The Journal of Open Source Software

Science Capsule - Capturing the Data Life Cycle

Devarshi Ghoshal[1], **Ludovico Bianchi**[1], **Abdelilah Essiari**[1], **Michael Beach**[1, 2], **Drew Paine**[1], **and Lavanya Ramakrishnan**[1]

1 Lawrence Berkeley National Lab **2** University of Washington

DOI: 10.21105/joss.02484

Software

- Review
- Repository
- Archive

Editor: Amy Roberts

Reviewers:

- @colbrydi
- @gflofst
- @atrisovic

Submitted: 28 June 2020
Published: 17 June 2021

Summary

The data generated from scientific workflows often become unusable due to the lack or incompleteness of information required for processing and analyzing the data. Reproducibility of scientific data and workflows facilitates efficient processing and analyses. A key to enabling reproducibility is to capture the end-to-end workflow life cycle, and any contextual metadata and provenance. Existing tools (Guo & Seltzer (2012), Chirigati et al. (2016), Brinckman et al. (2019), Šimko et al. (2019)) require researchers to either modify or instrument their analyses, which is a barrier to use.

Science Capsule is a free open source software that allows researchers to automatically capture their end-to-end workflows including the scripts, data, and execution environment. Science Capsule monitors the workflow environment to capture the provenance at runtime. It provides a timeline view and a web interface to represent the workflow and data life cycle, and the associated provenance information. Science Capsule also leverages container technologies to provide a lightweight, executable package of the scripts and required dependencies, ensuring portability and reproducibility.

The Science Capsule software has the following key features:

- Captures provenance through file system and process monitoring
- Uses container technologies to create re-executable packages containing scripts and metadata
- Runs on a user's desktop and is also compatible with container technologies such as Shifter (Gerhardt et al. (2017)) running on HPC environments
- Provides a timeline view of an execution through a web interface
- Provides an interface to add artifacts including lab notebooks and notes
- Supports multiple OS platforms including Windows, Mac and Linux

Statement of Need

Science Capsule addresses the need for reproducible science. Users can download the Science Capsule software and easily set it up to capture their workflow and provenance at runtime. Unlike many existing tools, Science Capsule does not require users to modify or instrument their code. The timeline view provides a visualization for the scientist to understand their workflows and data life cycle. Science Capsule lets scientists monitor, share, reproduce or use their workflows across different software and hardware platforms.

Science Capsule is currently being used by researchers at the Lawrence Berkeley National Laboratory. Generally, scientists with any data processing or analyses workflow will benefit from the Science Capsule software package. For example, a scientist using experimental

facilities such as light sources or electron microscopes will be able to capture their data analyses environment during experiment time and reuse it for post-analyses of the data or share the workflows with other researchers.

Science Capsule Software

Science Capsule is implemented using Python, and uses Node JS and Javascript for the web interface. It also uses a mongodb database for storing and managing events captured by the different monitoring tools. Science Capsule has been evaluated using synthetic and real-world workflows (e.g., Montage) on Windows, MacOS and Linux.

Science Capsule supports two modes for capturing the information about workflows and the associated provenance: a) container mode, where Science Capsule captures metadata for all the processes and artifacts that are encapsulated within a Docker container, and b) bare-metal mode, where Science Capsule captures the execution time provenance for user-specified artifacts. The two modes in Science Capsule provide different levels of reproducibility. In the container mode, all process and data events of a workflow, which run inside the container, are captured. This allows for complete reproducibility of the workflow. When using the bare-metal mode, Science Capsule monitors the file system events of the user-specified directories to understand the data life cycle of a workflow. This is typically useful for the workflows that cannot be managed in a container.

Automatic Monitoring and Capture

Science Capsule monitors the environment where the workflows are managed by the researchers by using different system-level file and process monitoring tools. The use of these tools depend on the underlying platform and the granularity at which researchers intend to capture the information. Currently, Science Capsule captures events from inotify (Fisher (2017)), Linux's strace utility and the Python watchdog library. These tools are configured and set up during the installation of Science Capsule. These raw events captured by the various monitoring tools are processed to extract the high-level data life cycle and task execution information. Finally, both raw and processed events are stored in a mongodb database for sharing and visualizing the execution and provenance of scientific workflows.

Interactive Web Interface

The processed events in Science Capsule are used to represent various activities of a scientific workflow in a chronological order, a.k.a. the timeline. Science Capsule provides an interactive web interface where researchers can view the timeline in near real-time and/or annotate with notes/images that might capture a researcher's thought process and experiment design. The interactive web interface provides a way for the researchers to enrich the execution and provenance information collected through the monitoring framework for enabling better reusability and sharing of knowledge. Figure 1 shows an example timeline for a workflow as captured by Science Capsule.

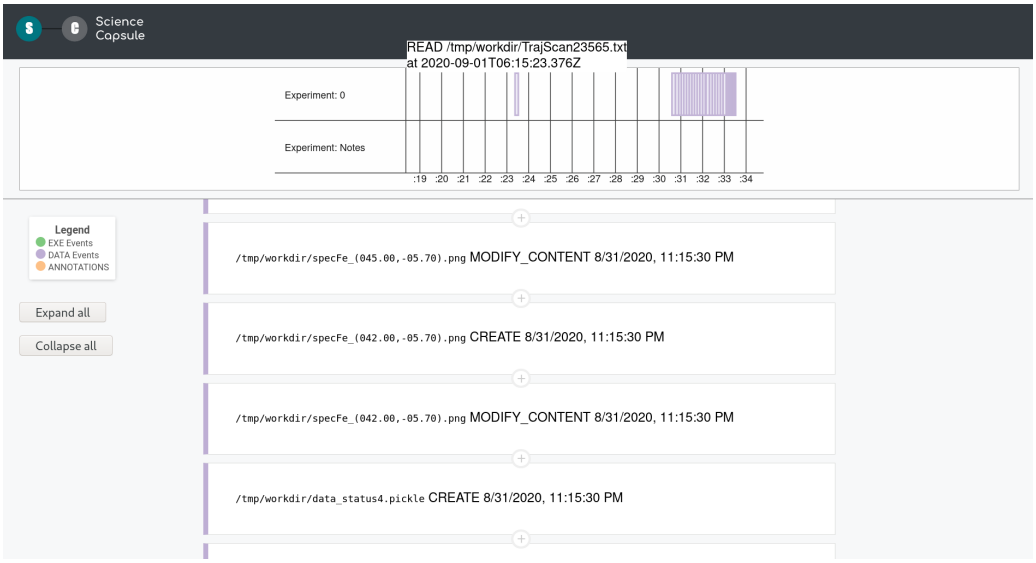

Figure 1: Science Capsule web user interface.

Science Capsule Containers

A Science Capsule container allows users to monitor their workflows running inside the container. The containers themselves are managed by docker commands. However, the Science Capsule software inside the container transparently monitors and captures the metadata necessary to understand and reproduce the workflow. The docker commands are also used to save and export the container with all the workflow data and metadata. One limitation of this approach is that only the current state of the workflow can be saved and exported for sharing. We are working on tracking and managing different stages and versions of a workflow execution, which will allow users to create and share different states, timelines and execution traces of a workflow.

Related Work

Over the past decade, several frameworks have been developed to enable computational reproducibility. Burrito (Guo & Seltzer (2012)) captures a researcher's computational activities and provides user interfaces to annotate the captured provenance. Other tools provide specific wrappers and dedicated interfaces to create reproducible packages for software-based experiments and computational narratives (Chirigati et al. (2016), Brinckman et al. (2019), Ton That et al. (2017)). Reprozip (Chirigati et al. (2016)) is a Linux-only tool designed to help scientists package up their software-based experiment after it is completed. This tool traces systems calls using ptrace to generate provenance information, and identify software packages that can then be reconstituted in a virtual environment or Docker image. Sciunit is another Linux command-line tool (Ton That et al. (2017)) that lets users reproduce their computational experiments, and captures any command a user runs through the UNIX shell. REANA (Šimko et al. (2019)) provides a platform for defining and managing reusable workflows through cloud computing. Unlike Science Capsule, these tools often require modifications to researchers' existing work practices, can only run on specific platforms (Linux), and do not allow users to capture ad-hoc resources (e.g., lab notebooks etc).

Metadata and provenance are critical in building knowledge for enabling reproducibility of scientific workflows. Past research has shown the use of data provenance for sharing and reproducing scientific workflows (Goble et al. (2010), fomel2013madagascar). However, existing workflow management systems (Oinn et al. (2004), Deelman et al. (2015), Altintas

et al. (2006), barga2006automatic) are explicitly instrumented for capturing data provenance from scientific workflows. Science Capsule is agnostic to workflow tools and uses system-level monitoring to extract information from experimental processes and artifacts. It augments and complements existing real world practices and tools without requiring a wholesale adaptation of scientist's work to fit the design of Science Capsule, while providing usable and understandable interfaces for engaging with detailed provenance information.

Additionally, scientists and researchers may not be using existing workflow tools for managing their scientific pipelines, which makes the case for other alternatives for collecting relevant metadata to enable reproducibility. Traditionally, filesystem metadata have been used for performance monitoring and anomaly detection (Miller et al. (2010), Muniswamy-Reddy et al. (2006), Huang & Wong (2011)). Metadata and data context services like Ground (Hellerstein et al. (n.d.)) and Bluesky data broker (Arkilic et al. (2015)) provide integrated interfaces to access data and metadata from various sources. We envision Science Capsule to use metadata collected from these systems to enrich the provenance in addition to what is automatically captured through system-level monitoring tools in Science Capsule .

Acknowledgements

This work is supported by the U.S. Department of Energy, Office of Science and Office of Advanced Scientific Computing Research (ASCR) under Contract No. DE-AC02-05CH11231.

References

Altintas, I., Barney, O., & Jaeger-Frank, E. (2006). Provenance collection support in the kepler scientific workflow system. *International Provenance and Annotation Workshop*, 118–132. https://doi.org/10.1007/11890850_14

Arkilic, A., Allan, D., Chabot, D., Dalesio, L., Lewis, W., & others. (2015). Databroker: An interface for NSLS-II data management system. *15th Int. Conf. On Accelerator and Large Experimental Physics Control Systems (ICALEPCS'15), Melbourne, Australia, 17-23 October 2015*, 645–647.

Brinckman, A., Chard, K., Gaffney, N., Hategan, M., Jones, M. B., Kowalik, K., Kulasekaran, S., Ludäscher, B., Mecum, B. D., Nabrzyski, J., & others. (2019). Computing environments for reproducibility: Capturing the "whole tale." *Future Generation Computer Systems*, *94*, 854–867. https://doi.org/10.1016/j.future.2017.12.029

Chirigati, F., Rampin, R., Shasha, D., & Freire, J. (2016). ReproZip: Computational reproducibility with ease. *Proceedings of the 2016 International Conference on Management of Data*, 2085–2088. https://doi.org/10.1145/2882903.2899401

Deelman, E., Vahi, K., Juve, G., Rynge, M., Callaghan, S., Maechling, P. J., Mayani, R., Chen, W., Da Silva, R. F., Livny, M., & others. (2015). Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, *46*, 17–35. https://doi.org/10.1016/j.future.2014.10.008

Fisher, C. (2017). Linux filesystem events with inotify. *Linux Journal*, *2017*(280), 2.

Gerhardt, L., Bhimji, W., Fasel, M., Porter, J., Mustafa, M., Jacobsen, D., Tsulaia, V., & Canon, S. (2017). Shifter: Containers for hpc. *J. Phys. Conf. Ser.*, *898*, 082021. https://doi.org/10.1088/1742-6596/898/8/082021

Goble, C. A., Bhagat, J., Aleksejevs, S., Cruickshank, D., Michaelides, D., Newman, D., Borkum, M., Bechhofer, S., Roos, M., Li, P., & others. (2010). myExperiment: A

repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Research*, *38*(suppl 2), W677–W682. https://doi.org/10.1093/nar/gkq429

Guo, P. J., & Seltzer, M. I. (2012). *Burrito: Wrapping your lab notebook in computational infrastructure*.

Hellerstein, J. M., Sreekanti, V., Gonzalez, J. E., Dalton, J., Dey, A., Nag, S., Ramachandran, K., Arora, S., Bhattacharyya, A., Das, S., & others. (n.d.). *Ground: A data context service*.

Huang, L., & Wong, K. (2011). Anomaly detection by monitoring filesystem activities. *2011 IEEE 19th International Conference on Program Comprehension*, 221–222. https://doi.org/10.1109/ICPC.2011.23

Miller, R., Hill, J., Dillow, D. A., Gunasekaran, R., Shipman, G. M., & Maxwell, D. (2010). Monitoring tools for large scale systems. *Proceedings of Cray User Group Conference (CUG 2010)*.

Muniswamy-Reddy, K.-K., Holland, D. A., Braun, U., & Seltzer, M. I. (2006). Provenance-aware storage systems. *Usenix Annual Technical Conference, General Track*, 43–56.

Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., Carver, T., Glover, K., Pocock, M. R., Wipat, A., & others. (2004). Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, *20*(17), 3045–3054. https://doi.org/10.1093/bioinformatics/bth361

Šimko, T., Heinrich, L., Hirvonsalo, H., Kousidis, D., & Rodríguez, D. (2019). REANA: A system for reusable research data analyses. *EPJ Web of Conferences*, *214*, 06034. https://doi.org/10.1051/epjconf/201921406034

Ton That, D. H., Fils, G., Yuan, Z., & Malik, T. (2017). *Sciunits: Reusable research objects*. 374–383. https://doi.org/10.1109/eScience.2017.51