



www.chameleoncloud.org

FROM TESTBED TO INSTRUMENT: BUILDING EXPERIMENT SUMMARIES AND REPEATABILITY SERVICES IN CHAMELEON

Kate Keahey

Mathematics and CS Division, Argonne National Laboratory
Computation Institute, University of Chicago

keahey@anl.gov

October 10, 2017

GENI NICE Conference

Toronto, Canada

OCTOBER 11, 2017

I



CHAMELEON PHASE 1 TESTBED IN A NUTSHELL

- ▶ **Open** production testbed for **Computer Science Research**
 - ▶ Available since 07/2015, just renewed for another 3 years
 - ▶ Currently 1,800+ users, 300+ projects, 100+ institutions, 100+ publications
- ▶ **Large-scale:** “Big Data, Big Compute research”
 - ▶ ~650 nodes (~15,000 cores), 5 PB of storage distributed over 2 sites connected with 100G network – new hardware with user allocatable network switches coming soon!
 - ▶ Operated as a single instrument
- ▶ **Reconfigurable:** “As close as possible to having it in your lab”
 - ▶ Deep reconfigurability (bare metal) and isolation
 - ▶ Power on/off, reboot from custom kernel, serial console access, etc.
- ▶ Blueprint for a **sustainable** production testbed: “cost-effective to deploy, operate, and enhance”
 - ▶ Powered by OpenStack with bare metal reconfiguration (Ironic)
 - ▶ Chameleon team contributions (advance reservations) now recognized as official OpenStack component!

TOWARDS A SCIENTIFIC INSTRUMENT



- ▶ **Deploy:** what we are doing today – and more
- ▶ **Capture:** observe, monitor, measure – easily
- ▶ **Record:** a comprehensive “active record”
 - ▶ Re-examine, share/publish, review, re-play

CAPTURING EXPERIMENT INFORMATION

- ▶ Everything in a testbed is a recorded event
 - ▶ The resources you used
 - ▶ The appliance/image you deployed
 - ▶ The monitoring information your experiment generated
 - ▶ Plus any information you choose to share with us: e.g., “start power_exp_23” and “stop power_exp_23”
-
- ▶ **Experiment précis:** information about your experiment made available in a “consumable” form
 - ▶ (Bonus: it can be integrated with many existing tools, e.g., Jupyter or Grafana)...

For screencast see: <https://www.youtube.com/user/ChameleonCloud>
Experiment Metric Visualization

The screenshot shows the Chameleon OpenStack dashboard. The main heading is "Instances". Below the heading, there are filter buttons for "Instance Name" and "Filter", and action buttons for "Launch Instance" and "Terminate Instances". A table lists the instances:

Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Time since created	Actions
power-experiment-node0	CC-CentOS7	10.40.0.140 Floating IPs: 129.114.108.70	baremetal	nick-mp17	Active	climate:2f4bed2c-daa6-4dd9-9dba-6af8a41b2221	None	Running	5 minutes	Disassociate Flo

Displaying 1 item

REPEATABILITY: THE FOUNDATION

- ▶ Testbed versioning
 - ▶ Fine-grain representation
 - ▶ Automated discovery and updates
 - ▶ 53 versions since public availability – and counting
 - ▶ Still working on: better firmware version management
- ▶ Appliance management
 - ▶ Configuration, versioning, publication
 - ▶ Still working on: repository vs catalog connection
- ▶ Monitoring and logging
 - ▶ Still working on: making it accessible in easier ways

REPEATABILITY: MOVING TOWARDS

- ▶ Experiment précis: closing the gap between resource versions, appliances, and events
- ▶ Publishing experiment précis
- ▶ Integration with popular tools
- ▶ From experiment précis to experiment replays
 - ▶ Model-based experiment capture
 - ▶ Re-play tools

PARTING THOUGHTS

- ▶ Repeatability/replicability versus reproducibility
- ▶ The barrier to repeatability/reproducibility is the cost
 - ▶ Lost opportunity to do other research
- ▶ Make it cheaper to create repeatable experiments...
- ▶ ...than non-repeatable experiments
 - ▶ Incentives: reviews, iterations, challenges/awards, supporting students moving through the lab, etc.
 - ▶ Tools: notebooks, visualizations, data processing, experimental workflow management, etc.
- ▶ Infrastructure projects are very well positioned to solve a large part of the problem – but not all of it – we need partners!