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

October 15, 2019
SDSC Seminar
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.

- Cloud++: leveraging mainstream cloud technologies
  - Powered by OpenStack with bare metal reconfiguration (Ironic) + “special sauce”
  - 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 3,500+ 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

SkyLake
Standard Cloud Unit
32 compute
Corsa Switch
x2

SkyLake
Standard Cloud Unit
32 compute
Corsa Switch
x1

Chameleon Associate Site Northwestern
GENI and other partners

Chicago
Austin

Haswell
Standard Cloud Unit
42 compute
4 storage
x2

Haswell
Standard Cloud Unit
32 compute
Corsa Switch
x1

Corsa
Switch
x2

Corsa
Switch
x1

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
x1

Corsa
Switch
x2

Corsa
Switch
x1

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
x1

Corsa
Switch
x2

Corsa
Switch
x1
“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
- 3 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/GPUs
EXPERIMENTAL WORKFLOW

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

**allocate resources**
- Allocatable resources: nodes, VLANs, IPs
- Advance reservations and on-demand
- Expressive interface
- Isolation

**configure and interact**
- Deeply reconfigurable
- Appliance catalog
- Snapshotting
- Orchestration
- Networks: stitching and BYOC

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

\[ \text{CHI} = 65\% \times \text{OpenStack} + 10\% \times G5K + 25\% \times \text{“special sauce”} \]
PHASE 2 ADDITIONS

- Better allocatable resources – it’s not just for the nodes!
  - Multiple resource management (nodes, VLANs, IP addresses), adding/removing nodes to/from a lease, lifecycle notifications, advance reservation orchestration
  - Chameleon blog: “Save the planet, use fewer IPs”

- All the networking that’s fit to experiment with!
  - Multi-tenant networking,
  - Stitching dynamic VLANs from Chameleon to external partners (ExoGENI, ScienceDMZs),
  - VLANs + AL2S connection between UC and TACC for 100G experiments
  - BYOC– Bring Your Own Controller: isolated user controlled virtual OpenFlow switches
  - Coming soon: stitching to commercial clouds

- Miscellaneous features – and there’s more...
  - Power metrics, usability features, new appliances, 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, parameters
  - 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 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 orchestration

www.chameleoncloud.org
ADAPTIVE BITRATE VIDEO STREAMING

- Divyashri Bhat, UMass Amherst
- Research: application header based traffic engineering using P4
- Testbed requirements:
  - Distributed testbed facility
  - BYOC – the ability to write an SDN controller specific to the experiment
  - Multiple connections between distributed sites
- https://vimeo.com/297210055

*LCN’18: “Application-based QoS support with P4 and OpenFlow”*
POWER CAPPING EXPERIMENTATION

- Harper Zhang, University of Chicago
- Research: hierarchical, distributed, dynamic power management system for dependent applications
- Testbed requirements:
  - Support for large-scale experiments
  - Complex appliances and orchestration (NFS appliance)
  - RAPL/power management interface
- Finalist for SC19 Best Paper and Best Student Paper
  - bit.ly/SC19PoDD

(Above) The performance delivered by each power management system across 5 power budgets for 11 coupled application pair.

(Right) Harper presents his research.

Testbed requirements:
- Support for large-scale experiments
- Complex appliances and orchestration (NFS appliance)
- RAPL/power management interface

Finalist for SC19 Best Paper and Best Student Paper
- bit.ly/SC19PoDD
AN OPEN PLATFORM
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
  - Creating compatible digital artifacts: tools, appliances, orchestration templates, notebooks, etc.
  - Publishing, sharing, and discovering artifacts: appliance catalog, blog as a publishing platform, etc.
  - Testbed as a “player” for common artifacts
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 by default -- interactive stories/papers -- tools to represent experiments in terms of sharable/publishable digital artifacts
- Reproducibility as side-effect: lowering the cost of repeatable research
  - A Linux “history” analogue
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... or could be
- 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)
INTERACTIVE PAPERS

- What does it mean to document a process?
- Some requirements
  - Easy to work with: human readable/modifiable format
  - One process to rule them all: integrates well with ALL aspects of experiment management
  - Bit by bit – allows for modification and introspection as well – reflects the meandering scientific process
  - 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 -- OpenStack Heat and Flame – a declarative approach
  - Notebooks -- Jupyter, NextJournal, and others – an imperative and integrative approach
CHAMELEON JUPYTER INTEGRATION

Combining the ease of notebooks and the power of a shared platform

- Storytelling with Jupyter: ideas/text, process/code, results
- Chameleon: sophisticated experimental containers in need of “storytelling”

JupyterLab server for our users

- Go to jupyter.chameleoncloud.org and use with your Chameleon credentials

Chameleon/Jupyter integration

- Python/bash interfaces to the testbed, storing and sharing, Chameleon credentials
- Named containers

Templates of existing experiments

Screencast of a complex experiment: https://vimeo.com/297210055

“A Case for Integrating Experimental Containers with Notebooks”, CloudCom 2019
SHARING, PUBLISHING, LEVERAGING

- We now have everything we need to share experiments
  - Ways to establish an experimental environment + player
  - Ways to document an experimental process
- But wait... how do I actually share them?
  - Send mail, Chameleon object store, github...
  - Publishing via Zenodo: store your experiments and make them citable via DOIs
- Creating bridges, integration
  - Import/Export from/to Zenodo
- Making research findable: the sharing platform

SC19 Poster: Sharing and Replicability of Notebook-Based Research on Open Testbeds
Well-documented process

Accessible, consistent code environment

Easy to find experiment

Notebooks

Open testbeds

Sharing platform

Integration

Experiment actions

Publicly shared experiment

Publishing platform
PARTING THOUGHTS

- Chameleon is a rapidly evolving experimental platform
  - A Cloud++ with extra capabilities to support Computer Science research – and beyond
  - Originally: “Adapts to the needs of your experiment”
  - Now also: “Adapts to the needs of its community and the changing research needs”

- 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...
  - ...as well as education!

- Be a part of the change: come to run experiments, stay to tell us about new requirements, join us to build new infrastructure capabilities