CHAMELEON: TAKING SCIENCE FROM CLOUD TO EDGE

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

July 20, 2021
Supercomputing Frontiers Europe 2021
CHAMELEON IN A NUTSHELL

- We like to change: a testbed that adapts itself to your experimental needs
  - Deep reconfigurability (bare metal) and isolation – but also a small KVM cloud
  - power on/off, reboot, custom kernel, serial console access, etc.
- Balance: large-scale versus diverse hardware
  - Large-scale: ~large homogenous partition (~15,000 cores), ~6 PB of storage distributed over 2 sites (UC, TACC) connected with 100G network
  - Diverse: ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
- Cloud++: CHameleon Infrastructure (CHI) via mainstream cloud tech
  - Powered by OpenStack with bare metal reconfiguration (Ironic) + “special sauce”
  - Blazar contribution recognized as official OpenStack component
- We live to serve: open, production testbed for Computer Science Research
  - Started in 10/2014, available since 07/2015, renewed in 10/2017, and recently till end of 2024
  - Currently 5,500+ users, 700+ projects, 100+ institutions, 300+ publications
BY THE NUMBERS

- 300+ Papers published
- 700+ Projects
- Over 5,500 Users
- 5+ Years Old
- 160+ Institutions
- 45 Countries
- and 3+ more years to grow!
CHAMELEON HARDWARE

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

Commercial Clouds via CloudBank

Chameleon Associate Sites (Northwestern and others)

FABRIC and other partners

Chicago

Austin

Haswell
Standard Cloud Unit

SkyLake
Standard Cloud Unit

CascadeLake
Standard Cloud Unit

100Gbps uplink public network (each site)
CHAMELEON HARDWARE (DETAILS)

“Start with large-scale homogenous partition”

- 12 Haswell racks, each with 42 Dell R630 compute servers with dual-socket Intel Haswell processors (24 cores) & 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 racks (32 nodes each); Corsa (DP2400 & DP2200), 100Gb uplinks to core network
- CascadeLake rack (32 nodes), 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 (TACC) + 0.5 (UC) PB global storage, 100Gb Internet connection between sites

“Graft on heterogeneous features”

- Infiniband with SR-IOV support, High-mem, NVMe, SSDs, P100 GPUs (total of 22 nodes), RTX GPUs (40 nodes), FPGAs (4 nodes)
- ARM microservers (24) and Atom microservers (8), low-power Xeons (8)

Coming in Phase 3: upgrading Haswells to CascadeLake and IceLake + AMD, new GPUs and FPGAs, more and newer IB fabric, variety of storage options for disaggregated hardware experiments, composable hardware (LiQid), networking (P4, integration with FABRIC), IoT devices -- and strategic reserve
CHI 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
- Jupyter integration
- Networks: stitching and BYOC

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

Authentication via federated identity,
Interfaces via GUI, CLI and python/Jupyter
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
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

- 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
- Talk information at bit.ly/SC19PoDD

SC’19: “PoDD: Power-Capping Dependent Distributed Applications”
FEDERATED LEARNING

- Zheng Chai and Yue Cheng, George Mason University
- Research: federated learning
- Testbed requirements:
  - Bare metal, ability to record network traffic precisely
  - Support for large-scale and diverse hardware
  - Powerful nodes with large memory

GIVING CHAMELEON AN EDGE

- What does an edge testbed look like?
  - A lot like a cloud: all the features you know and love – but via containers
  - Not like a cloud at all: location, location, location (...and network to that location!) -- cameras, actuators, software defined radios (SDRs), etc.
  - CHI@Edge: mixed-ownership devices managed via an SDK by a virtual site
  - Practice makes perfect: listen to users and adjust

- How to build an edge testbed quickly
  - Familiar challenges: access management, secure network connections, resource management, and other sharing considerations
  - New challenges: remote locations, power/networking constraints, peripheral devices
  - Leverage existing investment in (1) open source (OpenStack), and (2) Chameleon
BUILDING CHI@EDGE

From this... …to this!

www.chameleoncloud.org
CHI@EDGE EXPERIMENTAL WORKFLOW (PREVIEW)

- discover resources
  - Complete
  - Up-to-date

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

- configure and interact
  - Container
  - Catalog of images
  - Snapshotting
  - Jupyter integration for orchestration

- monitor

Authentication via federated identity,
Interfaces via GUI, CLI and python/Jupyter
CHI AND CHI@EDGE SIDE BY SIDE

Chameleon for bare metal

- Advanced reservations for **bare metal machines**
- **Bare metal reconfigurability**
- Single-tenant isolation
- Heterogeneous collection of interesting hardware
- Isolated networking, public IP capability, **OpenFlow SDN**
- Composable cloud APIs (GUI, CLI, Python+Jupyter)
- Owned and operated by Chameleon

Chameleon for edge

- Advanced reservations for **IoT/edge devices**
- **Container deployment**
- Single-tenant isolation
- Heterogeneous collection of interesting hardware and **peripherals/locations**!
- Isolated networking, public IP capability
- Composable cloud APIs (GUI, CLI, Python+Jupyter)
- Mixed ownership model: bring your own device(s)!

[www.chameleoncloud.org]
JOIN US FOR THE SUMMER OF CHAMELEON!

- June 2021: CHI@Edge releases, shared hardware (nvidia nanos and raspberry pis), community webinars
- July 2021: “bring your own device” with attestations/SLAs, peripherals, support for limited sharing
- To use: https://www.chameleoncloud.org/experiment/chiedge/
- To learn: https://www.youtube.com/user/ChameleonCloud/videos
- Chameleon-edge-users mailing list: https://groups.google.com/g/chameleon-edge-users?pli=1
- Help us build a better testbed!
PRACTICAL REPRODUCIBILITY

- Can experiments be as sharable as papers are today?
- Reproducibility baseline: sharing hardware via instruments held in common
- Clouds: sharing experimental environments
  - Disk images, orchestration templates, and other artifacts
- What is missing?
  - Telling the whole story: hardware + experimental container + experiment workflow + data analysis + story – literate programming
  - The easy button: it has to be easy to package, easy to repeat, easy to find, easy to get credit for, easy to reference, etc.
  - Nits and optimizations: declarative versus imperative, transactional versus transparent

*Paper: “The Silver Lining”, IEEE Internet Computing 2020*
EXPERIMENT SHARING IN CHAMELEON

- Hardware and hardware versions
  - >105 versions over 5 years
  - Expressive allocation

- Images and orchestration
  - >130,000 images, >35,000 orchestration templates and counting

- Packaging and repeating: integration with JupyterLab

- Share, find, publish and cite: Trovi and Zenodo
PACKAGING SHARABLE EXPERIMENTS

Literate Programming with Jupyter

Experimental storytelling:
ideas/text, process/code, results

Complex Experimental containers

- Repeatability by default: Jupyter notebooks + Chameleon experimental containers
  - JupyterLab for our users: use jupyter.chameleoncloud.org with Chameleon credentials
  - Interface to the testbed in Python/bash + examples (see LCN’18: https://vimeo.com/297210055)
  - Especially for highly distributed experiments (CHI@Edge) notebook as terminal multiplexer

*Paper: “A Case for Integrating Experimental Containers with Notebooks”, CloudCom 2019*
TROVI: CHAMELEON'S EXPERIMENT PORTAL

Create a new packaged experiment out of any directory of files in your Jupyter server. It is private to you unless shared. Supports sharing similar to Google Drive.

Any user with a Chameleon allocation can find and "replay" the packaged experiment.
PUBLISHING EXPERIMENTS

- Digital publishing with Zenodo: make your experimental artifacts citable via Digital Object Identifiers (DOIs)

- Integration with Zenodo
  - Export: make your research citable and discoverable
  - Import: access a wealth of digital research artifacts already published
PARTING THOUGHTS

- Scientific instruments: laying down the pavement as science walks on it
- Chameleons like to change:
  - Experimental environments that can adapt to your experiment
  - Testbed that adapts itself to your scientific needs -- from cloud to edge: CHI@Edge
- Chameleon is a shareable research instrument -- but it is also a sharing platform
  - The easy button: making reproducibility sustainable will rely on creating “research marketplace”: sharing experiments as naturally as we share papers now
  - Clouds help us package experimental environments almost as a side-effect
  - Literate programming is a convenient vehicle for “closing the gap”: packaging the whole experiment so that it can be reproduced easily
We’re here to change – come and change with us!

www.chameleonclickcloud.org