



Utkarsh Pathak

Performance Comparison of HPC Container Runtimes

A Comprehensive Analysis of Apptainer, Charliecloud, and Podman

Table of contents

- 1 Quick Recap
- 2 Methodology and Experimental Setup
- 3 OSU Micro-Benchmark Results
- 4 IOR and MDTest Results
- 5 Analysis and Future Work
- 6 Conclusion

Research Objectives

- Evaluate Container runtime overhead in HPC environments
- Compare:
 - ▶ Podman
 - ▶ Apptainer
 - ▶ CharlieCloud
- Key metrics
 - ▶ Performance
 - ▶ Resource Utilization
 - ▶ Ease of use

Motivation: Why Containers in HPC?

■ Reproducibility and Portability

- ▶ Same environment across different HPC systems
- ▶ Results can be validated and reproduced

■ Dependency Management

- ▶ Complex software stacks bundled together
- ▶ Eliminate library conflicts and installation issues

■ Isolated Environments

- ▶ Multiple users with different requirements
- ▶ Security and flexibility without admin overhead

The Challenge We're Addressing:

- Which container runtime performs best for HPC workloads?
- What is the actual performance overhead?
- Can we achieve portability without sacrificing performance?

Outline

- 1 Quick Recap
- 2 Methodology and Experimental Setup**
 - Methodology Overview
 - Container Setup and Configuration
- 3 OSU Micro-Benchmark Results
- 4 IOR and MDTest Results
- 5 Analysis and Future Work
- 6 Conclusion

Methodology Overview

■ Test Environment Specification

- ▶ Number of Nodes: 3 (1 Controller and 2 Workers)
- ▶ CPU: 4 per Node
- ▶ Memory: 8 GB
- ▶ Storage: 80 GB
- ▶ OS: Ubuntu 22.04.5 LTS
- ▶ SLURM Version: 21.08.5

■ Benchmarks

- ▶ OSU Micro-Benchmarks
- ▶ IOR and MDTest
- ▶ Real-World Applications (in progress)

■ Container Runtime Versions

- ▶ Podman: 3.4.4
- ▶ Apptainer: 1.4.5
- ▶ CharlieCloud: 0.38

Container Technology Comparison

Aspect	Apptainer	Charliecloud	Podman
Version	1.2.5	0.36	4.8.0
Installation Method	From source	From source	DNF/APT repository
Execution Mode	Rootless (default)	Rootless (always)	Rootless
Storage Driver	OverlayFS	Tarball extraction	overlay/vfs
Base Image	Ubuntu 22.04 LTS	Ubuntu 22.04 LTS	Ubuntu 22.04 LTS
MPI Integration	Host MPI bind	Host MPI bind	Host MPI bind



(a) Apptainer

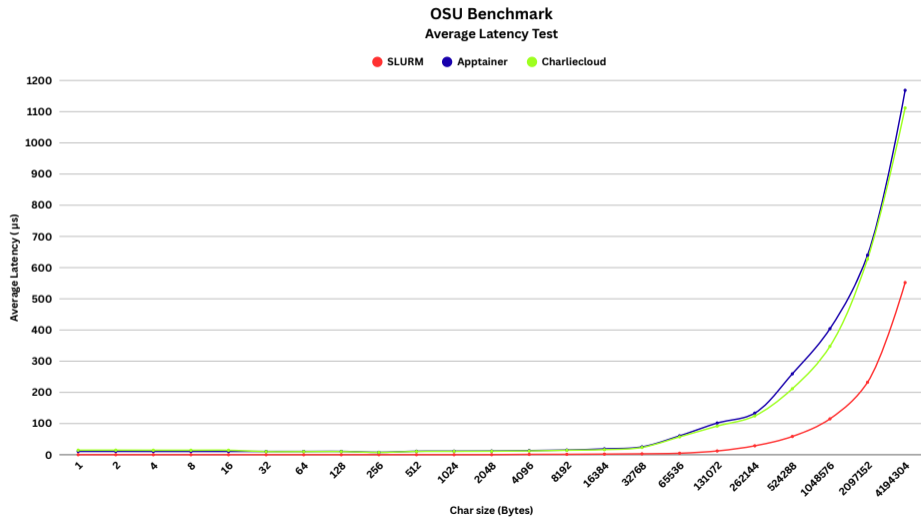


(b) Charliecloud

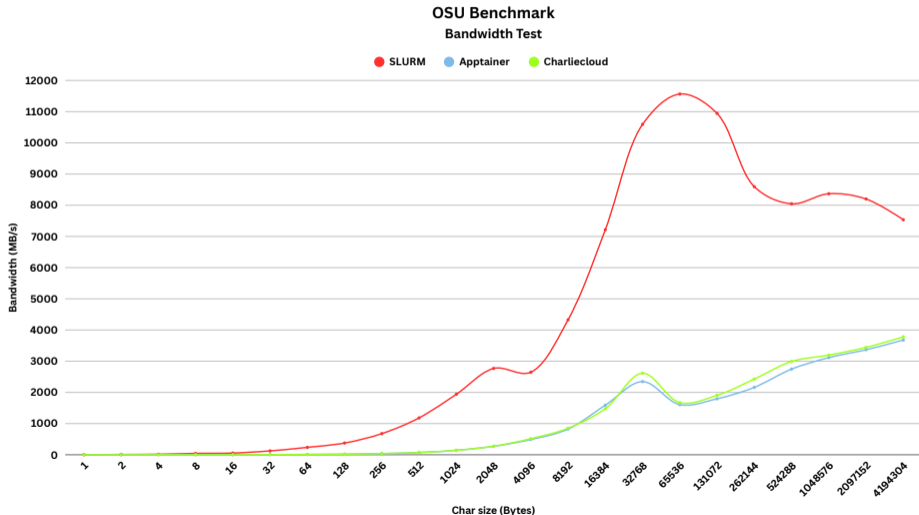


(c) Podman

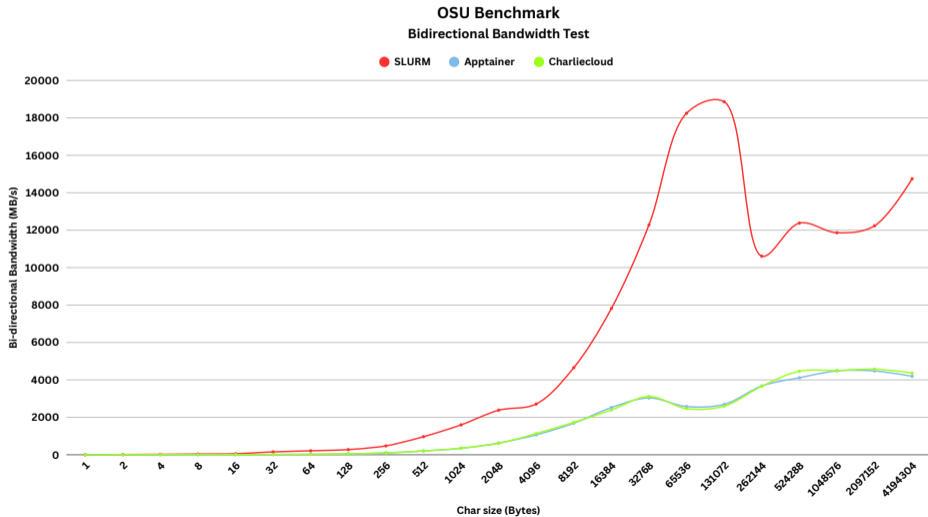
Latency Test



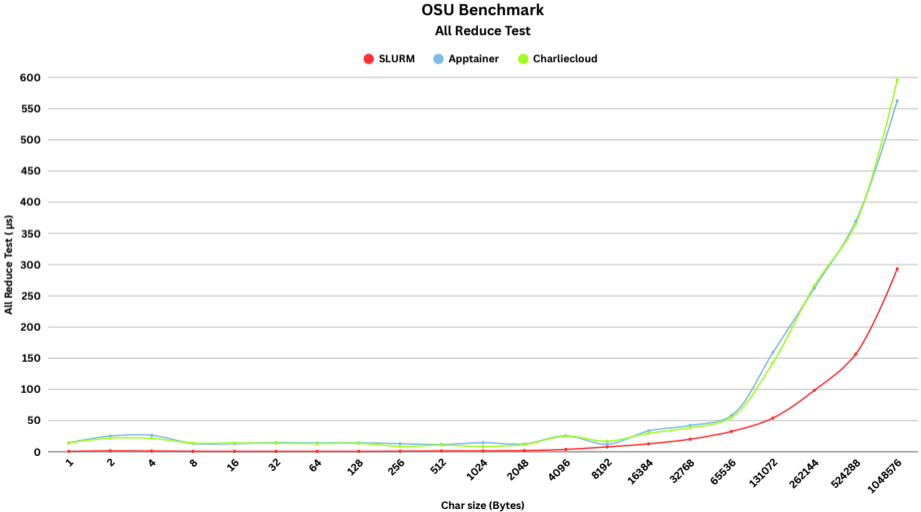
Bandwidth Test



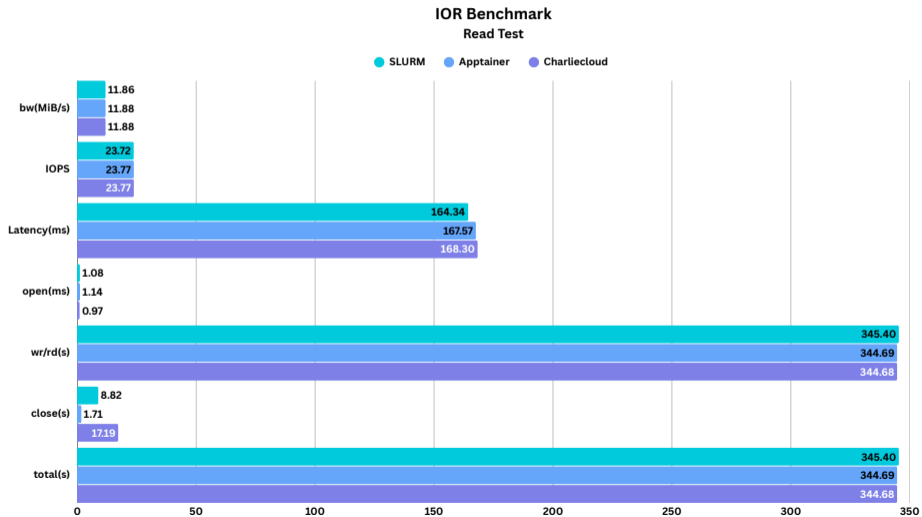
Bidirectional Bandwidth Test



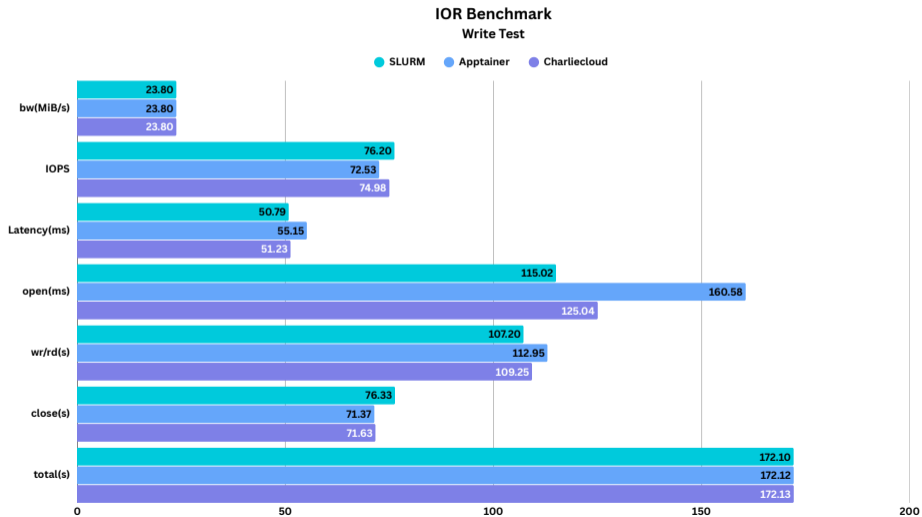
Allreduce Latency Test



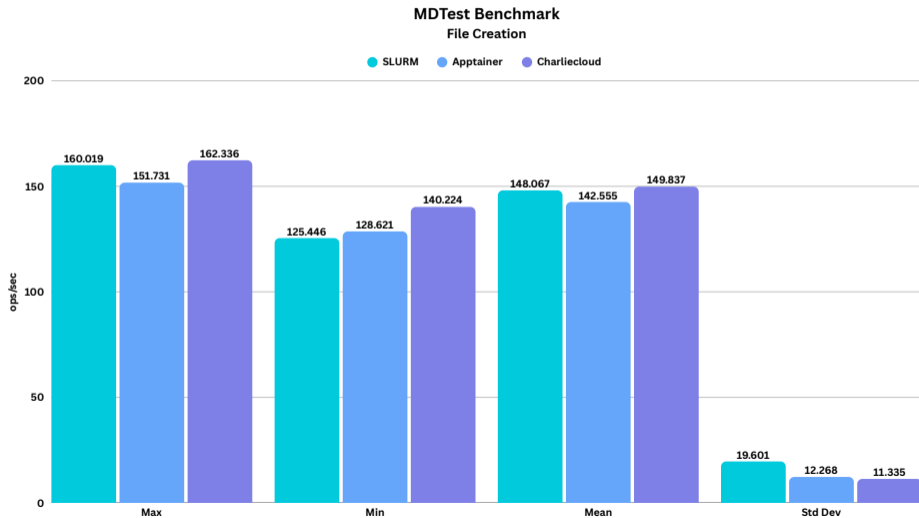
IOR Read Test



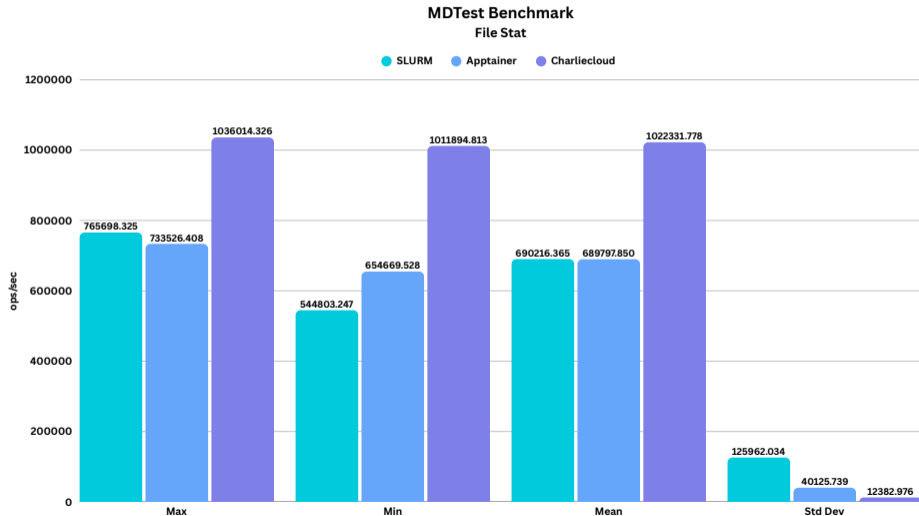
IOR Write Test



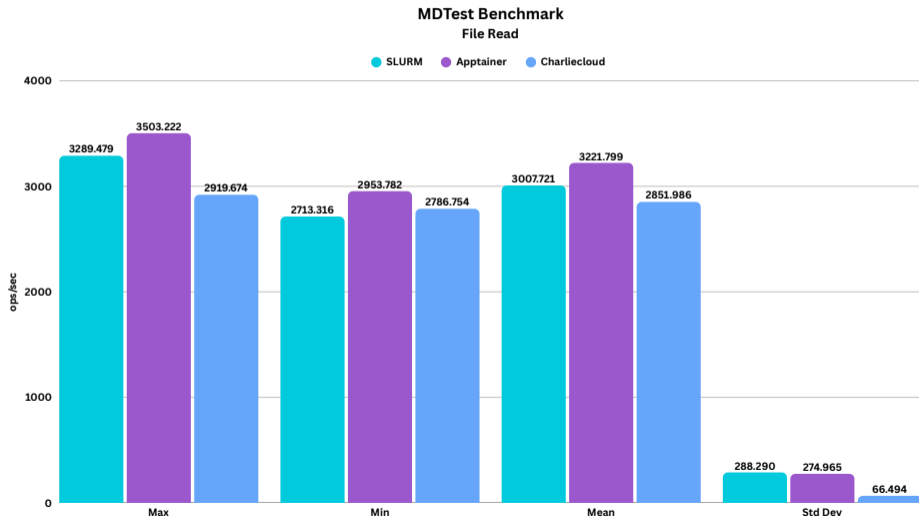
MDTest File Creation



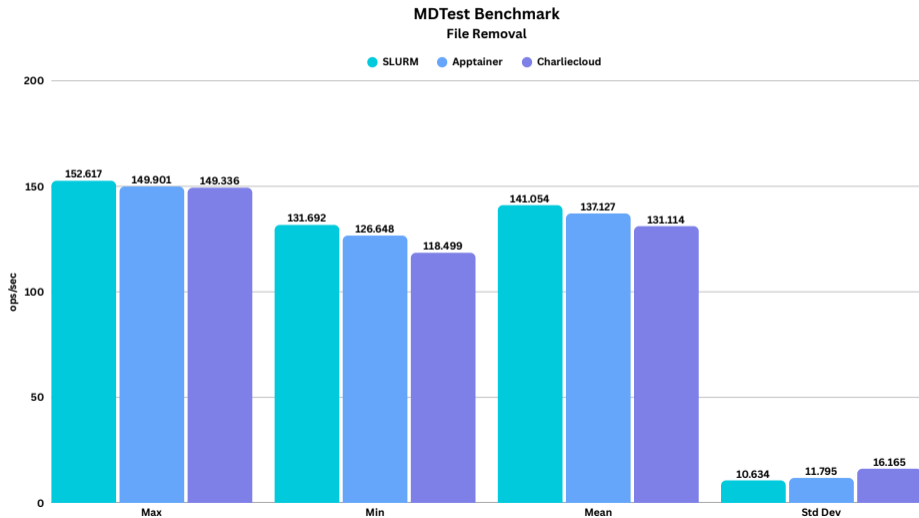
MDTest File Stat



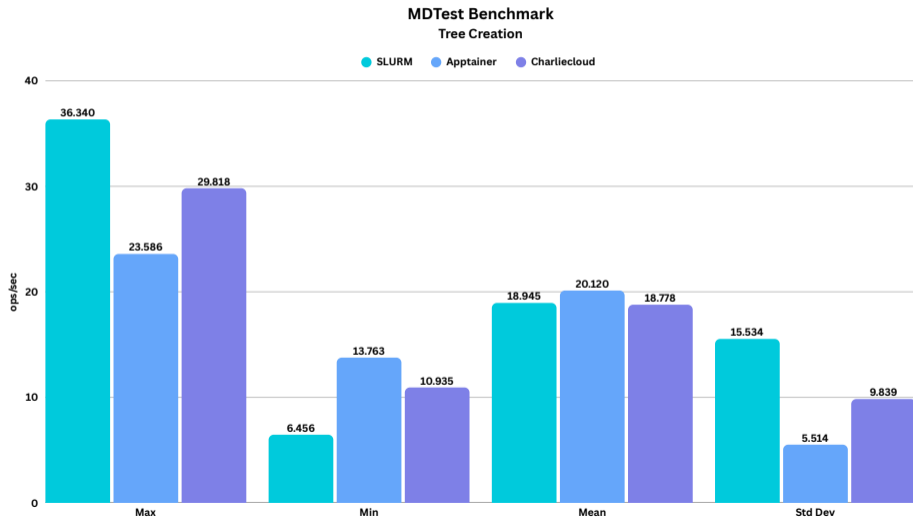
MDTest File Read



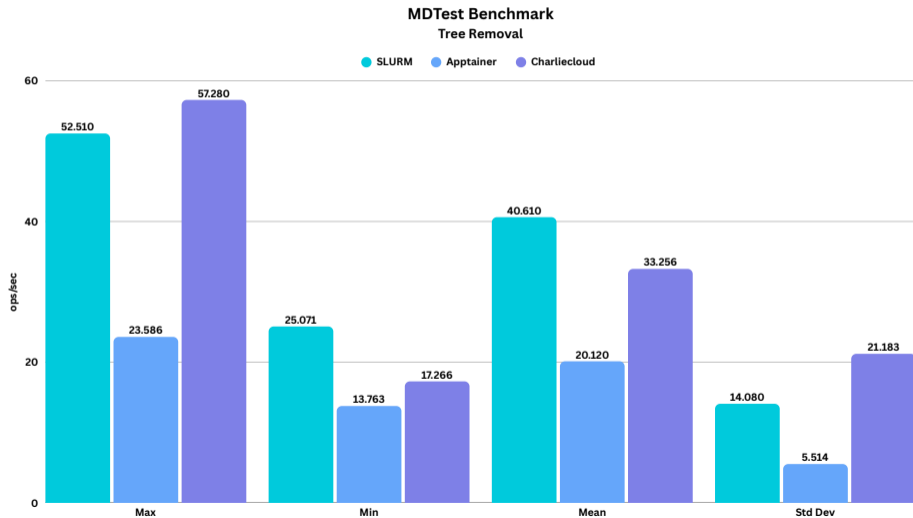
MDTest File Removal



MDTest Tree Creation



MDTest Tree Removal



Preliminary Findings Summary

- The Difference Between Apptainer and Charliecloud is negligible during OSU Benchmarking.
- Charliecloud edges ahead of Apptainer when it comes to File Creation and File Stat tests, while Apptainer performs better in read intensive works.
- Trying to make MPI work with Podman is difficult and inter container communication is difficult.
- Apptainer has a wider community and resources compared to Charliecloud.

Future Work: Extending Benchmark Coverage

- Include System stats like CPU usage, Memory Utilization in comparison
- If time permits, extend the evaluation to selected real-world HPC benchmarks (e.g., LINPACK, GROMACS, NAS, LAMMPS)

Conclusion

- There is a sizeable reduction in MPI communication when using any form of containerization strategy.
- Container runtimes show measurable performance differences in HPC environments.
- No single runtime is optimal for all workloads—use-case matters.
- It is not advisable to use Podman in HPC if other options are available.
- Learning curve for Apptainer and Charliecloud is fairly similar.

Thank you for listening! Questions?

References

<https://apptainer.org/>.

<https://blog.devops.dev/slurm-complete-guide-a-to-z-concepts-setup-and-trouble-shooting-for-admins-8dc5034ed65b>.

<https://charliecloud.io/>.

<https://github.com/forresti/osu-micro-benchmarks>.

<https://podman.io/>.

<https://top500.org/project/linpack/>.

<https://wiki.lustre.org/MDTest>.

<https://www.gromacs.org/>.

<https://www.lammps.org/#gsc.tab=0>.

<https://www.nas.nasa.gov/software/npb.html>.