CHAMELEON:
A NEW ECOSYSTEM FOR
EXPERIMENTAL COMPUTER SCIENCE

Kate Keahey
Mathematics and CS Division, Argonne National Laboratory CASE, University of Chicago
keahey@anl.gov

November 8, 2018
Boston University seminar
CHAMELEON IN A NUTSHELL

► **Deeply 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.
  ► But also – modest KVM cloud for ease of use

► **Combining large-scale and diversity:** “Big Data, Big Compute research”
  ► **Large-scale:** ~660 nodes (~15,000 cores), 5 PB of storage distributed over 2 sites connected with 100G network...
  ► ...and **diverse:** ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
  ► **Coming soon:** more storage, more accelerators

► Blueprint for a **sustainable** production testbed: “cost-effective to deploy, operate, and enhance”
  ► Powered by OpenStack with bare metal reconfiguration (Ironic)
  ► Chameleon team contribution recognized as official OpenStack component

► **Open, collaborative, production** testbed for **Computer Science Research**
  ► Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
  ► Currently 2,700+ users, 450+ projects, 100+ institutions
CHAMELEON HARDWARE (DETAILS)

▶ “Start with large-scale homogenous partition”
▶ 12 Haswell Standard Cloud Units (48 node racks), each with 42 Dell R630 compute servers with dual-socket Intel Haswell processors (24 cores) and 128GB RAM and 4 Dell FX2 storage servers with 16 2TB drives each; Force10 s6000 OpenFlow-enabled switches 10Gb to hosts, 40Gb uplinks to Chameleon core network
▶ 2 SkyLake Standard Cloud Units (32 node racks); Corsa (DP2400 & DP2200) switches, 100Gb uplinks to Chameleon core network
▶ Allocations can be an entire rack, multiple racks, nodes within a single rack or across racks (e.g., storage servers across racks forming a Hadoop cluster)

▶ Shared infrastructure
▶ 3.6 + 0.5 PB global storage, 100Gb Internet connection between sites
▶ “Graft on heterogeneous features”
▶ Infiniband with SR-IOV support, High-mem, NVMe, SSDs, GPUs (22 nodes), FPGAs (4 nodes)
▶ ARM microservers (24) and Atom microservers (8), low-power Xeons (8)
▶ Coming soon: more nodes (CascadeLake), and more accelerators
REQUIREMENTS FOR EXPERIMENTAL WORKFLOW

- **discover** resources
- **allocate** resources
- **configure and interact**
- **monitor**

- Fine-grained
- Complete
- Up-to-date
- Versioned
- Verifiable

- Advance reservations & on-demand
- Isolation
- Different resource types

- Deeply reconfigurable
- Appliance catalog
- Snapshotting
- Complex Appliances
- Network Isolation

- Hardware metrics
- Fine-grained information
- Aggregate and archive
BUILDING CHI (CHAMELEON INFRASTRUCTURE)

- Requirements for core functionality (proposal stage)
  - Interviews with ~20 research groups
- Architecture: **discover, provision, configure**, and **monitor**
- Technology Evaluation and Risk Analysis
  - Many options: Grid’5000, Nimbus, LosF, OpenStack
  - Final round: Grid’5000 and OpenStack
- Criteria: sustainability as design criterion
  - **Does it fit our purpose?** Feature coverage, incl. ease of use
  - **Can we customize it?** Open-source, configurable, extendable
  - **Can we rely on it?** Stable, scalable, supported
  - Can a CS testbed be built from commodity components?
- A mix of technologies with lots of tweaks (aka “special sauce”)
  - Grid’5000 for resource discovery and hardware verification
  - OpenStack for the rest (using Blazar, Ironic, and core OpenStack services)
- Core functionality built in just 3 months after evaluation
WHAT IS OPENSTACK?

- Leading open-source IaaS implementation... and more

Traditional software

OpenStack

- Community: ~1,500-2,000 developers contributing to each release including many big companies contributing, e.g. Huawei, Red Hat
- Deployment base:
  - 2017 user surveys logged 1,000 unique deployments (~millions of end users)
  - 60 public cloud data centers, from e.g. Rackspace, OVH
  - Large-scale deployments, e.g. ~100sK cores at CERN
THE MISSING COMPONENT: OPENSTACK BLAZAR

- **Advanced reservation service** for OpenStack
- Originally developed 2013-2014 in the context of power management research
- From early 2015: adaptation for Chameleon
  - Improve stability, integration with Ironic
  - Dashboard improvements (Gantt chart)
  - Incremental operational improvements
- Fall 2016: revival
  - Joined forces with NTT and others working on capacity reservation for NFV
- **Official OpenStack project** in Sep 2017
OPENSTACK: LESSONS LEARNED

- The good
  - Large community rapidly developing new features
  - Common requirements → shared effort
  - Commodity infrastructure for sustained use
  - Many users already familiar with OpenStack

- The bad
  - Large community rapidly developing new features
  - Complexity: need to understand core components
  - Some users assume Chameleon is like any OpenStack
SUPPORT FOR EXPERIMENTAL WORKFLOW

discover resources → allocate resources → configure and interact → monitor

Grid’5000 Resource Discovery

OpenStack:
- Nova
- Blazar
- Swift

OpenStack:
- Ironic
- Neutron
- Glance
- Heat
Other
- Appliances++
- Snapshotting
Network Isolation

OpenStack
- Gnocchi Agents, custom integration, etc.

CHI = 65%*OpenStack + 10%*G5K + 25%*”special sauce”
SPECIAL SAUCE, LATELY...

Networking:

- **Multi-tenant networking** allows users to provision isolated L2 VLANs and manage their own IP address space (since Fall 2017)

- **Stitching** dynamic VLANs from Chameleon to external partners (ExoGENI, ScienceDMZs) (since Fall 2017)

- VLANs + AL2S connection between UC and TACC for **100G experiments** (since Spring 2018)

- **BYOC—Bring Your Own Controller**: isolated user controlled virtual OpenFlow switches (Summer 2018)

- New lease management features, multi-region configuration, power consumption metrics, whole disk image boot for ARM nodes, serial console access, etc.

- And many more...
  - Appliances, usability improvements, upgrades, etc.
VIRTUALIZATION OR CONTAINERIZATION?

- Yuyu Zhou, University of Pittsburgh
- Research: lightweight virtualization
- Testbed requirements:
  - Bare metal reconfiguration, isolation, and serial console access
  - The ability to “save your work”
  - Support for large scale experiments
  - Up-to-date hardware

SC15 Poster: “Comparison of Virtualization and Containerization Techniques for HPC”
EXASCALE OPERATING SYSTEMS

- Swann Perarnau, ANL
- Research: exascale operating systems
- Testbed requirements:
  - Bare metal reconfiguration
  - Boot from custom kernel with different kernel parameters
  - Fast reconfiguration, many different images, kernels, params
  - Hardware: accurate information and control over changes, performance counters, many cores
  - Access to same infrastructure for multiple collaborators

*HPPAC'16 paper: “Systemwide Power Management with Argo”*
CLASSIFYING CYBERSECURITY ATTACKS

- Jessie Walker & team, University of Arkansas at Pine Bluff (UAPB)
- Research: modeling and visualizing multi-stage intrusion attacks (MAS)
- Testbed requirements:
  - Easy to use OpenStack installation
  - A selection of pre-configured images
  - Access to the same infrastructure for multiple collaborators
CREATING DYNAMIC SUPERFACILITIES

- NSF CICI SAFE, Paul Ruth, RENCI-UNC Chapel Hill
- Creating trusted facilities
  - Automating trusted facility creation
  - Virtual Software Defined Exchange (SDX)
  - Secure Authorization for Federated Environments (SAFE)
- Testbed requirements
  - Creation of dynamic VLANs and wide-area circuits
  - Support for slices and network stitching
  - Managing complex deployments
DATA SCIENCE RESEARCH

- ACM Student Research Competition semi-finalists:
  - Blue Keleher, University of Maryland
  - Emily Herron, Mercer University

- Searching and image extraction in research repositories

- Testbed requirements:
  - Access to distributed storage in various configurations
  - State of the art GPUs
  - Easy to use appliances and complex deployments
Divyashri Bhat, UMass Amherst

Research: application header based traffic engineering

Testbed requirements:
- Distributed testbed facility
- BYOC – the ability to write an SDN controller specific to the experiment
- Multiple connections between distributed sites
BUILDING AN ECOSYSTEM

- Helping hardware providers interact
  - Bring Your Own Hardware (BYOH)
  - CHI-in-a-Box: deploy your own Chameleon
- Helping scientists interact
  - Leveraging the common denominator
  - Integrating tools for experiment management
  - Making reproducibility easier
  - Facilitating sharing
CHI-IN-A-BOX

- CHI-in-a-box: packaging a commodity-based testbed
- CHI-in-a-box scenarios
  - **Testbed extension**: join the Chameleon testbed: generalize and package + define operations models
  - **Part-time extension**: define and implement contribution models
  - **New testbed**: generalize policies
- Available since Summer 2018
- New Associate Site at Northwestern
  - Nodes with 100G network cards
REPRODUCIBILITY DILEMMA

Should I invest in making my experiments repeatable? Should I invest in more new research instead?

- Reproducibility as side-effect: lowering the cost of repeatable research
  - Example: Linux “history” command
  - From a meandering scientific process to a recipe
- Documenting the process: interactive papers
REPEATABILITY MECHANISMS IN CHAMELEON

- Testbed versioning (collaboration with Grid’5000)
  - Based on representations and tools developed by G5K
  - >50 versions since public availability – and counting
  - Still working on: better firmware version management

- Appliance management
  - Configuration, versioning, publication
  - Appliance meta-data via the appliance catalog
  - Orchestration via OpenStack Heat

- Monitoring and logging

However... the user still has to keep track of this information
KEEPPING TRACK OF EXPERIMENTS

- 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
REPEATABILITY: EXPERIMENT PRÉCIS

- OpenStack services
- Instance monitoring
- Infrastructure monitoring
- User events

Experiment précis

Store and share

Orchestrator (Heat)
EXPERIMENT PRÉCIS IMPLEMENTATION

User

Experiment
1. Create a lease
2. Launch instances
3. Add networks
4. ......

RabbitMQ

Testbed

Experiment Precis
1. lease_start
2. Instance_start
3. ......

Listener

Chameleon Commandline

English Description

Experiment Precis Formatter

Events

SC18 poster: “Reproducibility as Side-Effect”
EXPERIMENT PRÉCIS: A CASE STUDY

Based on Wang et al., Understanding and Auto-Adjusting Performance-Sensitive Configurations. ASPLOS, 2018
REPEATABILITY: EXPERIMENT PRÉCIS

- OpenStack services
- Instance monitoring
- Infrastructure monitoring
- User events

Experiment précis

?
WHAT DOES IT MEAN TO DOCUMENT A PROCESS?

- **Requirements**
  - Human readable/modifiable format
  - Integrates well with ALL aspects of experiment management
  - Bit by bit replay – allows for bit by bit modification (and introspection) as well – element of interactivity
  - Support story telling: allows you to explain your experiment design and methodology choices
  - Has a direct relationship to the actual paper that gets written
  - Can be version controlled and easily shared
  - Sustainable, a popular open source choice

- **Implementation options**
  - Orchestrators: Heat, the dashboard, and Flame
  - Notebooks: Jupyter, Nextjournal
JUPYTER ON CHAMELEON

Run experiment and get output

Log in to Jupyter Notebook server
Generate output suitable for publishing

\[ \text{LATEX} \]
JUPYTER ON CHAMELEON

username

********

Re-run experiment with modifications

Log in to Jupyter Notebook server

Compare results

What if...
CHAMELEON JUPYTER INTEGRATION

- Storytelling with Jupyter
  - text, process, results
- Jupyter as an interface to Chameleon
  - All the main testbed functions
  - Jupyter.chameleoncloud.org
  - “Hello World” template
  - Save and share via our object store
- Create and modify your experiment bit by bit
- Screencast of a complex experiment
  - https://vimeo.com/297210055
HOW DO I GET STARTED?

» Go to www.chameleoncloud.org
» Click the big orange Get started button
  » Create account
  » Create or join a project/allocation (10,000 SUs)
  » Follow the documentation to start a lease
» Keep in touch and let us know how we can help!
PARTING THOUGHTS

- Chameleon: rapidly evolving testbed
  - Changes as the research frontier changes

- Testbeds are not just experimentation platforms
  - Ecosystem: a meeting place of users sharing resources and research
  - Common/shared platform is a “common denominator” that can eliminate much complexity that goes into sharing and reproducibility: it allows you to do something interesting and powerful and then share it

- Get engaged – come to the User Meeting!
  - https://www.chameleontcloud.org/user-meeting-2019/
Questions?

www.chameleoncloud.org
keahey@anl.gov