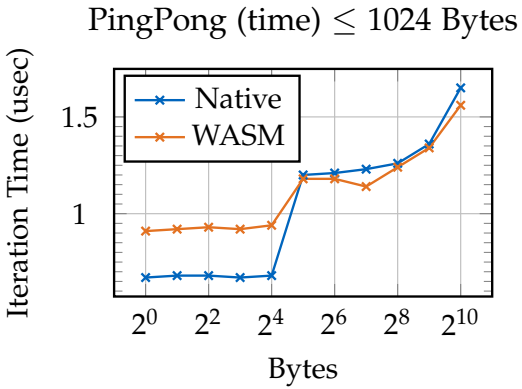
AWS Graviton 2 Specification

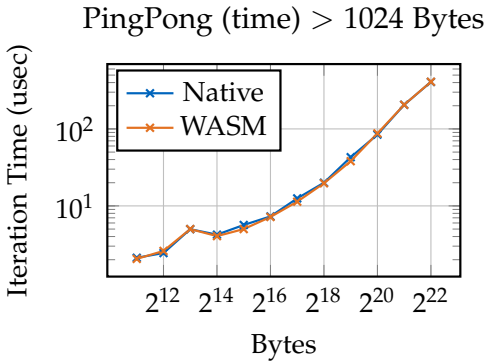
CPU Model	Intel Skylake Xeon Platinum 8174
CPU Cores	48
CPU Base Frequency	2.10 GHz
CPU Turbo Frequency	3.00 GHz
CPU L1 Cache	32 KB
CPU L2 Cache	1 MB
CPU L3 Cache	16.5 MB
Memory	96 GB

SuperMUC-NG Thin Node Specifications

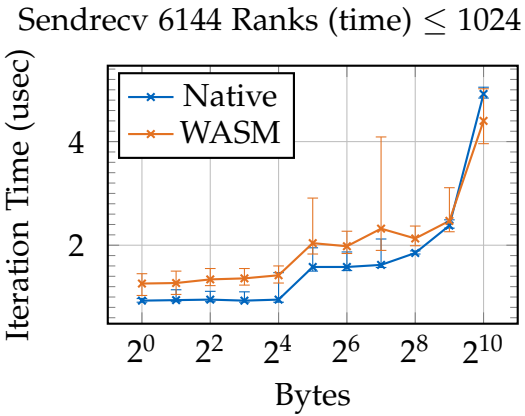
PingPong and SendRecv (x86_64)

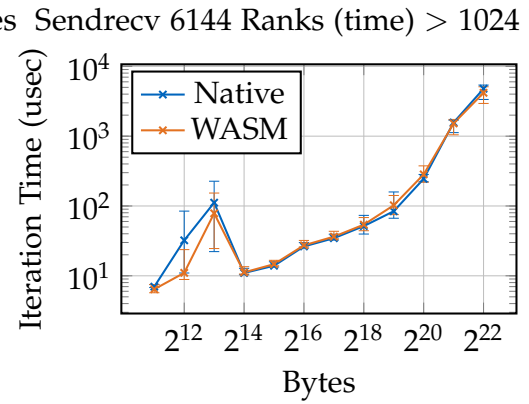
PingPong: 0.05x GM average slowdown





SendRecv: 0.06x GM average slowdown





More Details

Exploring the Use of WebAssembly in HPC

Mohak Chadha, Nils Krueger, Jophin John, Anshul Jindal, Michael Gerndt
Chair of Computer Architecture and Parallel Systems, Technische Universität München, Germany

Shajulin Benedict
Department of Computer Science and Engg., Indian Institute of Information Technology Kottayam, Kerala

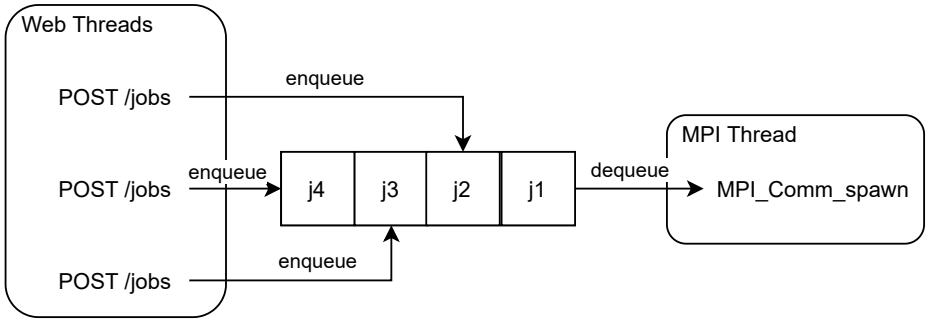
Abstract

Containerization approaches based on *namespaces* offered by the Linux kernel have seen an increasing popularity in the HPC community both as a means to isolate applications and as a format to package and distribute them. However, their adoption and usage in HPC systems faces several challenges. These include difficulties in unprivileged running and build-

ACM Reference Format:
Mohak Chadha, Nils Krueger, Jophin John, Anshul Jindal, Michael Gerndt and Shajulin Benedict. 2023. Exploring the Use of WebAssembly in HPC. In *The 28th ACM SIGPLAN Annual Symposium on Principles and Practice of Parallel Programming (PPoPP '23), February 25-March 1, 2023, Montreal, QC, Canada.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3572848.3577436

Scan Me

Serverless MPI with Wasm




```
{
  "uuid": "d4c6570e-991b-467c-a601-789241ec8d1a",
  "path": "imb.wasm",
  "argv": [],
  "world_size": 8,
  "state": "Submitted",
  "callback": http://localhost:8080/jobs/d4c6...8d1a/callback
}
```


JSON API Response for MPI Job Status

WASI(X)

- WASI on **Steroids**

Essential Features:

- Support for networking
- Support for efficient multi-threading
- Support for process forking

More Information

Questions?

Key Takeaways:

- ❑ Wasm and HPC is an exciting research direction.
- ❑ MPIWasm delivers competive native application performance.
- ❑ Support fo x86_64 and aarch64 architectures.
- ❑ Support for applications written with the MPI-2.2 standard.
- ❑ Support for OpenMPI and MVAPICH.

Thank you for your attention!

Find Us:

MPIWasm: