

Containerized Environment for Reproducibility and Traceability of Scientific Workflows

Student: Paula Olaya* Mentors: Jay Lofstead⁺, Michela Taufer*

*University of Tennessee, Knoxville, ⁺Sandia National Laboratory

Problem Overview

- Scientific domains use **simulations** to understand and predict natural phenomena
- **Trusting the output** of these simulations is **vital** for the scientists
- Trust **requires reproducibility, replicability, transparency and traceability** of the simulation process and its results

Hypothesis: *Annotating the workflow execution at system-level provides a way to ensure reproducibility, replicability, transparency and traceability of simulations*

- When we **annotate** the execution at **system-level**, we are able to **build the record trail of data** moving across the workflow
- This **record trail of data includes** every **workflow component** used to generate the new data (e.g., input datasets, applications, and parameter values)

Our Solution: a Containerized Environment

Leverage cutting edge **container technologies** to address **metadata from the OS level** to build the record trail and ensure the reproduction and traceability of scientific workflows

- Create an **application-agnostic containerized environment**
- Capture **workflow record trails** at runtime

Why container technology?

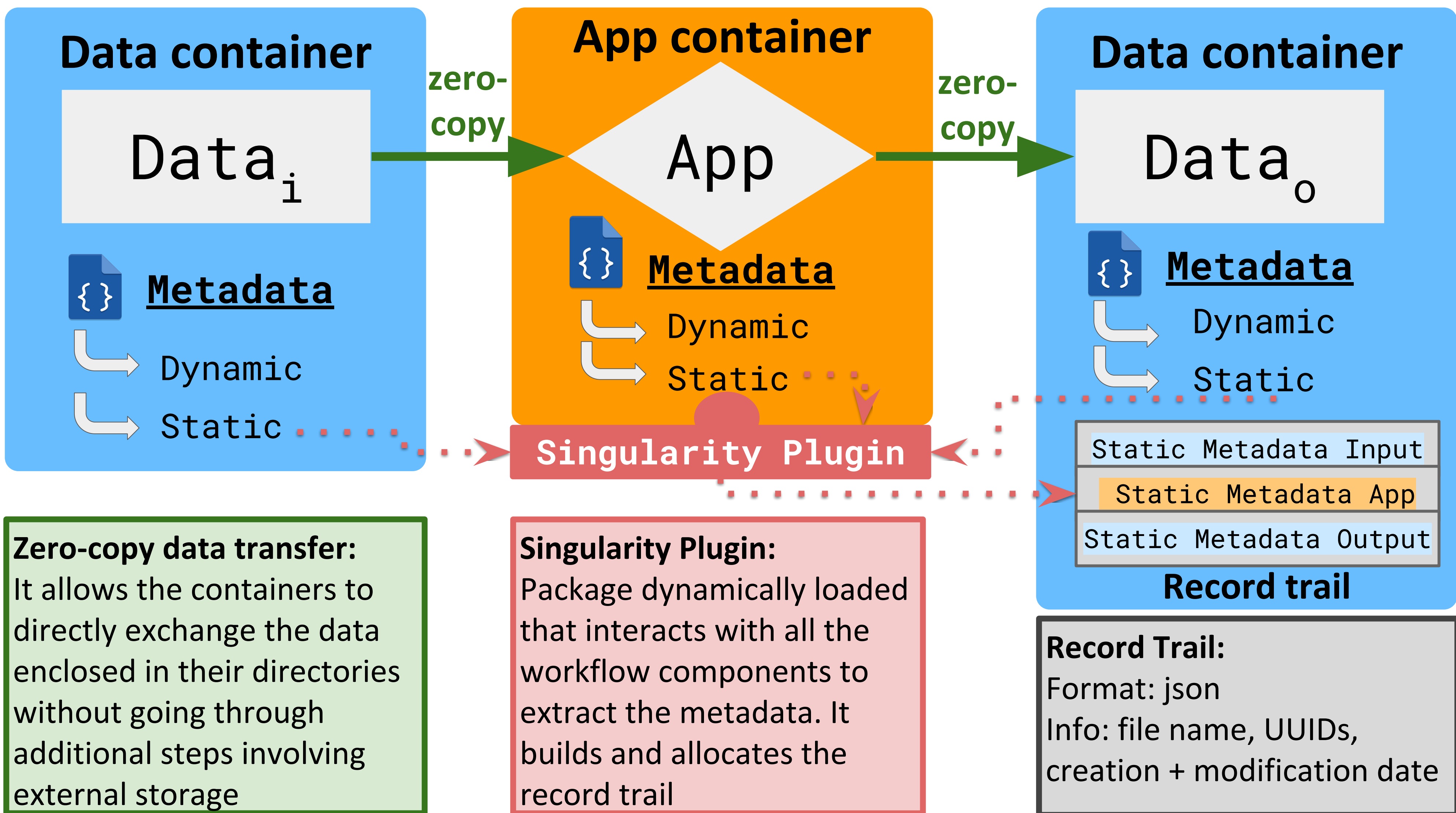
- Portability: Immutable applications machine-agnostic
- Isolation: namespaces
- Encapsulation: sif format
- Unique identification: UUID

Why Singularity?

- Reproducible software stacks
- Mobility of compute
- Compatibility with complex architectures
- Security model

Data Container:
Data compressed and added as a single and independent partition

App Container:
System with specific software stack built from recipe and added as a single and independent partition



Zero-copy data transfer:
It allows the containers to directly exchange the data enclosed in their directories without going through additional steps involving external storage

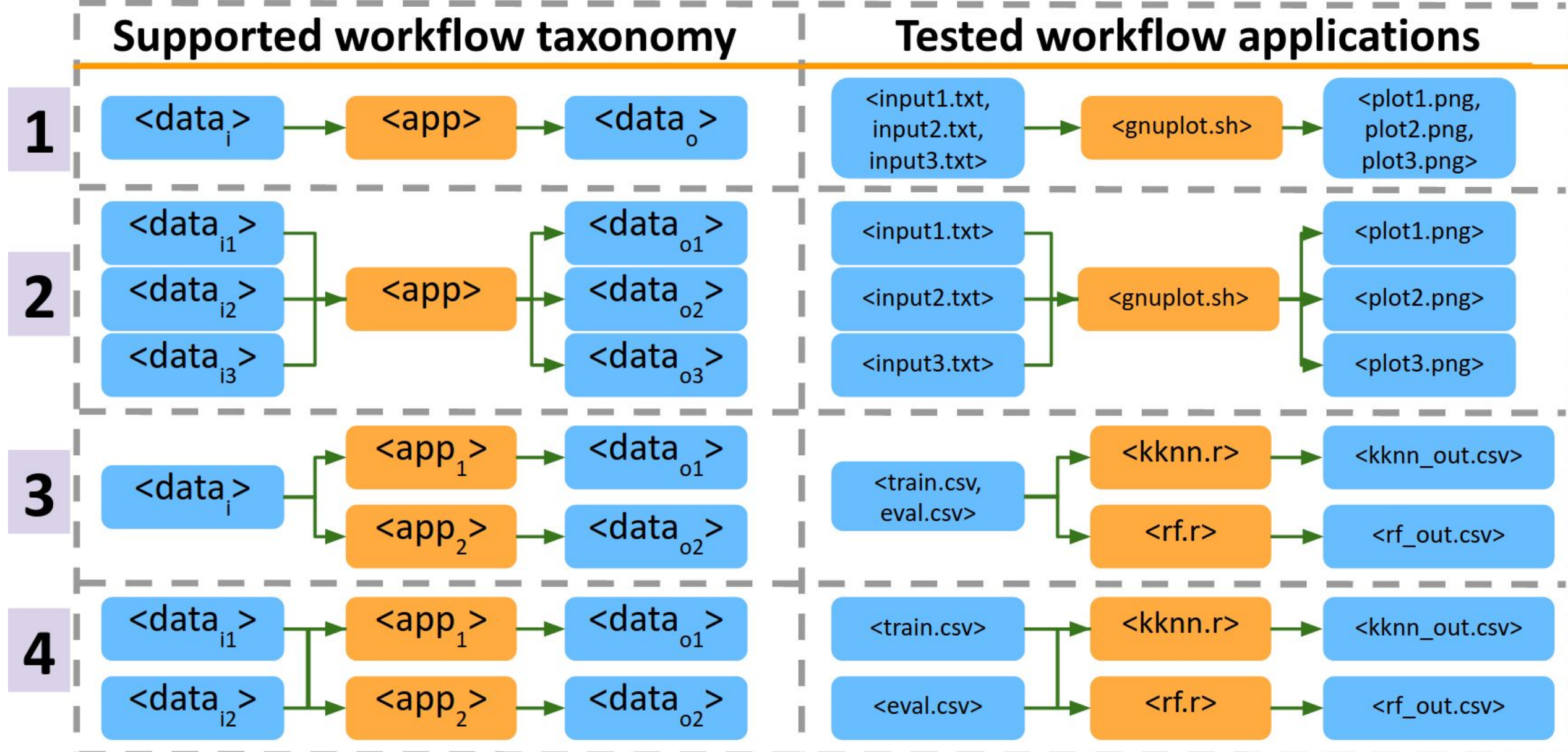
Singularity Plugin:
Package dynamically loaded that interacts with all the workflow components to extract the metadata. It builds and allocates the record trail

Record Trail:
Format: json
Info: file name, UUIDs, creation + modification date

Supported Workflows

We implement our prototype to support four workflows with two base applications:

- Visualization applications
- ML applications



Costs: Time and Memory

Execution Time: Measure average wall-clock over 500 executions for workflow 4

With a more complex and larger application like the RF model the **wall-clock time is tolerable** (0.7% overhead)

Space Overhead: Size of the workflow components (e.g., data and applications)

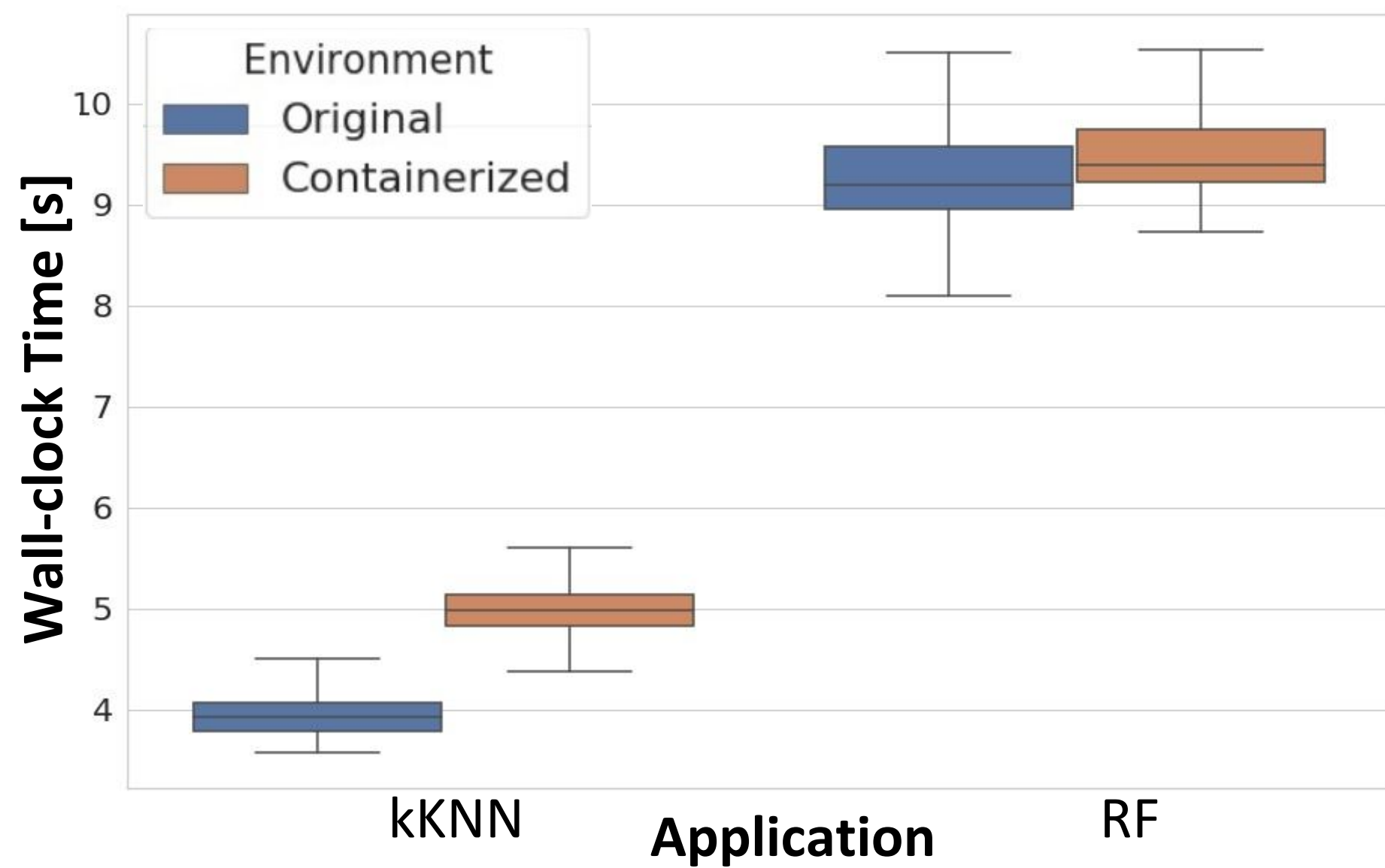
- Analyze overhead introduced by data and application containers size

Data container includes:

- Data files (i.e., Inputs, outputrf.csv, and outputkkn.csv)
- Workflow metadata (i.e. metadata.json)

Original Workflow		Containerized Workflow	
File	Size [KB]	File	Size [KB]
Inputs	12,072	inputs.sif	14,368
outputkkn.csv	2,264	kkn_output.sif	4,132
outputrf.csv	2,264	rf_output.sif	4,132

The **space overhead** for data containers is ~2 MB, caused by the filesystem used



Application container includes:

- Application executable (i.e., gnuplotScript.sh)
- SW package includes system tools, system libraries, and settings (i.e., Ubuntu, gnuplot)

Original Workflow		Containerized Workflow	
File	Size [KB]	File	Size [KB]
gnuplotScript.sh	4	gnuPlotScript.sh	4
gnuplot	139	app.sif/gnuplot	139
Ubuntu 16.04	>40000	Ubuntu 16.04	153,000

The **app container space overhead** is driven by the software stack and OS

Conclusions

Our containerized environment supports:

- No modification of the applications
- Zero-copy transfer of data between containers
- Record trail of different scenarios for workflow metadata
- Metadata attached to dataflow
- Tolerable overhead as workflow complexity increases

Future Work

- Leverage our containerized environment for a broader range of workflows
- Expand containers with automatic set-up, retrieved, and enabled reproducibility of the workflow



For more information, please visit the following link:

Acknowledgements

Thank you to my mentors Dr. Michela Taufer and Dr. Jay Lofstead. Special regards to the Singularity team, specially to Cedric (@cclerget) and Ian Kaneshiro. In collaboration with Sandia National Laboratory and UTK.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

