Department: View from the Cloud
Editor: Ewa Deelman, deelman@isi.edu

The Silver Lining

Kate Keahey
Consortium for Advanced Science and Engineering (CASE), University of Chicago
Mathematics and Computer Science Division, Argonne National Laboratory

***Abstract*—Clouds are shareable scientific instruments that create the potential for reproducibility by ensuring that all investigators have access to a common execution platform on which computational experiments can be repeated and compared. By virtue of the interface they present, they also lead to the creation of digital artifacts compatible with the cloud, such as images or orchestration templates, that go a long way—and sometimes all the way—to representing an experiment in a digital, repeatable form. In this article, I describe how we developed these natural advantages of clouds in the Chameleon testbed and argue that we should leverage them to create a digital research marketplace that would make repeating experiments as natural and viable part of research as sharing ideas via reading papers is today.**

Infrastructure clouds have transformed our world with thunder and lightning—they democratize computing by making expensive resources available to all, eliminate or reduce much of the complexity associated with their operation, and tap into economies of scale making science more cost-effective. And now they are also quietly helping us transform science by fostering ways of sharing academic knowledge through enabling packaging and publishing of a digital unit of scientific research: an experiment.

Digital Object Identifier 10.1109/MIC.2020.3013361
Date of current version 9 September 2020.

We all aspire to a world where experiments presented in publications can be easily accessed, instantiated, and played with: can I run the presented algorithm with my dataset? check out how the system might behave on different hardware? try it with my noise generator? teach by having students examine examples of the latest published research hands-on? The availability of all the data needed to reproduce an experiment is of course a necessary condition—the sufficient condition is time. That time is measured in how easy to repeat an experiment is—and it is becoming increasingly clear that clouds can help in fundamental ways.

The observations described here have been developed in the context of operating the Chameleon testbed[1] for the last 5 years. Chameleon is a

Published by the IEEE Computer Society

bare metal reconfigurable research cloud—a scientific instrument for Computer Science research that so far has served over 4500 users working on more than 600 research and education projects. It is interesting to consider, because it expresses the systems research testbed use case in terms of a mainstream cloud. Unlike traditional research testbeds, which have overwhelmingly been configured by using custom solutions developed in-house, our team implemented the system by using the OpenStack open source cloud implementation[2]—albeit in adventurous configuration that supports bare metal reconfiguration, network stitching, and other types of system experimentation. This means that our work at any point is driven by two joint considerations: how to better support science—and furthermore how to do it in terms of cloud configuration that is accessible to all.

SHARABLE SCIENTIFIC INSTRUMENTS AND RESEARCH SHARING

The original purpose of research clouds or testbeds—such as Chameleon[1], FABRIC[3], or the wireless PAWR testbeds[4]—was to provide a platform for research, in other words, a scientific instrument representing scale, diversity, and overall cost that was beyond the reach of individual scientist, department, or institute. This led to the development of a class of open, *shareable instruments* whose goal is to not only provide functions essential to the type of research they support—but also develop effective sharing mechanisms for that type of research so that a large investment can be amortized across many users and experiments. An important side-effect of sharable instruments is that by supporting sharing they become *instruments held in common* and thus create an essential reproducibility platform. Thus, for example, it is no longer the case that if I publish results of work using the newest type of accelerator I happen to have access to, others cannot verify or extend them because this same type of accelerator is not available to them: an instrument held in common ensures that all investigators have access to the same type of hardware—or even the same exact hardware resource—which means that a baseline criterion for repeating the same experiment has been met.

There is another interesting though more subtle side-effect of sharable instruments that advances the potential for repeatability even further.

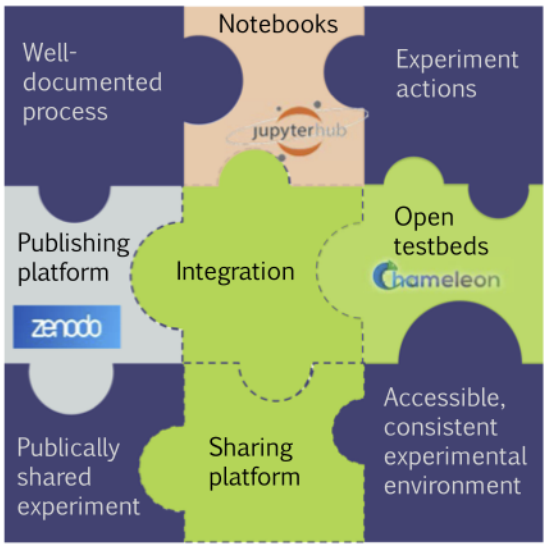

Figure 1. Interdependencies of the publishing ecosystem for digital artifacts.

To access clouds, users have to create digital artifacts such as a virtual machine or bare-metal images and orchestration templates that allow them to combine those images into complex topologies such as clusters or complicated network configurations. These artifacts are then deployed on the cloud or testbed to create the configuration of an experiment—typically refined as the experiment evolves—and ultimately snapshotted (i.e., saved) so that they can be used to recreate the experimental environment at a future date. This does not happen if a user is creating an experiment in a traditional lab setting where they are likely to use a machine configured by someone else, modify that configuration further to suit their experiment, but that they are also likely to share with colleagues who can introduce further modifications. In these cases, lack of sole control and convenient methods to capture and redeploy the complete experimental environment means that the user has to resort to effort-intensive and often incomplete note keeping. Thus, clouds formalize sharing: simply by virtue of using cloud interfaces users create digital artifacts compatible with a given cloud that go a long way—and sometimes all the way—to fully capturing their experimental environment. But if those digital artifacts can be used to fully capture the experimental environment, they can also be used to recreate it in future deployments—including by others—and thereby allow them to repeat an experiment.

How can we best leverage these natural advantages of using clouds, and reliably provide all that is needed to help users create repeatable experiments—and, most importantly, what is missing? Creating an experimental environment solves but half the problem—to fully package the experiment it needs to be combined with an expression of the experimental workflow itself, as well as any resulting data analysis or visualization. Further, it is even more important that the experiment can be easily redeployed by any future audience than that it should be easy to package in the first place: it is the actual repetition rather than theoretical availability that is the ultimate goal. To support this, any packaging will have to allow for introspection—so that an experiment can be replayed bit by bit, allowing for intermediate troubleshooting and introduction of variation—but also support "story telling" so that design and methodology choices can be documented and explained. A promising approach is the idea of "literate programming,"[5] an intermingling of process/code and explanatory text, expressed in a number of notebook implementations, the most popular of which is Jupyter.[6] Notebooks allow users to script their environment creation—as well as the actual experimental steps—while simultaneously documenting insight into how a result was obtained, from experimental process to data analysis. While a notebook will thus likely rely on other digital artifacts—images, data, or scripts—it provides a convenient "glue" that helps automate the experiment creation further, holds all its pieces together, and provides an easy interface for modifications.

Packaging experiments for repeatability is less than half the battle—the real holy grail is to ensure that they increase our understanding and enrich experience sharing—in other words, that they are actually repeated, verified, and modified. Conferences, professional organizations such as ACM, and funding agencies encourage and sometimes require providing reproducible artifacts. This is helpful, but sharing research in digital form won't become truly sustainable until there is a "marketplace" that creates real demand for such sharing—that is, until it becomes as natural to replay an experiment or analysis as it is right now to do a literature search and read relevant papers to learn from the insight of others before starting our own investigations. For this to happen, experiments and analyses need to be not only packaged in a reproducible form but also published in such a way that relevant experiments are easy to discover and reference. Our traditional research sharing environment is composed of an ecosystem of proceedings and journals, accessible through libraries and repositories, and discoverable via indexing systems; the practice of citing research allows us to not only reference ideas but also give credit to original authors. An ecosystem for sharing digital artifacts will likely develop gradually and organically as the existing one did, and it is natural to assume that they will share many common elements. There is one difference however in which sharing digital artifacts is different from sharing traditional research. Unlike traditional research digital media typically need systems that help interpret them: music needs to be played, data analyzed or visualized, and experiments deployed. Since clouds and testbeds are natural "players" for digital representations of experiments, this suggests that testbeds and clouds will become an essential element of such digital research sharing.

REPRODUCIBILITY IN A SHAREABLE CLOUD: CHAMELEON AS A CASE STUDY

The considerations above drove the design of the Chameleon testbed in many ways. First, we are conscious of the fact that users do not have as much insight into the details and evolution of the system hardware as is the case with individually held resources. This is important as memory enhancements, node repairs, and firmware upgrades can all introduce variability, which can impact experiments and the extent to which they can be repeated. To counteract that, we provide detailed, automatically maintained, and rigorously up-to-date hardware descriptions that are also versioned every time system hardware changes in any way. To illustrate the rate of change, over the last 5 years Chameleon went through over a hundred different hardware configuration versions. We also seek to combine convenience with control by allowing users to allocate this hardware at different levels: from model-based descriptions (e.g., "I need 4 nodes with memory of at least 2GB per core") that can be instantiated over different hardware configurations fulfilling that condition, to indicating specific nodes (often needed in performance variability or power management studies),

through intermediate descriptions (e.g., "I need 4 Haswell nodes").

As in other clouds, Chameleon users produce digital artifacts as a side-effect of using the system. Since its public availability at the end of July 2015 users produced over 130 000 images and over 35 000 orchestration templates, all of them representing experimental environments of various types. These images can be versioned to track the evolution of an experimental environment, shared at various levels, and ultimately published in a testbed catalog. This still fell short of a holistic experiment packaging strategy so following some trial and error in late summer of 2018, we integrated JupyterLab with the testbed to provide an explicit tool that can be used to provide that function.[7] The integration allows users to log into the Chameleon Jupyter server with their testbed credentials, which are then also implicitly bound to the user's notebook, allowing them to call Chameleon API functions directly within their code blocks and use them to construct experimental environments, e.g., a cluster of bare metal instances connected via an isolated network. A notebook constructed in this way usually contains not only a recipe for deploying the experimental environment but also the experimental workflow, including references to software and data. Most importantly, it can be rerun in a similar manner to that in which it was constructed to repeat the experiment as well as explicitly modified to introduce variation.

A Chameleon experiment can now be fully represented as a collection of digital artifacts: images, software, orchestration templates, notebooks, or data. These components can be put together in many different ways: some, like images, are testbed-specific—others, like data, are specific only to the experiment; some users will prefer to rely heavily on orchestration, others may find scripting more convenient—while we seek to provide practical solutions to our user community, we also strive not to limit their choices for sharing. We, therefore, sought ways of supporting the publishing of experiments that were testbed-independent while providing the essential research sharing characteristics of archival storage and citation. We found the answer in the Zenodo digital publishing platform[8] that allows users to publish depositions consisting of collections of related digital artifacts—or links to such digital objects (e.g., images that can be stored with the testbed that can deploy them)—together representing an experiment and assigns to them digital object identifiers for ease of sharing and citation. An integration of Chameleon and Zenodo achieved last year[9] gives our user community direct access to research artifacts already published that can be imported into Chameleon—as well as a convenient way to export representations of their experiments at the click of a button. Ongoing work on packaging influential experiments and implementing a search platform identifying ones relevant to a specific line of inquiry will ultimately give our users a way to discover—and hopefully replay—experiments of others.

DISCUSSION

Clouds represent a shareable instrument—thus, moving research to clouds should in principle increase the opportunity for sharing. What are the limitations of this opportunity? First, not all types of clouds are suitable for all types of research: computer science experiments require detailed architecture specifications, others may require some information (e.g., single versus double precision), and yet others can leverage almost any platform more economically. Most commercial clouds offer a variety of "instances," sometimes with vaguely described properties such as "high I/O bandwidth" that may be mapped to a different type of resource for every deployment. Even when the instances correspond to a specific hardware type, the available information is often limited and little or no versioning information is provided. While the repeatability of many computations may not be impacted by this loose resource model, work that explores power management or performance variability often needs to be carried out on the same hardware to eliminate sources of variability. Similarly, many CS experiments (such as performance studies) require the strong isolation of bare metal while other types of experiments can be run equally well and more economically using virtual machines or containers. Different types of research will thus map to different cloud platforms.

The images users create when using clouds represent an opportunity—but also a potential liability in the repeatability debate. This is because while an image snapshot provides an exact and easy to deploy representation of an experimental environment, it often does not contain the recipe on how it was created and may become outdated relatively quickly as the software it contains ages. In contrast,

using a base image combined with a configuration mechanism, e.g., scripts, orchestration, or notebooks, provides that recipe and the implied possibility of an update—but usually requires significantly more effort to create, deploy, and maintain—and may also need to be carefully managed to prevent introducing variation (e.g., a common issue is installing a latest version rather than a specific version of a software tool). The snapshot method is convenient for short-term repeatability (~6 months) that may be suitable for providing availability during, e.g., paper review period and initial experimentation with results; the latter is more appropriate for ensuring that an experiment may be repeated in the longer-term. To an extent these two different approaches mirror the difference between repeatability and reproducibility[10]: one provides an exact replica, the other opens the door to recreating the process under different conditions and via different means.

Finally, leveraging artifacts created in the cloud, while convenient, comes with the danger of creating silos: sets of experiments that become repeatable only on a certain cloud or testbed but not across them. While this puts more emphasis on the importance of cloud/testbed interoperability it also creates an additional set of challenges: what properties of clouds need to be preserved for digital representations of experiments to be portable for repeatability from one cloud to another? However, as with the ability to control hardware mappings, types of isolation, and image construction methodologies, different types of research may map to different types of clouds which may mean that a certain extent of segregation is natural and desirable.

Making research reproducible—as well as actually reproducing it—both take time. All too often this means that scientists face a reproducibility dilemma: on the packaging end, they have to choose whether they should invest this time into making an experiment reproducible at the cost of pursuing new ideas—on the reproducibility end, they have to decide if they should spend time on replaying somebody else's experiment or focus on their own work. This dilemma will be hard to resolve unless the time investment can be made manageable on both ends. A promising way of doing that is changing our research practices such that digital artifacts—and especially the experiments we create to bolster our investigations and arguments—are produced in a way that makes them reproducible by default. Using shareable instruments—such as clouds—helps us package experimental environments simply as a side-effect of using them. While these methods may emphasize short-term repetition, they get us closer to creating a research marketplace where playing with experiments of others becomes as viable and natural part of doing research as reading papers today. Combined with tools that close the gap to achieve a fully packaged experiment they define a potential that just might help us change the way we do science.

REFERENCES

1. K. Keahey *et al.*, "Lessons learned from the chameleon testbed," in *Proc. USENIX Annu. Tech. Conf.*, Jul. 2020, pp. 219–233.
2. OpenStack. Accessed: 2020. [Online]. Available: https://docs.openstack.org
3. I. Baldin *et al.*, "FABRIC: A national-scale programmable experimental network infrastructure," *IEEE Internet Comput.*, vol. 23, no. 6, pp. 38–47, Nov./Dec. 1, 2019, doi: 10.1109/MIC.2019.2958545.
4. Platforms for Advanced Wireless Research (PAWR). [Online]. Available: https://advancedwireless.org
5. D. E. Knuth, "Literate programming," *Comput. J.*, vol. 27, no. 2, pp. 97–111, 1984.
6. Project Jupyter. [Online]. Available: https://jupyter.org/
7. J. Anderson and K. Keahey, "A case for integrating experimental containers with notebooks," in *Proc. 11th IEEE Int. Conf. Cloud Comput. Technol. Sci.*, Dec. 2019, pp. 151–158.
8. Zenodo. [Online]. Available: https://zenodo.org
9. M. King, J. Anderson, and K. Keahey, "Sharing and replicability of notebook-based research on open testbeds," in *Proc. Int. Conf. High Performance Comput., Netw., Storage Anal.*, Nov. 2019.
10. D. G. Feitelson, "From repeatability to reproducibility and corroboration," *ACM SIGOPS Operating Syst. Rev.*, vol. 49, no. 1, pp. 3–11, 2015.

Kate Keahey is a Senior Scientist with Argonne National Laboratory, Lemont, IL, USA, and the University of Chicago, Chicago, IL, USA. She is the PI of the NSF Chameleon project that provides an open, deeply reconfigurable, and large-scale experimental platform for computer science research as well as Cofounder and Co-Editor-in-Chief of the SoftwareX journal that publishes software as a scientific instrument. Her research interests focus on cloud computing, resource management, and reproducibility. Contact her at keahey@anl.gov.