Chameleon: A Large-Scale, Deeply Reconfigurable Testbed for Computer Science Research

Kate Keahey*, Joe Mambretti†, Paul Ruth‡ and Dan Stanzione§
*Argonne National Laboratory, Lemont, IL
Email: keahey@anl.gov
†Northwestern University, Evanston, IL
Email: j-mambretti@northwestern.edu
‡Renaissance Computing Institute, Chapel Hill, NC
Email: pruth@renci.org
§Texas Advanced Computing Center, Austin, TX
Email: dan@tacc.utexas.edu

Computer Science experimental testbeds allow investigators to explore a broad range of different state-of-the-art hardware options, assess scalability of their systems, and provide conditions that allow deep reconfigurability and isolation so that one user does not impact the experiments of another. An experimental testbed is also in a unique position to support methods facilitating experiment analysis and improve repeatability and reproducibility of experiments. Providing these capabilities at least partially within a commodity framework improves the sustainability of systems experiments and thus makes them available to a broader range of experimenters.

Chameleon [1] is an open, large-scale, deeply reconfigurable testbed built specifically to support the features described above. The project began in 2014 and became publicly available in July 2015. To date, the testbed has supported 3,000+ users working on 500+ research and education projects. Community projects range from systems research developing new operating systems, virtualization methods, performance variability studies, and power management research to projects in software defined networking, artificial intelligence, and resource management.

Experiments of this type, cannot be supported by submitting jobs to batch schedulers. To cover the broadest possible range of experimental requirements, Chameleon supports a reconfigurable bare-metal system giving users full control of the software stack including root privileges, kernel customization, console access, as well as the ability to experiment with software defined networking using innovative features such as the Bring-Your-Own-Controller (BYOC) functionality [2, 3]. While most testbed resources are configured with Chameleon Infrastructure (CHI) which provides this type of user access and control, a small part of the system is configured as a virtualized KVM cloud to balance the need for finer-grained resource sharing sufficient for some projects, with coarse-grained and stronger isolation properties of bare metal.

Chameleon hardware balances the need to support experiments at scale with the need for diversity. The need for scale is satisfied by a large-scale homogeneous partition of nearly 15,000 cores, 5PB of total disk space, hosted across two sites connected by 100 Gbps network, the University of Chicago and the Texas Advanced Computing Center (TACC). The diversity of hardware configurations and architectures is reflected by support for innovative networking solutions including SDN-enabled Corsa switches and Infiniband support, accelerators such as FPGAs and a range of different GPU technologies, nodes with storage hierarchies containing a mix of large RAM, non-volatile memory, SSDs, and HDDs, a diversity x86 technologies, as well as support for non-x86 architectures such as ARMs.

Unlike traditional Computer Science experimental systems which have overwhelmingly been configured by technologies developed in-house, Chameleon adapted OpenStack, a mainstream open source cloud technology, to provide its capabilities. This has a range of practical benefits including familiar interfaces for users and operators, workforce development potential, leverage of contributions from a large development community, and the potential to contribute to infrastructure used by millions of users worldwide. The Chameleon team leveraged the latter opportunity in particular, by contributing to OpenStack the Blazar component that implements management of allocatable resources [4] via advance reservations. The most commonly used allocatable resources are compute nodes, but can also include networks, IP addresses, and can be extended to support other resources such as IoT devices. Since September 2017 the Blazar service has now been recognized as one of the OpenStack top level components. Beyond practical benefits, configuring an experimental platform as a cloud also provides a direct answer in the debate of whether Computer Science systems research can be supported on clouds. It is also a convenient means of influencing that debate through direct mainstream contributions such as the one described above.

While OpenStack is an open source system, customizing, extending, and augmenting it so that it presents a complete and viable platform for Computer Science experimentation requires additional work. Over the years of managing the system we have also developed many tools that make the operation of Chameleon easier and more cost-effective, automating problem discovery and repair [5]. The holistic system is called CHameleon Infrastructure (CHI) and can be packaged and shared with others intending to operate experimental testbeds. Our approach to packaging Chameleon is called CHI-in-a-Box and addresses three scenarios: (1) independent testbed, where a provider wants to configure a new, potentially private, testbed independent of Chameleon, (2) Chameleon Associate Site, i.e., a site that directly contributes resources to the Chameleon testbed, and (3) Part-Time Associate, for cases where a site contributes resources to Chameleon on a part-time only basis. The Chameleon team will provide limited operational support and full user support to partners participating as Associate Site and Part-Time Associate making it a good option for providers who want to provide their resources to a wide user base. To date one site, at the Northwestern University, has joined Chameleon as an Associate Site and two others are exploring using CHI-in-a-Box as independent sites.

Although the primary purpose of open testbeds is to provide resources to users who would not be able to satisfy their experimental needs otherwise, an important side-effect is that multiple users and user groups have access to the same resources, that are compatible with the same experimental artifacts, such as appliances/images or orchestration templates. This creates conditions which allow users to share experiments and replicate each other's work more easily and creates an opportunity to foster good experimental practices as well as create a sharing ecosystem. To facilitate the creation of such ecosystem we have developed two mechanisms that we call, respectively, reproducibility by side-effect, and reproducibility by default.

Reproducibility by side-effect is intended to aid users in much the same the Linux "history" command allows you to see what commands you typed while working on a problem. Similarly in a testbed, we often need to refresh our memory on the exact configuration, conditions, or steps taken in the conduct of an experiment. Chameleons Experiment Précis [6] captures all the distributed events generated in a testbed by a user, and presents them with a summary (a précis) of an experiment; the user can then filter or preview the events to include only the relevant ones, thus working with an accurate and impartial record of their work. The précis can be further used to generate a description of the experiment in English or potentially a set of experimental artifacts - images, orchestration templates, or commands - that will reproduce the experiment.

Searching for the best expression of an experiment led us to experiment with notebooks which combine *ideas* in the form of text, with the *experimental process* expressed as code and *results* expressed as data processing. Using notebooks, users can develop their experiments step by step; as each step is configurable and the notebooks can be shared, each represents a convenient vehicle for repeating and replicating an experiment. To facilitate the use of notebooks we integrated Jupyter with Chameleon by providing a JupyterHub server for our users. We also integrated python and bash libraries that capture the interface to the testbed. While notebook code is generally limited to executing in limited containers (such as Docker), with this integration, Chameleon users can define arbitrarily complex containers/environments powered by the testbed - and then use them from their Jupyter notebooks to run complex experiments. The integration thus combines the flexibility of notebooks with the power of experimental testbeds and allows us to make our user's work more efficient.

Acknowledgment

These results were supported by the NSF awards 1419141 and 1743358 and, in part, by the U.S. DOE, Office of Science, contract number DE-AC02-06CH11357.

References

[1] K. Keahey, P. Riteau, D. Stanzione, T. Cockerill, J. Mambretti, P. Rad, and P. Ruth, "Chameleon: a Scalable Production Testbed for Computer Science Research," in *Contemporary High Performance Computing: From Petascale toward Exascale*, 1st ed., ser. Chapman Hall/CRC Computational Science, J. Vetter, Ed. Boca Raton, FL: CRC Press, May 2019, vol. 3, ch. 5, pp. 123–148.

[2] P. Ruth, M. Cevik, K. Keahey, and P. Riteau, "Wide-area Software Defined Networking Experiments using Chameleon," in *Proceedings of the IEEE Conference on Computer and Networking Experimental Research using Testbeds (CNERT 2019).* IEEE Press, 2019.

[3] D. Bhat, J. Anderson, P. Ruth, M. Zink, and K. Keahey, "Application-based QoE Support with P4 and OpenFlow," in *Proceedings of the IEEE Conference on Computer and Networking Experimental Research using Testbeds (CNERT 2019).* IEEE Press, 2019.

[4] K. Keahey, P. Riteau, J. Anderson, and Z. Zhen, "Managing Allocatable Resources," in *Proceedings of The IEEE International Conference on Cloud Computing (CLOUD 2019).* IEEE Press, 2019.

[5] K. Keahey, J. Anderson, P. Ruth, J. Colleran, C. Hammock, J. Stubbs, and Z. Zhen, "Operational Lessons from Chameleon," in *Proceedings of the Humanware Advancing Research in the Cloud (HARC'19) at PEARC'19.* ACM, 2019.

[6] S. Wang, Z. Zhen, J. Anderson, and K. Keahey, "Reproducibility as Side Effect," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage, and Analysis (SC'18 Poster).* IEEE Press, 2018.