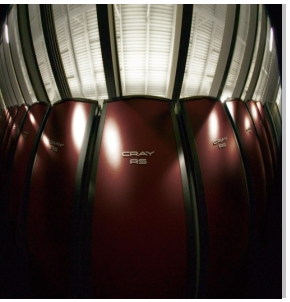
SAND2022-7194C



Using Containers and Automatic Provenance Collection for Sustainable Scientific Software

Jay Lofstead

ASC Sustainable Scientific Software Conference

25 May 2022

Sandia National Laboratories

Exceptional service in the national interest

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

SAND # HERE

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and 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.

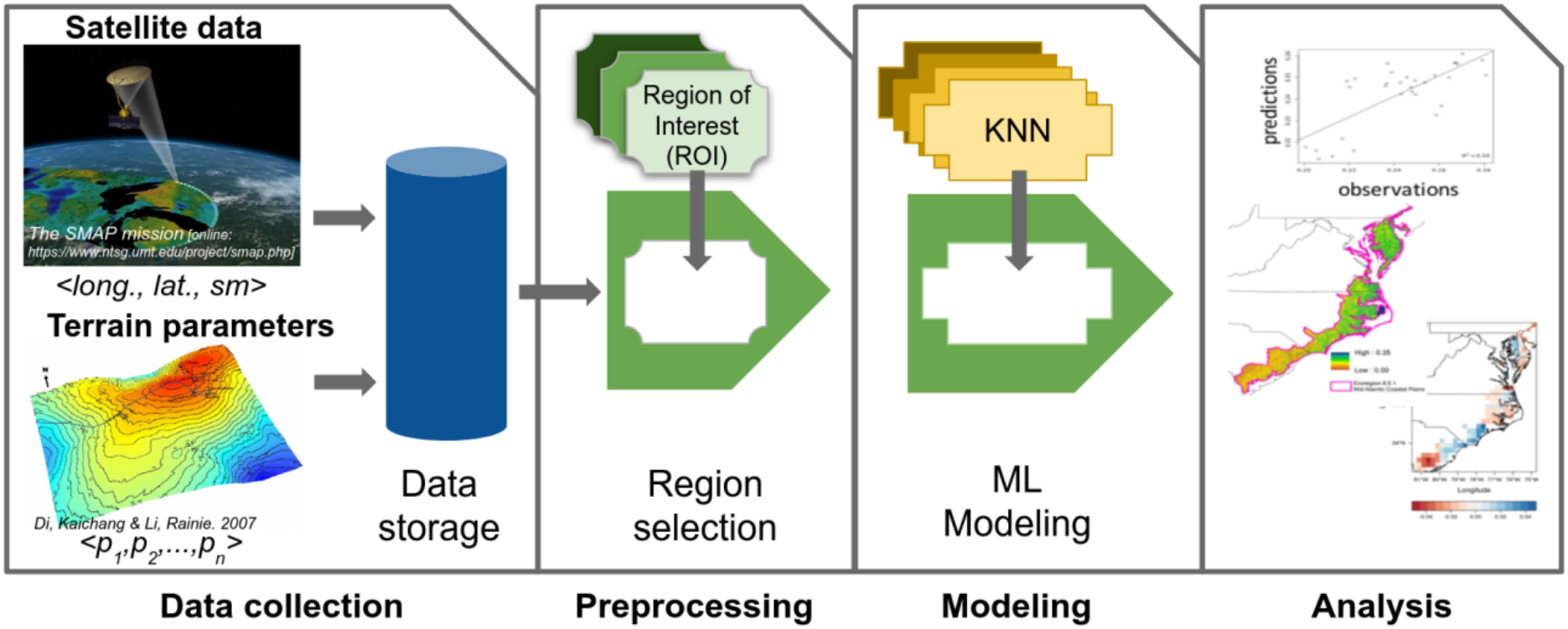
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Project Context

Sandia National Laboratories

- Paula Olaya ; Leobardo Valera ; Ricardo Llamas ; Rodrigo Vargas ; Jay Lofstead; Michela Taufer, “Building Trust in Earth Science Findings through Data Traceability and Results Explainability”, under review at ACM Transactions on Parallel and Distributed Systems
  - Paula’s MS thesis (on arXiv)
- Builds upon Data Pallets work published in 2019

Process example



- SOMOSPIE workflow uses any of four different ML algorithms to extrapolate satellite measurements to different, finer-grained resolution for soil moisture and visualizes the results

Sustainability Problems

Sandia National Laboratories

- How do we know what algorithm was used to create which output?
- What parameters were used to process the data to generate the output?
- How do we enable linking the processed data back to the original source and all processing elements?
- Given two different outputs, what was different about how each was created?
  - Critical for mission needs: if new output differs from old/expected, is the old correct, the new correct, or both wrong? How do you guarantee you answer that question correctly?

Existing Approaches

Sandia National Laboratories

- Separate provenance databases
  - Not linked to data directly and only for data
- Custom file systems
  - Not portable
- Manual documentation
  - Human error has always been and will always be a factor

Containers EVERYWHERE!

Sandia National Laboratories

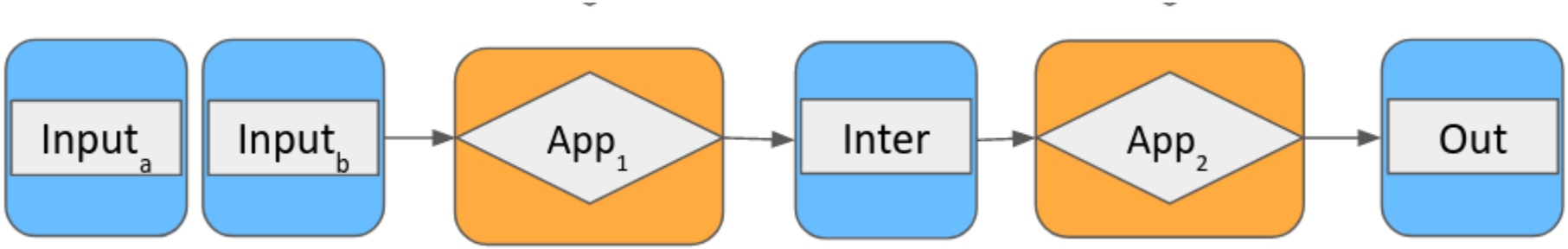
- Basic abstraction is a file system in a file with a unique hash code ID!
- Singularity offers SIF that extends the idea like a physical device enabling separate partitions
- Automatic provenance collection plug-in tags and links invisibly using a separate partition.

- Idea: add detailed, automatically gathered provenance information to a separate, hidden data partition part of every container that shows the history. Add zero-copy capability for data usage/storage (mount multiple containers simultaneously). ALL containers, as explained later, have the metadata.

Fine grained containerization

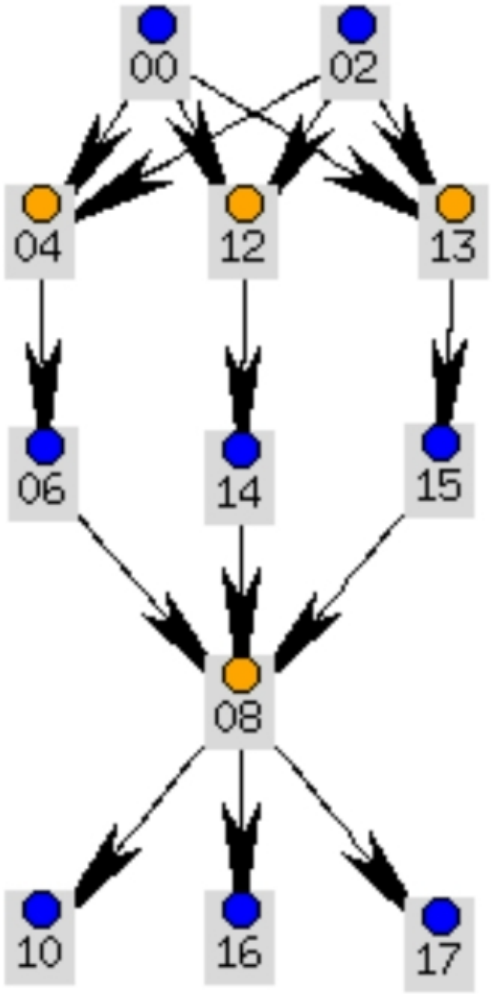
Sandia National Laboratories

- Basic workflow component has input, processing component, and output. Some outputs are used as inputs to another component.



- We choose to track the intermediate data as well by default

Workflow Graph Examples



UUID	00
Container_name	train_27km.sif
Creation_time	2021-09-01T12:07:26-4
Command_line	noop
Record_trail	[00, train_27km.sif]

UUID	02
Container_name	eval_250m.sif
Creation_time	2021-09-09T13:40:02-4
Command_line	noop
Record_trail	[02, eval_250m.sif]

UUID	04
Container_name	**knn.sif**
Creation_time	2021-09-07T12:14:54-EDT
Command_line	noop
Record_trail	NULL

UUID	12
Container_name	**rf.sif**
Creation_time	2021-09-09T21:08:55-EDT
Command_line	noop
Record_trail	NULL

UUID	13
Container_name	**sbm.sif**
Creation_time	2021-09-07T10:34:56-EDT
Command_line	noop
Record_trail	NULL

UUID	06
Container_name	predictions_oklahoma.sif
Creation_time	2021-09-09T13:40:06-4
Command_line	python3 knn.py Train/train.csv Eval/eval_250m.csv Predictions/predictions_oklahoma.csv
Record_trail	[06, predictions_oklahoma.sif] **[04, knn.sif]** [02, eval_250m.sif] [00, train_27km.sif]

UUID	14
Container_name	predictions_oklahoma.sif
Creation_time	2021-09-09T22:37:47-4
Command_line	python3 rf.py Train/train.csv Eval/eval_250m.csv Predictions/predictions_oklahoma.csv
Record_trail	[14, predictions_oklahoma.sif] **[12, rf.sif]** [02, eval_250m.sif] [00, train_27km.sif]

UUID	15
Container_name	predictions_oklahoma.sif
Creation_time	2021-09-08T08:22:50-4
Command_line	python3 sbm.py Train/train.csv Eval/eval_250m.csv Predictions/predictions_oklahoma.csv
Record_trail	[15, predictions_oklahoma.sif] **[13, sbm.sif]** [02, eval_250m.sif] [00, train_27km.sif]

UUID	08
Container_name	visualization.sif
Creation_time	2021-09-11T04:29:08-EDT
Command_line	noop
Record_trail	NULL

UUID	10
Container_name	output_oklahoma.sif
Creation_time	2021-09-11T04:34:17-4
Command_line	python3 visualization.py Predictions/predictions_oklahoma.csv Output/out_oklahoma.png 0.175 0.35
Record_trail	[10, output_oklahoma.sif] [08, visualization.sif] **[06, predictions_oklahoma.sif]**

UUID	16
Container_name	output_oklahoma.sif
Creation_time	2021-09-11T06:08:37-4
Command_line	python3 visualization.py Predictions/predictions_oklahoma.csv Output/out_oklahoma.png 0.175 0.35
Record_trail	[16, output_oklahoma.sif] [08, visualization.sif] **[14, predictions_oklahoma.sif]**

UUID	17
Container_name	output_oklahoma.sif
Creation_time	2021-09-11T06:08:37-4
Command_line	python3 visualization.py Predictions/predictions_oklahoma.csv Output/out_oklahoma.png 0.175 0.35
Record_trail	[17, output_oklahoma.sif] [08, visualization.sif] **[15, predictions_oklahoma.sif]**

Conclusion

Sandia National Laboratories

- Moving to a fine-grained FULLY containerized approach offers longevity for data and applications.
- Yes, there are overheads for storage, but for computation, it is nearly zero.
- Yes, we really need to have ALL system tools in containers and the OS in a VM and the hardware provided with emulators from the vendors, but all of these are solvable.