www. chameleoncloud.org

CHAMELEON: CHANGING THE WAY WE SHARE

Kate Keahey

Mathematics and CS Division, Argonne National Laboratory

CASE, University of Chicago

keahey@anl.gov

January 22, 2021

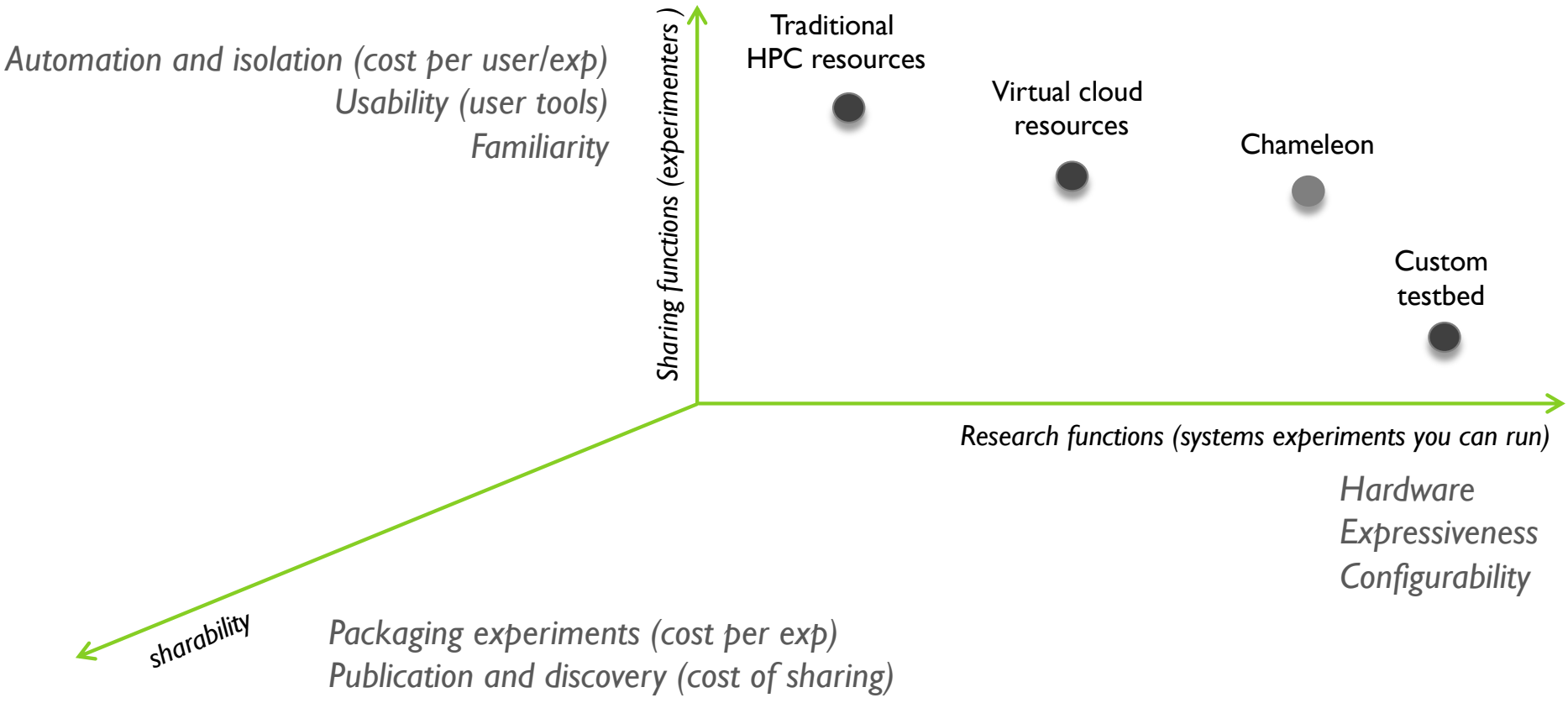
Department of Computer Science Seminar Series, Florida International University

THE UNIVERSITY OF CHICAGO TACC renci NORTHWESTERN UNIVERSITY NSF

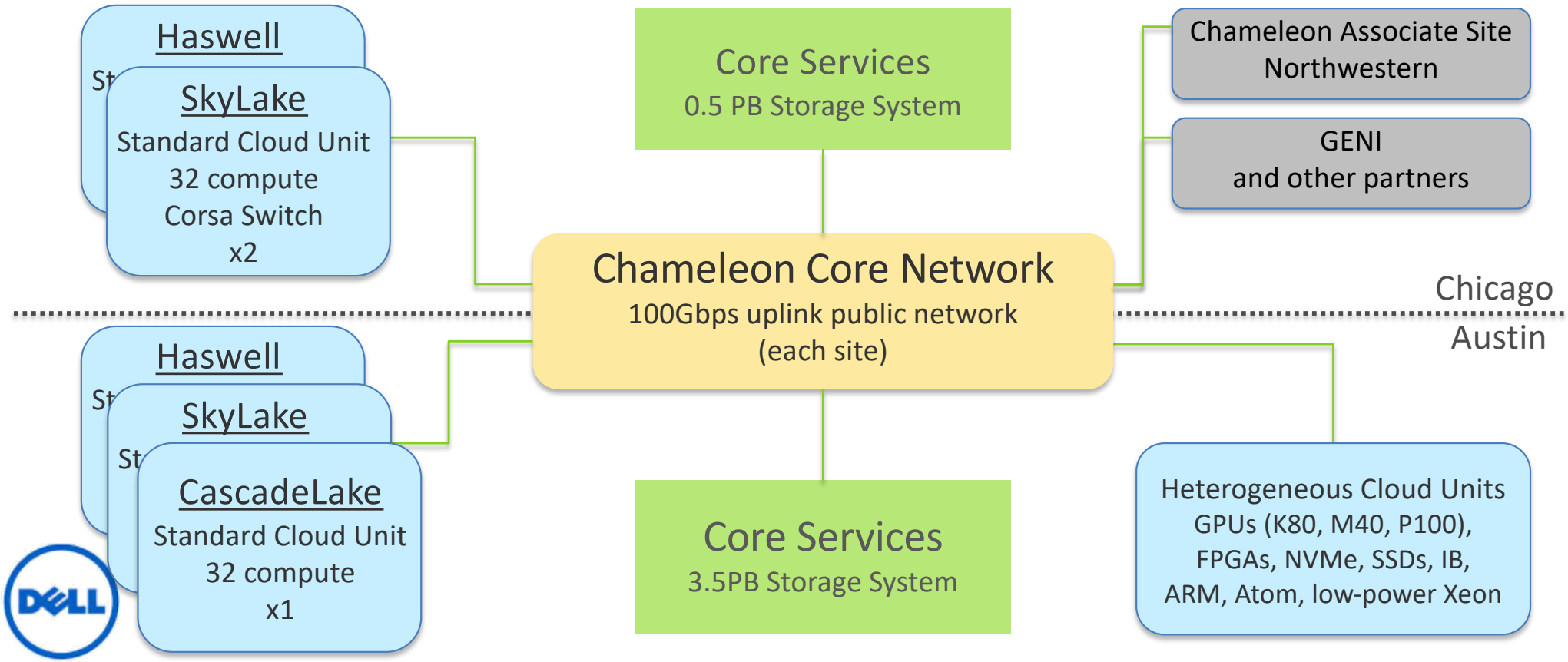
CHAMELEON IN A NUTSHELL

- We like to change: a testbed that adapts itself to your experimental needs
  - Deep reconfigurability (bare metal) and isolation
  - 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++: leveraging mainstream cloud technologies
  - 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 just now!
  - Currently 4,000+ users, 600+ projects, 100+ institutions, 300+ publications

THE MOST CS EXPERIMENTS FOR THE MOST USERS



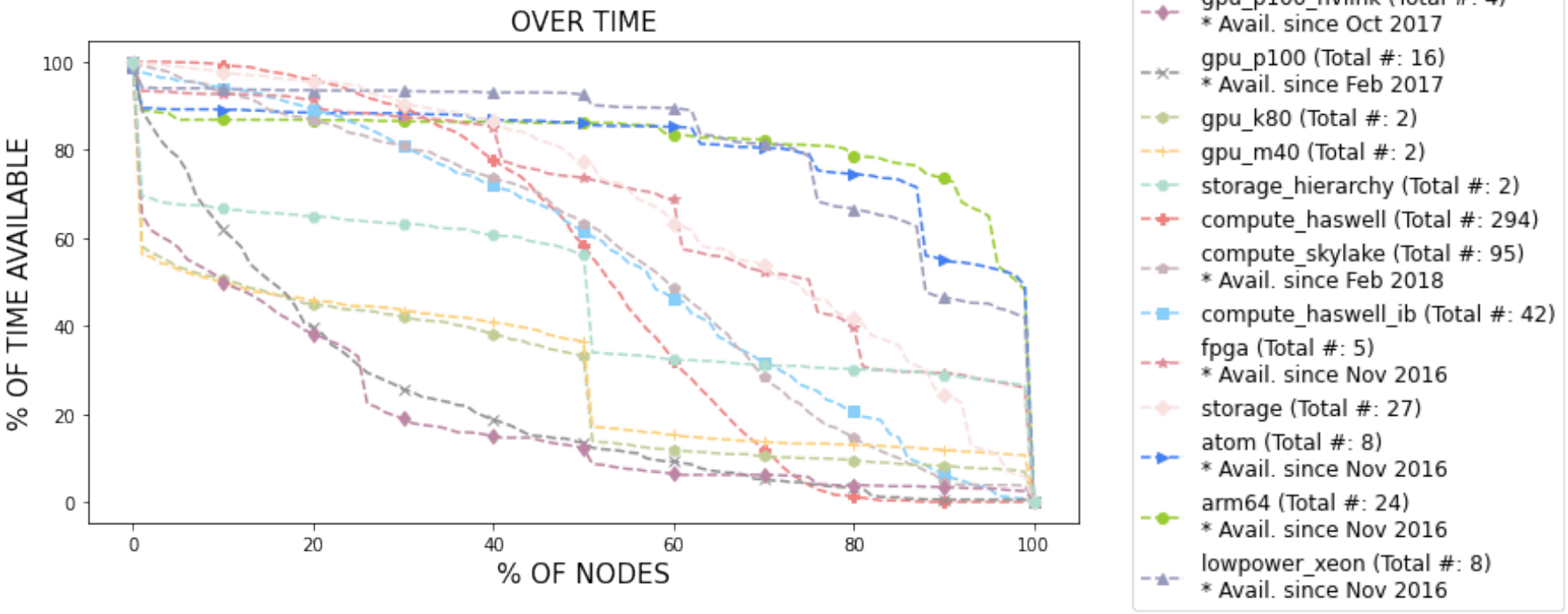
CHAMELEON HARDWARE



CHAMELEON HARDWARE (DETAILS)

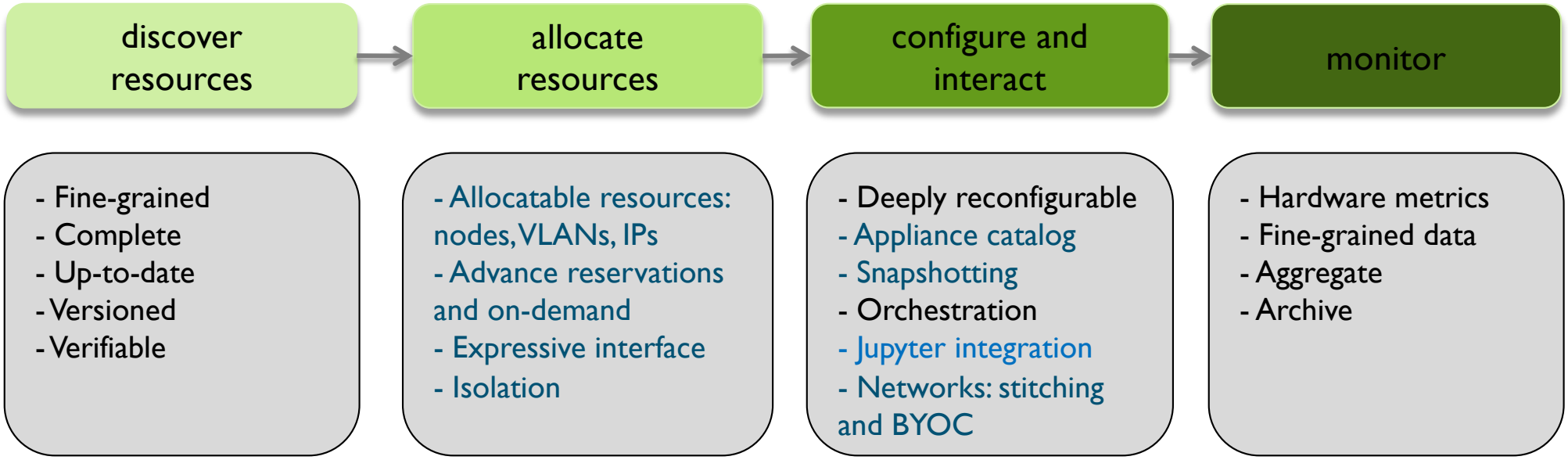
- "Start with large-scale homogenous partition"
  - 12 Haswell Standard Cloud Units (48 node racks), each with 42 Dell R630 compute servers with dual-socket Intel Haswell processors (24 cores) and 128GB RAM and 4 Dell FX2 storage servers with 16 2TB drives each; Force10 s6000 OpenFlow-enabled switches 10Gb to hosts, 40Gb uplinks to Chameleon core network
  - 3 SkyLake Standard Cloud Units (32 node racks); Corsa (DP2400 & DP2200) switches, 100Gb ulpinks to Chameleon core network
  - CascadeLake Standard Cloud Units (32 node rack) , 100Gb ulpinks 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, P100 GPUs (total of 22 nodes), RTX GPUs (40 nodes), FPGAs (4 nodes)
  - ARM microservers (24) and Atom microservers (8), low-power Xeons (8)

HARDWARE USAGE



Paper: "Lessons Learned from the Chameleon Testbed", USENIX ATC 2020

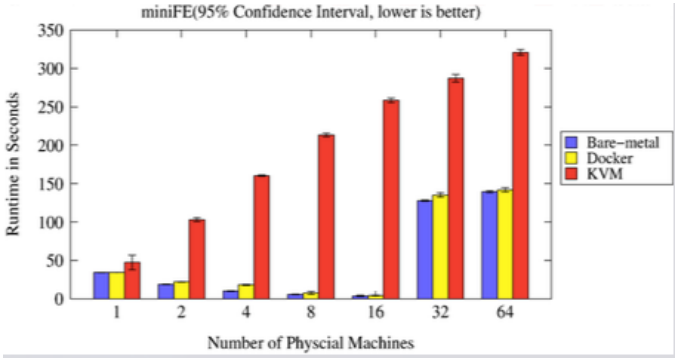
EXPERIMENTAL WORKFLOW

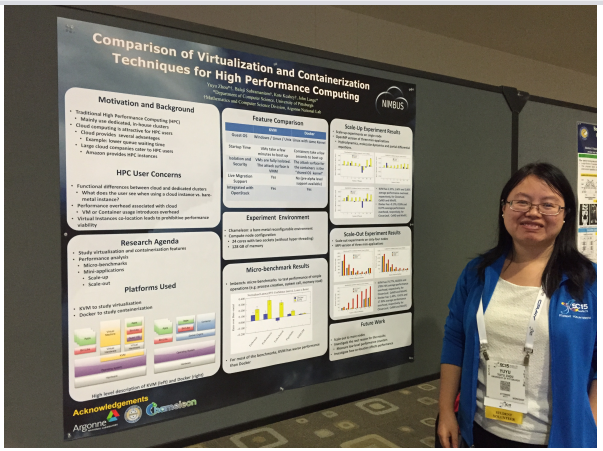


CHI = 65%*OpenStack + 10%*G5K + 25%*"special sauce"

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



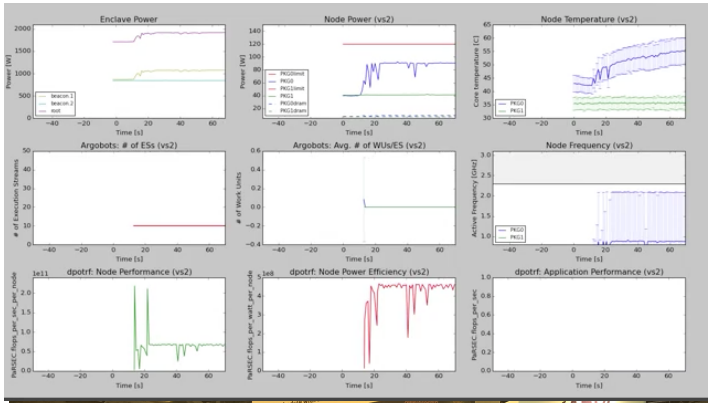


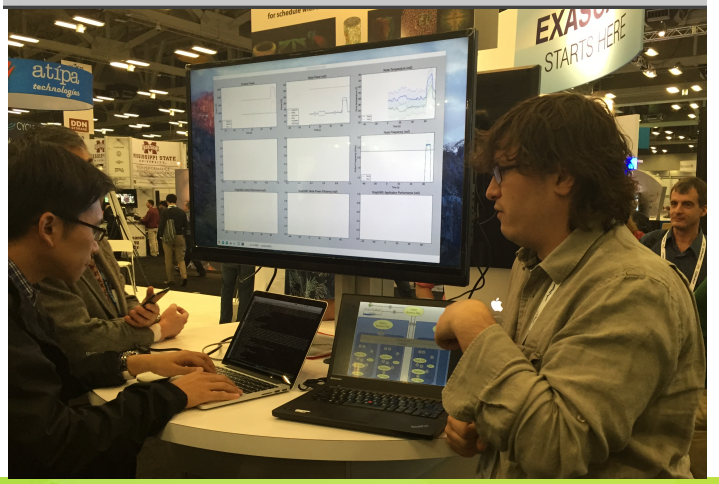
SC15 Poster: “Comparison of Virtualization and Containerization Techniques for HPC”

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"



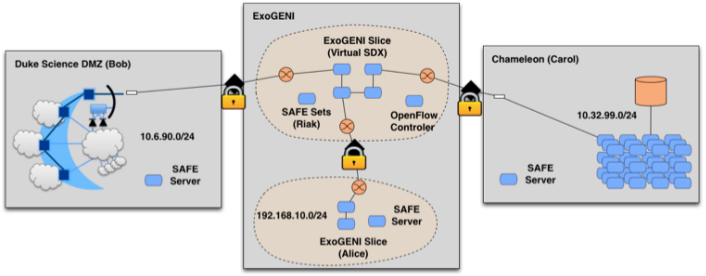


CLASSIFYING CYBERSECURITY ATTACKS

- 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

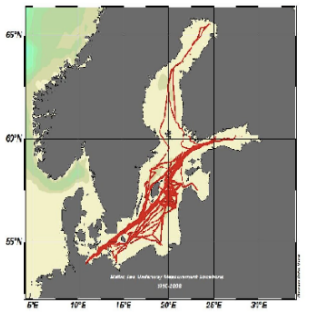
CREATING DYNAMIC SUPERFACILITIES

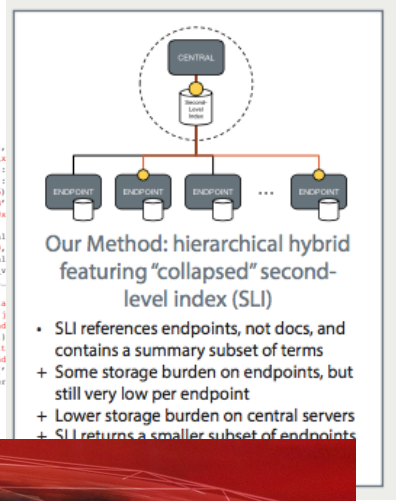
- NSF CICI SAFE, Paul Ruth, RENCI-UNC Chapel Hill
- Creating trusted facilities
  - Automating trusted facility creation
  - Virtual Software Defined Exchange (SDX)
  - Secure Authorization for Federated Environments (SAFE)
- Testbed requirements
  - Creation of dynamic VLANs and wide-area circuits
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



DATA SCIENCE RESEARCH

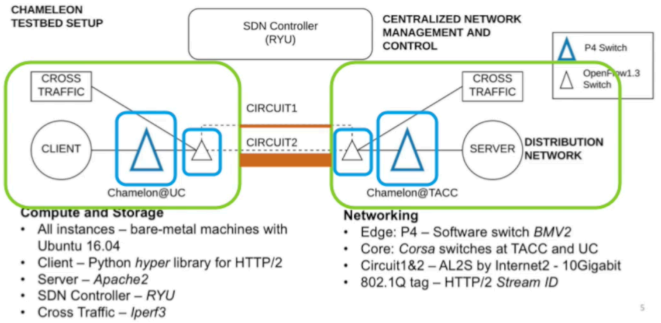
- ACM Student Research Competition semi-finalists:
  - Blue Keleher, University of Maryland
  - Emily Herron, Mercer University
- Searching and image extraction in research repositories
- Testbed requirements:
  - Access to distributed storage in various configurations
  - State of the art GPUs
  - Easy 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



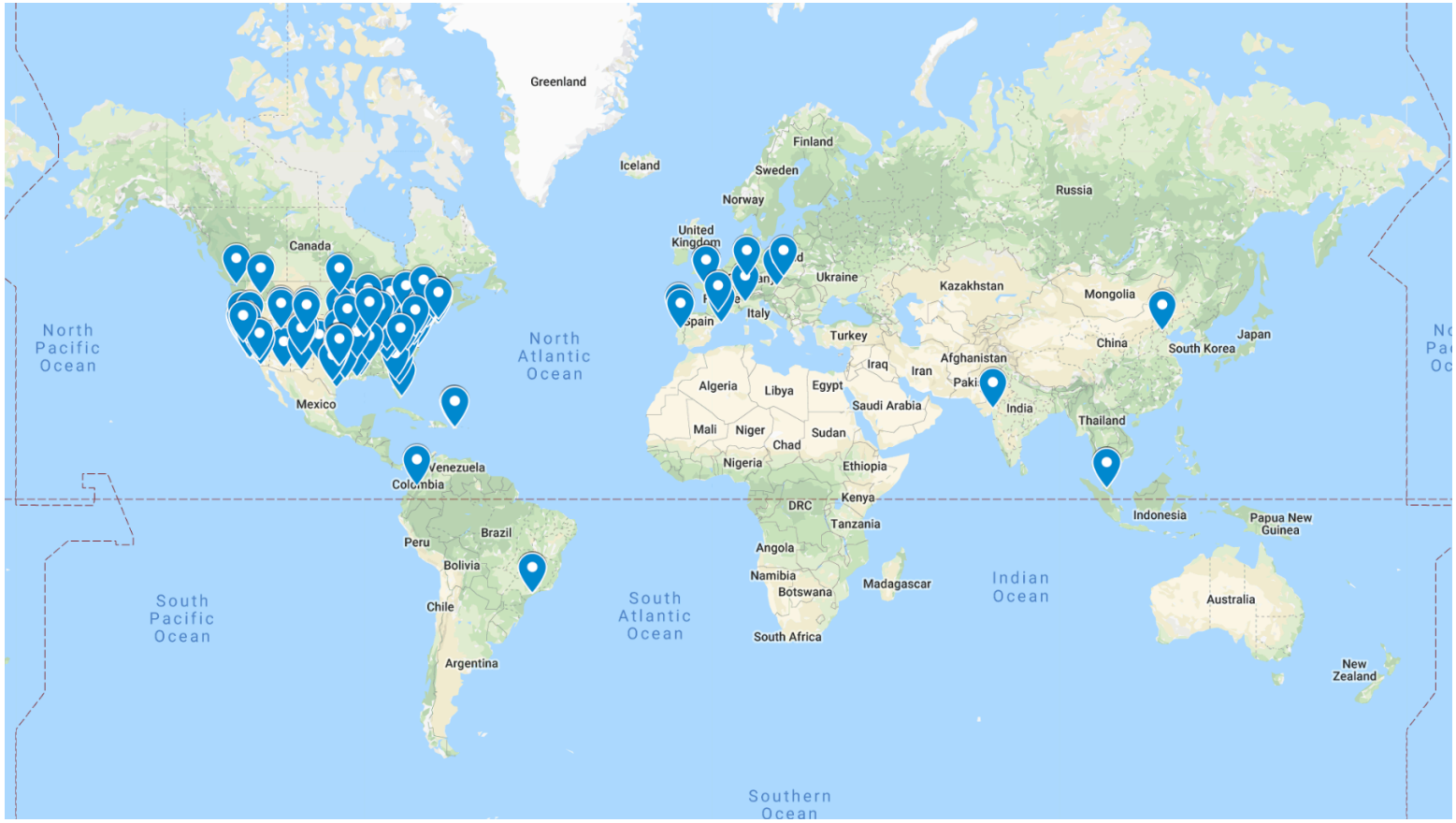
LCN'18: "Application-based QoS support with P4 and OpenFlow"

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"

AN OPEN PLATFORM



LESSER COST, MORE EXPERIMENTERS

- Working with mainstream open source project (OpenStack)
  - Familiar interfaces and tranferable skills: 858 deployments, 441 organizations, 63 countries
  - Working with large community (~8,400 total contributors, ~6,000 reviewing code)
  - Access to existing documentation and support systems
  - New features: whole disk image boot, support for non x86, multi-tenant networking
  - Opportunity to contribute (though at a cost): Blazar as OpenStack component
  - From the “Mother of All Upgrades” (~7 months) to manageable investment (~1 month)
- Support and reliability
  - Monitoring and alerting: smoke tests, live monitoring with coverage, centralized logging
  - Remediation: runbooks and hammers (automated repair)
  - Create a process around maintenance (automated scripts ensure uniformity)
- Average of ~13 help desk tickets per week

[Runbook] IronicNodeInErrorState

Jason Anderson edited this page yesterday · 2 revisions

Build of **neutron** (train) failed. View build log

Build of **neutron** (rocky) completed successfuly. View build log

15:41 **chameleon-ci** APP

Deployment of **neutron** (ansible-uc-dev) starting. View job

1 reply 20 hours ago

15:58 **GitHub** APP

diurnalist

1 new commit **pushed to** `master`

`44aa3f09` - Ensure latest version of Kolla checked out

ChameleonCloud/service-containers

wever, they
ize the

de cannot be
as a
does
ning network.

`json | jq`

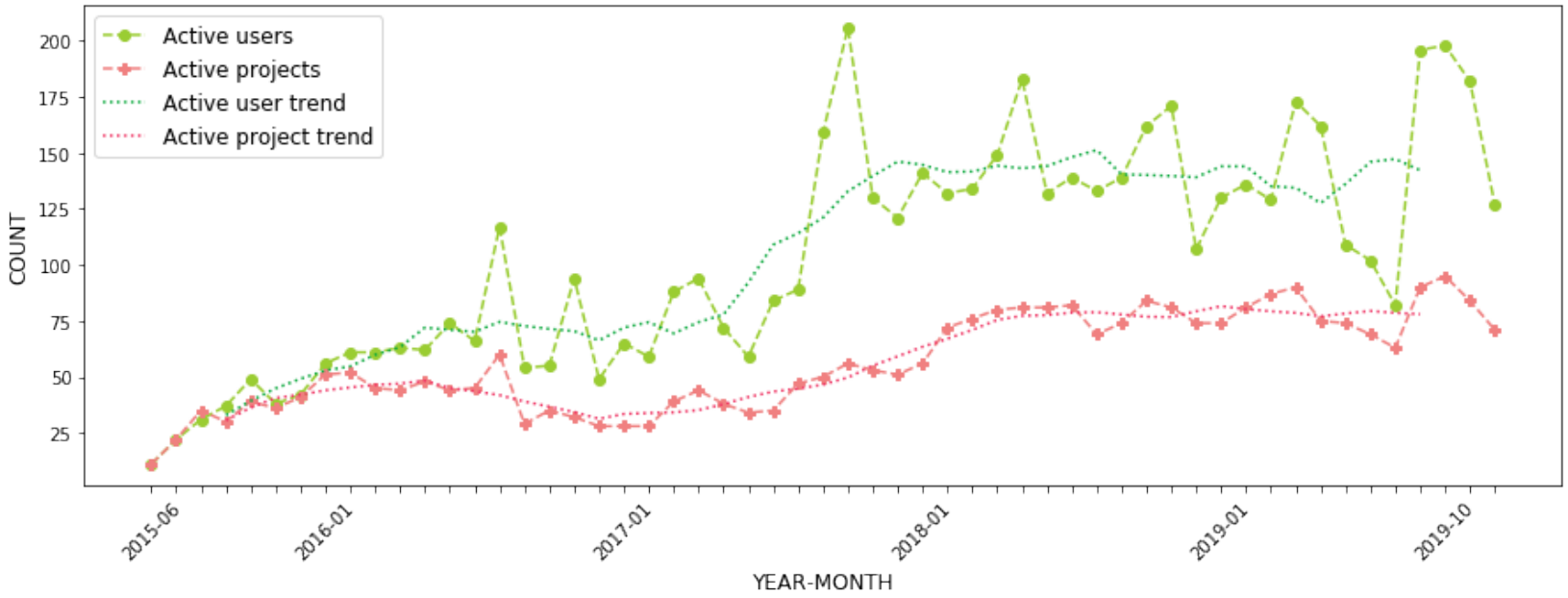
`.extra`. A node that has been reset by the hammer will have a "hammer_error_resets" key with timestamps for each time a reset was performed.

2. If there are more than `max_attempts` (3 at time of writing), then this node could have an issue with its IPMI interface and should be put into maintenance.

CHI-IN-A-BOX: CHAMELEON NEAR YOU!

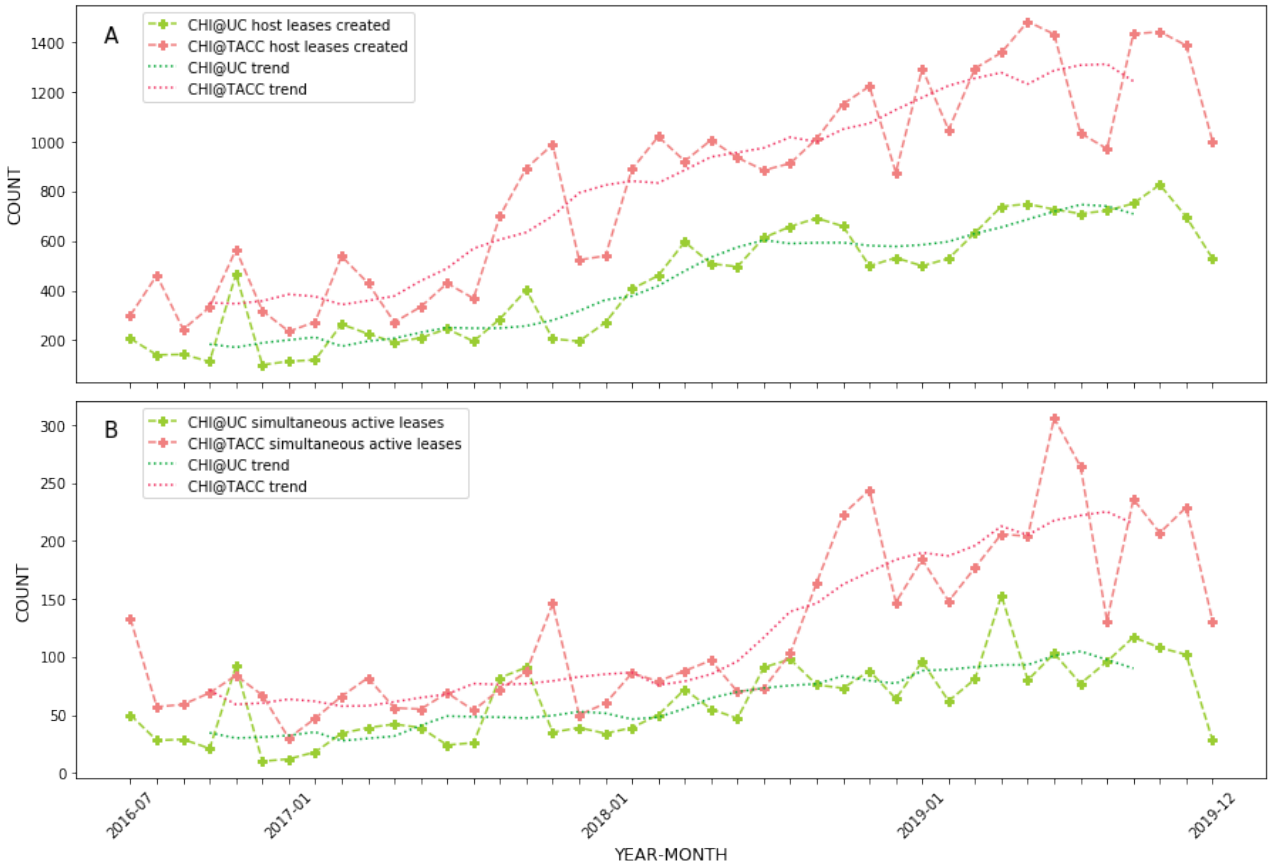
- CHI-in-a-box: packaging a commodity-based testbed
  - First released in summer 2018, continuously improving
  - Packaging systems as well as operations model
- CHI-in-a-box scenarios
  - Independent testbed: package assumes independent account/project management, portal, and support
  - Chameleon extension: 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 features!
  - Two organizations working on independent testbed configuration

EXPERIMENTERS: ACTIVE USERS



Paper: "Lessons Learned from the Chameleon Testbed", USENIX ATC 2020

EXPERIMENTERS: ACTIVE LEASES



TOWARDS SHARING EXPERIMENTS

- Towards a world where experiments are as sharable as papers today
- Instruments held in common: sharing hardware
- 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

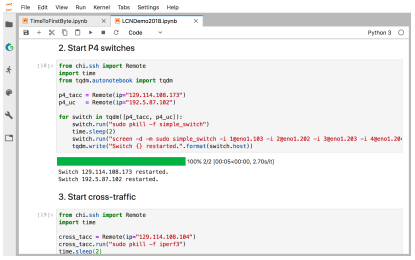
Paper: "The Silver Lining", IEEE Internet Computing 2020

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

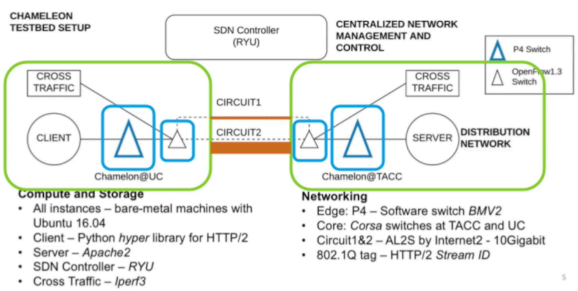
PACKAGING SHARABLE EXPERIMENTS

Literate Programming with Jupyter



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

+



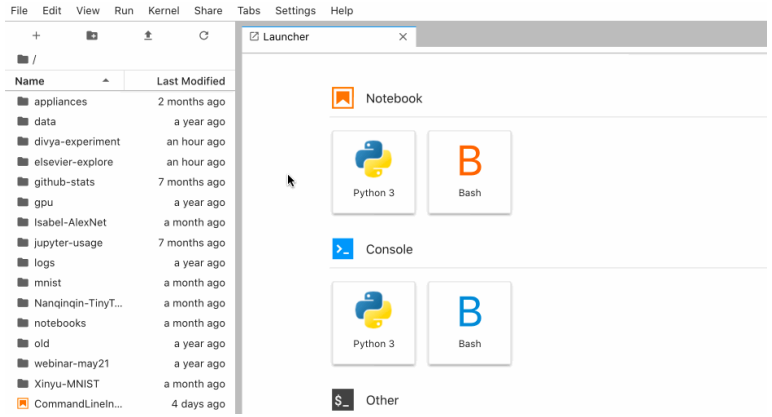
Complex Experimental containers

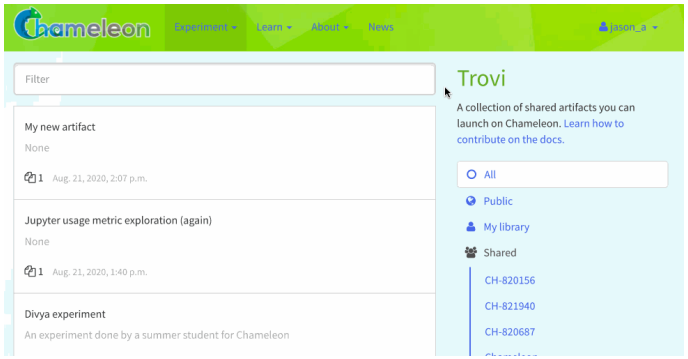
- Repeatability by default: Jupyter notebooks + Chameleon experimental containers
  - JupyterLab for our users: use jupyter.chameleoncloud.org with Chameleon credentials
  - Interface to the testbed in Python/bash + examples (see LCN'18: https://vimeo.com/297210055)
  - Named containers: your experimental process goes here

Paper: "A Case for Integrating Experimental Containers with Notebooks", CloudCom 2019

TROVI: SHARING ON THE FLY

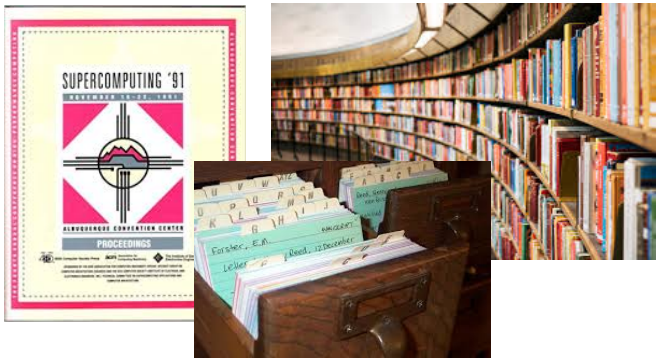
- Best Practice Highlights
  - Explain preconditions, manage expectations, define repeat condition, make workflow code cells idempotent, avoid noise
- Sharing
  - Create a bundle
  - Edit sharing settings
  - Browse experiment
- Making research findable





PUBLISHING

Familiar research sharing ecosystem



Digital research sharing ecosystem

Chameleon jupyterhub

?

- 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
- Towards making research findable: the digital sharing platform

zenodo

PHASE 3 ADVERTISEMENT

- New hardware: traditional servers, new GPUs and FPGAs, storage upgrades (FLASH arrays), composable hardware (LiQid), networking (P4, integration with FABRIC), IoT devices -- and strategic reserve
- New capabilities: federation, Bring Your Own Device (BYOD) & CHI@Edge, networking (allocatable switches with P4), core capability improvements
- Infrastructure: CHI-in-a-Box, integration with production infrastructures
- Research sharing: better methods of experiment packaging, publishing, and discovery, digital experiment libraries, engagement
- Engagement with research and education

A FEW WORDS OF SUMMARY

- 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 as a side-effect of using them and serve as ”player” for such experiment
- Literate programming is a convenient vehicle for “closing the gap”: packaging the whole experiment
- We now need a critical mass of content
- Come, help us change: chameleoncloud.org

We're here to change – come and change with us!

www.chameleoncloud.org