Augmenting Singularity to Generate Fine-grained Workflows, Record Trails, and Data Provenance

Dominic Kennedy*, Paula Olaya*, Jay Lofstead†, Rodrigo Vargas‡, Michela Taufer*

* University of Tennessee, Knoxville, TN, USA
† Sandia National Laboratories, Albuquerque, NM, USA
‡ University of Delaware, Newark, DE, USA

2022 IEEE 18th International Conference on e-Science (e-Science) | 978-1-6654-6124-5/22/$31.00 ©2022 IEEE | DOI: 10.1109/ESCIENCE55777.2022.00059

***Abstract*—The use of containerization technology in high performance computing (HPC) workflows has substantially increased recently because it makes workflows much easier to develop and deploy. Although many HPC workflows include multiple data and multiple applications, they have traditionally all been bundled together into one monolithic container. This hinders the ability to trace the thread of execution, thus preventing scientists from establishing data provenance, or having workflow reproducibility. To provide a solution to this problem we extend the functionality of a popular HPC container runtime, Singularity. We implement both the ability to compose fine-grained containerized workflows and execute these workflows within the Singularity runtime with automatic metadata collection. Specifically, the new functionality collects a record trail of execution and creates data provenance. The use of our augmented Singularity is demonstrated with an earth science workflow, SOMOSPIE. The workflow is composed via our augmented Singularity which creates fine-grained containers and collects the metadata to trace, explain, and reproduce the prediction of soil moisture at a fine resolution.**

***Index Terms*—Scientific workflows, containers, reproducibility**

I. Motivation and Contributions

Container technologies are software-agnostic runtimes that enable easy applications development and execution by bundling entire applications and their software stack in a single execution environment. In doing so, containers enable portability of applications across platforms. Traditionally, when container technologies are used in HPC or cloud platforms, an entire workflow is deployed in a single container, no matter whether the workflow is composed of one or more interoperable applications, each with its own software stack as well as its own input data and output data. Deploying an entire workflow into a single monolithic container has historically been simpler, and therefore more prevalent. However, this coarse-grained approach makes it difficult to precisely track the thread of execution. Thus, identifying all the workflow components and their interactions for building an in-depth data lineage and record trail poses a real problem to researchers that opt to deploy their entire workflow in one container. In addition, such an approach does not enable workflow reusability and composability of individual workflow components. A better solution would be to instead decouple the workflow into its components (application and data) and use a fine-grained containerization approach that encapsulates each workflow component into its own container. Then, each container can serve as an immutable object with a unique ID for permanent identification, enabling easy data lineage and the creation of record trails.

Many container technologies are available, including Singularity [1], that is de facto the containerization software for HPC applications. None of the existing technologies support the abstraction of a fine-grained workflow. However, Singularity comes with a major advantage compared to other technologies: it offers the possibility to add new functionalities through its plugins. Specifically, in this work, we leverage this feature and augment Singularity to support the concept of fine-grained workflow. Furthermore, we augment Singularity to support the automatic annotation of workflow components with metadata describing the execution record trail and the data provenance. Our work enables workflow reproducibility.

II. Contributions: our Augmented Singularity

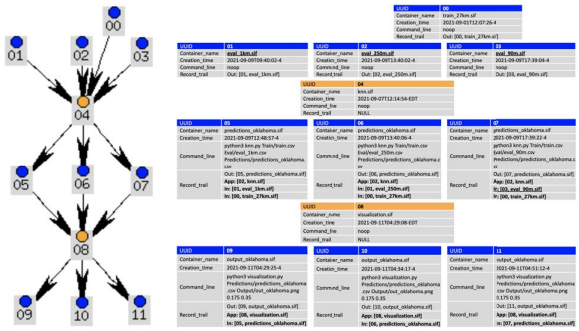

Fig. 1. Metadata partition for each containerized component of the SOMOSPIE workflow for Oklahoma on three resolutions (i.e., 1 km, 250 m, and 90 m) using a ML method (KNN) and visualizing these predictions.

We extend the Singularity runtime to support the concept of fine-grained workflow: we transform a traditionally monolithic workflow into a chain of fine-grained containers hosting applications and data separately. To this end, we implement a Singularity plugin that dynamically modifies the Singularity runtime. We add two main operating functions to the plugin: the ability to create a workflow that is fine-grained containerized, and the ability to execute one of these workflows while also annotating containers with execution metadata. Our

978-1-6654-6124-5/22/$31.00 ©2022 IEEE
DOI 10.1109/eScience55777.2022.00059

plugin provides users with both data provenance and execution record trails, thus supporting transparency, explainability, and reproducibility.

Our plugin extends the Singularity runtime via the Singularity plugin interface; it is written in Go and is compiled with the Singularity source tree. The augmented Singularity includes the additional command line argument `workflow`. We register the new command line argument `workflow` within the plugin and inject it at runtime. The argument allows users to either create a new fine-grained workflow (through the `--create` flag) or execute an existing fine-grained workflow (though the `--run` flag). The command `singularity workflow --create` spins up a web service on the local machine at port `5000`. The web service facilitates the user with the generation of the fine-grained workflow. It does so by asking the user to provide the definition file of an application (i.e., application executable and software stack); the number, location, and size of each input data; and the expected size of the output data. The plugin uses this information to construct the workflow with its individual application and data containers. For the application containers, the plugin encapsulates the application executables and the software stack in a squashFS partition. For the data containers, the plugin encapsulates and compresses the data in an Ext3 file system partition. Both types of containers include metadata in a JSON generic file system partition. The command `singularity workflow --run [workflow_description].json` grants the user the ability to execute a fine-grained workflow. It takes in a workflow description, previously created by the workflow creation command, and proceeds to execute the workflow described in the file. The plugin does this by first locating all of the containers, then executes the application container while mapping in the filesystems of the input data containers and output data containers. The execution may then continue as usual with all of the appropriate file systems linked. After execution has completed, data containers are dynamically annotated with a metadata partition that reflects the data provenance as well as any application that is used to produce the results.

III. Application to a Real Workflow

We demonstrate the use of our augmented Singularity to build a fine-grained workflow and collect its provenance using SOMOSPIE [2], an earth science workflow. This workflow uses a suite of machine learning modeling techniques to downscale the 27 km resolution satellite data from the ESA-CCI soil moisture database to finer-grain resolutions necessary for practical use in earth sciences including precision forestry and agriculture, hydrology for landscape ecology, and regeneration dynamics. The workflow consists of two applications: (i) machine learning methods [i.e., k-nearest neighbors (KNN), random forest (RF), and surrogate based modeling (SBM)] that transform satellite data into finer-resolution, gap-free predictions; and (ii) visualization methods that transform predictions into images and other graphical representations.

We use our augmented Singularity function to generate the fine-grained workflow for predicting soil moisture at different resolutions (1 km, 250 m, and 90 m) for Oklahoma (a rich agricultural area). We use four input data: one for the 27 km satellite data (training data) and three for the finer-resolutions 1 km, 250 m, and 90 m (evaluation data). We create the definition files with the software stack (i.e., libraries, packages) and executables for the three ML methods (KNN, RF, and SBM) and the visualization stage. Finally, we provide the expected size of the output data. After the creation of the fine-grained containerized SOMOSPIE workflow, we use our `singularity workflow --run somospie.json` operating function to execute the workflow and automatically capture the execution record trail and the data provenance. Figure 1 presents the metadata partitions generated by our Singularity extension for the execution of one ML method (KNN) on the three finer-resolutions input data containers followed by the visualization of the generated soil moisture predictions. With the metadata information, the scientists can understand the provenance of the data sources, as well as the applications that were used to produce the intermediary and final results. Furthermore, having the containerized components with their identifications makes it easier for the scientists to retrieve those containers and use them to reproduce the results or generate new studies.

IV. Broader Impacts

Our work facilitates domain scientists to generate and execute fine-grained containerized workflows while automatically and seamlessly providing provenance information about the execution environment. The fine-grained containerization of the workflow components (i.e., applications and data) enables easy retrieval for reusability and reproducibility, as well as portability across a broad spectrum of platforms. The provenance information enables traceability, explainability, and trustworthiness in the scientific discovery. With the data lineage and execution record trails, automatically generated by our extensions, domain scientists are able to 1) trace and understand the data transformations along the execution and 2) explain the different results by linking them to the specific methodology. All this combined allows users to earn trust in the data, software, and environment wrapped in the scientific discovery loop.

Code Availability

The code implementing the augmented Singularity can be found at: https://github.com/TauferLab/ContainerizedEnv.

Acknowledgements

The authors acknowledge the support of Sandia National Laboratories; the National Science Foundation through the awards #1841758, #1941443, #2103845, #2028923, #2138811, and #2103836. The authors want to thank Cedric Clerget and Ian Kaneshiro, for their support with Singularity.

References

[1] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLOS ONE*, vol. 12, 05 2017.

[2] D. Rorabaugh, M. Guevara, R. Llamas, J. Kitson, R. Vargas, and M. Taufer, "SOMOSPIE: A Modular SOil MOisture SPatial Inference Engine Based on Data-Driven Decisions," in *Proc. of the 2019 15th International Conference on eScience*, 2019, pp. 1–10.