THE MANY COLORS OF CHAMELEON

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

February 6, 2019
Chameleon User Meeting
CHAMELEON IN A NUTSHELL

- We like to change: testbed that adapts itself to your experimental needs
  - Deep reconfigurability (bare metal) and isolation (CHI) – but also ease of use (KVM)
  - CHI: power on/off, reboot, custom kernel, serial console access, etc.

- We want to be all things to all people: balancing large-scale and diverse
  - Large-scale: ~large homogenous partition (~15,000 cores), 5 PB of storage distributed over 2 sites (now +1!) connected with 100G network...
  - ...and diverse: ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.

- We want to last: cost-effective to deploy, operate, and enhance
  - Powered by OpenStack with bare metal reconfiguration (Ironic)
  - Chameleon team contribution recognized as official OpenStack component

- We live to serve: open, production testbed for Computer Science Research
  - Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
  - Currently ~3,000 users, ~500 projects, ~100 institutions
CHAMELEON HARDWARE

Chameleon Core Network
100Gbps uplink public network (each site)

Core Services
3.5PB Storage System

Core Services
0.5 PB Storage System

Heterogeneous Cloud Units
- GPUs (K80, M40, P100), FPGAs, NVMe, SSDs, IB, ARM, Atom, low-power Xeon

GENI and other partners

Chameleon Associate Site Northwestern

SkyLake
Standard Cloud Unit
32 compute
Corsa Switch x2

SkyLake
Standard Cloud Unit
32 compute
Corsa Switch x1

Haswell
Standard Cloud Unit
42 compute
4 storage x2

Haswell
Standard Cloud Unit
32 compute
Corsa Switch x10

SkyLake
Standard Cloud Unit
32 compute
Corsa Switch x1

Chicago
Austin
EXPERIMENTAL WORKFLOW

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

- allocate resources
  - Advance reservations
  - On-demand
  - Isolation
  - Across resource types

- configure and interact
  - Deeply reconfigurable
  - Appliance catalog
  - Snapshotting
  - Complex Appliances
  - Network Isolation

- monitor
  - Hardware metrics
  - Fine-grained data
  - Aggregate
  - Archive

\[ \text{CHI} = 65\% \times \text{OpenStack} + 10\% \times \text{G5K} + 25\% \times \text{"special sauce"} \]
IMPROVING THE PLATFORM: 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 (since Summer 2018)
- Managing multiple stitches (since Fall 2018)
- VLAN reservations (since Winter 2019), floating IP reservations coming soon!
BRING-YOUR-OWN-CONTROLLER (BYOC)

- Software Defined Networking (SDN)
  - Corsa Virtual Forwarding Context (VFC)
  - OpenFlow 1.3
  - User defined controller
    - Within Chameleon or anywhere on the Internet
  - Available on Skylake nodes
- Supported capabilities
  - SDN experiments
  - Experiments requiring non-standard networking capabilities

Diagram:
- Corsa Switch
  - VFC (Tenant A)
  - VFC (Tenant B)
  - Compute Node (Tenant A)
  - Compute Node (Tenant A)
  - Compute Node (Tenant B)
  - Compute Node (Tenant B)
  - OpenFlow Controller (Tenant A)
  - OpenFlow Controller (Tenant B)
EXTERNAL STITCHING

- Layer2 VLANs from Chameleon to external partners
  - ExoGENI, ScienceDMZs, Esnet, and AL2S
- VFCs with multiple L2 stitched links
  - Named VFCs
NETWORKING PATTERNS MADE EASY

- Sharednet1
  - Pre-configured local shared network
- Sharedwan1
  - Stitched shared network
  - Pre-configured
  - Connects UC and TACC
  - Up to 100 Gbps
  - Ask how to add it to your project!
IMPROVING THE PLATFORM: OTHER FEATURES

- Lease management: adding/removing nodes to/from a lease, notifications of lease start and impending termination
- Advance reservation orchestration
- Power and temperature metrics
- Whole disk image boot for ARM nodes
- New appliances (Hadoop, ExoGENI, BYOC examples) and a richer set of appliance features: FUSE module and networking support
- Usability features: multi-region configuration, single login to all web interfaces, better access to information, better error handling, software self-updates, better appliance publishing, documentation overhaul, etc.
- Chameleon traces are now available at www.scienceclouds.org
BEYOND THE PLATFORM: BUILDING AN ECOSYSTEM

- Helping hardware providers interact
  - Bring Your Own Hardware (BYOH)
  - CHI-in-a-Box: deploy your own Chameleon site

- Helping our user interact – with us but primarily with each other
  - Facilitating contributions of appliances, tools, and other artifacts: appliance catalog, blog as a publishing platform, and eventually notebooks
  - Integrating tools for experiment management
  - Making reproducibility easier

- Improving communication – not just with us but with our users as well
CHI-IN-A-BOX

- CHI-in-a-box: packaging a commodity-based testbed
  - First released in summer 2018, continuously improving
- CHI-in-a-box scenarios
  - Independent testbed: package assumes independent account/project management, portal, and support
  - Chameleon extension: join the Chameleon testbed (currently serving only selected users), and includes both user and operations support Part-time extension: define and implement contribution models
  - Part-time Chameleon extension: like Chameleon extension but with the option to take the testbed offline for certain time periods (support is limited)
- Adoption
  - New Chameleon Associate Site at Northwestern since fall 2018 – new networking!
  - Two organizations working on independent testbed configuration
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
- Reproducibility by default: documenting the process via 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
KEEPING 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

Experiment précis

OpenStack services
Instance monitoring
Infrastructure monitoring
User events

Store and share

Orchestrator (Heat)
EXPERIMENT PRÉCIS: A CASE STUDY

INTERACTIVE PAPERS

- What does it mean to document a process?
- Some requirements
  - Easy to work with: 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
  - Sustainable, a popular open source choice

- Implementation options
  - Orchestrators: Heat, the dashboard, and OpenStack Flame
  - Notebooks: Jupyter, NextJournal
CHAMELEON JUPYTER INTEGRATION

- Combining the ease of notebooks and the power of a shared platform
  - Storytelling with Jupyter: ideas/text, process/code, results
  - Chameleon shared experimental platform
- JupyterLab server for our users
  - Just go to jupyter.chameleoncloud.org and log in with your Chameleon credentials
- Chameleon/Jupyter integration
  - Alternative interface
  - All the main testbed functions
  - “Hello World” template

Screencast of a complex experiment: https://vimeo.com/297210055
SHARING, EXPERIMENTING, LEVERAGING

- Sharing Jupyter notebooks in Chameleon
  - Today: from home directory to sharing via our Swift storage with your project members
  - Challenges ahead: more flexible sharing policy implementation, integrating with github for better versioning and sharing support
- Automating experiments with Jupyter
PARTING THOUGHTS

- Physical environment: Chameleon is a rapidly evolving experimental platform
  - Originally: “Adapts to the needs of your experiment”
  - Now also: “Adapts to the needs of its community and the changing research frontier”
- Towards an Ecosystem: a meeting place of users and providers sharing resources and research
  - Testbeds are more than just experimental platforms
  - Common/shared platform is a “common denominator” that can eliminate much complexity that goes into systematic experimentation, sharing, and reproducibility
- Be part of the change: tell us what capabilities we should provide to help you share and leverage the contributions of others!