www. chameleoncloud.org

CHAMELEON: FROM CLOUD TO EDGE

Kate Keahey

Mathematics and CS Division, Argonne National Laboratory

CASE, University of Chicago

keahey@anl.gov

June 21, 2021
ScienceCloud virtual workshop

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++: 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

45
Countries

160+
Institutions

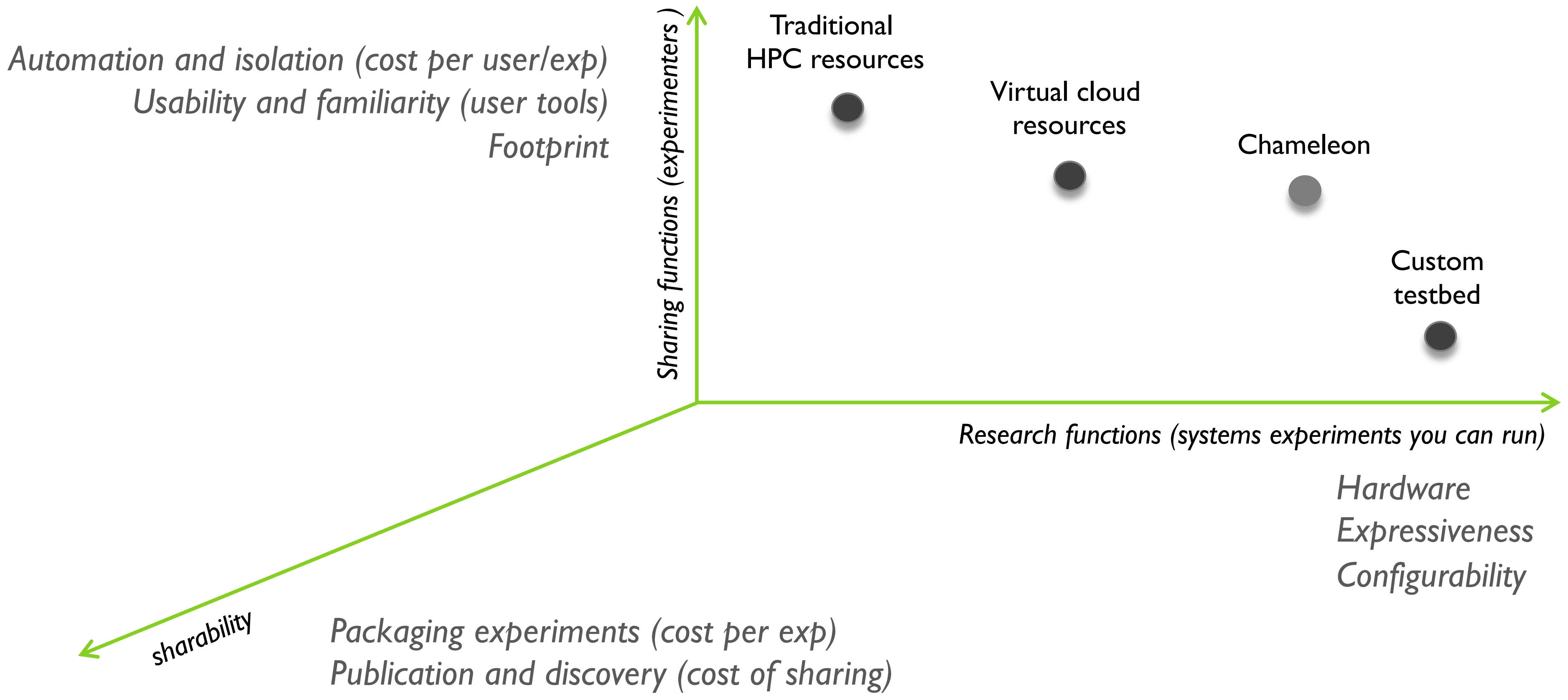
Over
5,500
Users

700+
Projects

6+
Years Old

and 3+ more years to grow!

THE MOST CS EXPERIMENTS FOR THE MOST USERS



Automation and isolation (cost per user/exp)
Usability and familiarity (user tools)
Footprint

Sharing functions (experimenters)

Traditional HPC resources

Virtual cloud resources

Chameleon

Custom testbed

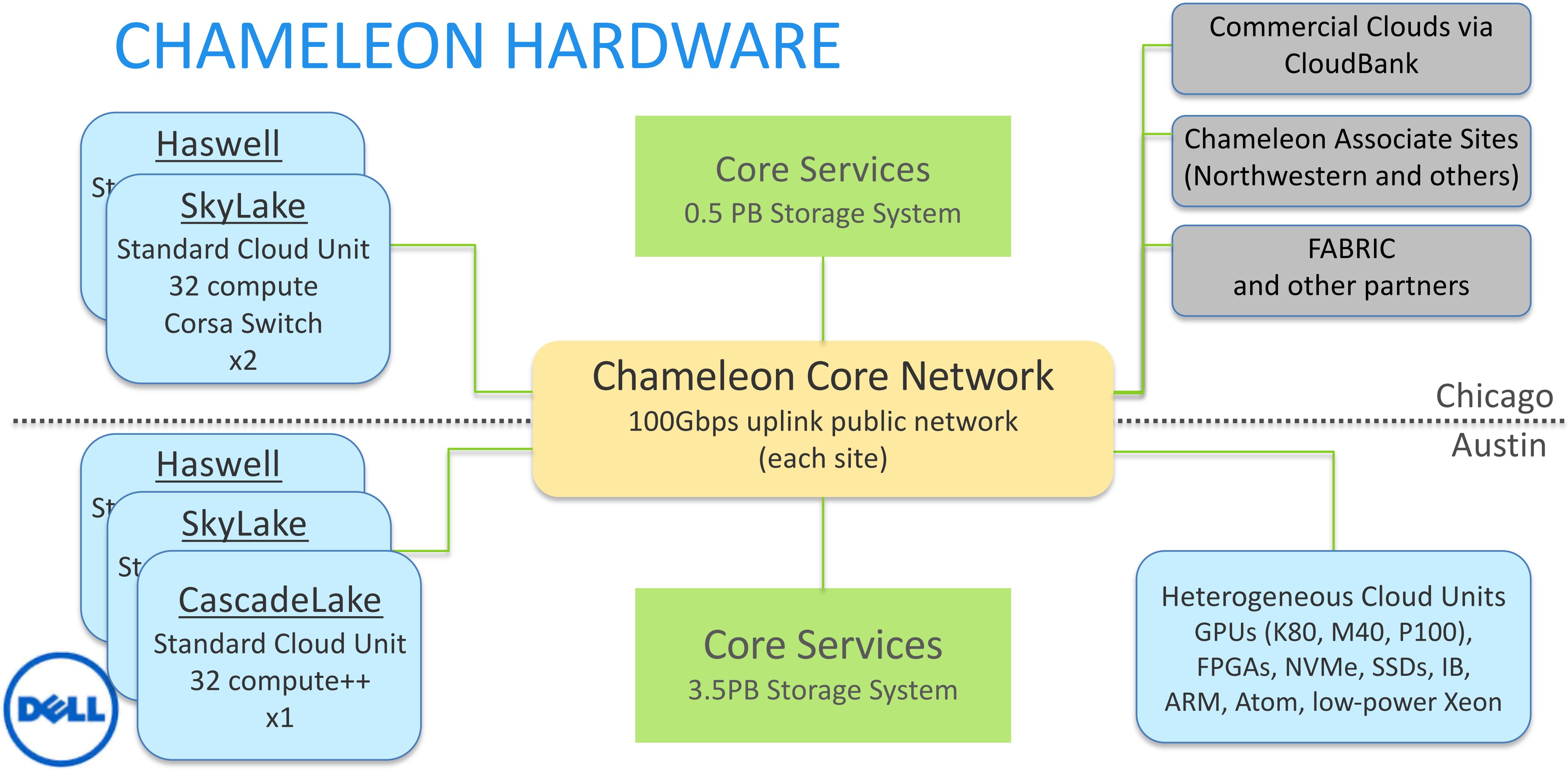
Research functions (systems experiments you can run)

Hardware
Expressiveness
Configurability

sharability

Packaging experiments (cost per exp)
Publication and discovery (cost of sharing)

CHAMELEON HARDWARE



Haswell

St

SkyLake

Standard Cloud Unit
32 compute
Corsa Switch
x2

Core Services
0.5 PB Storage System

Commercial Clouds via CloudBank

Chameleon Associate Sites (Northwestern and others)

FABRIC
and other partners

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

Chicago

Austin

Haswell

St

SkyLake

St

CascadeLake

Standard Cloud Unit
32 compute++
x1

DELL

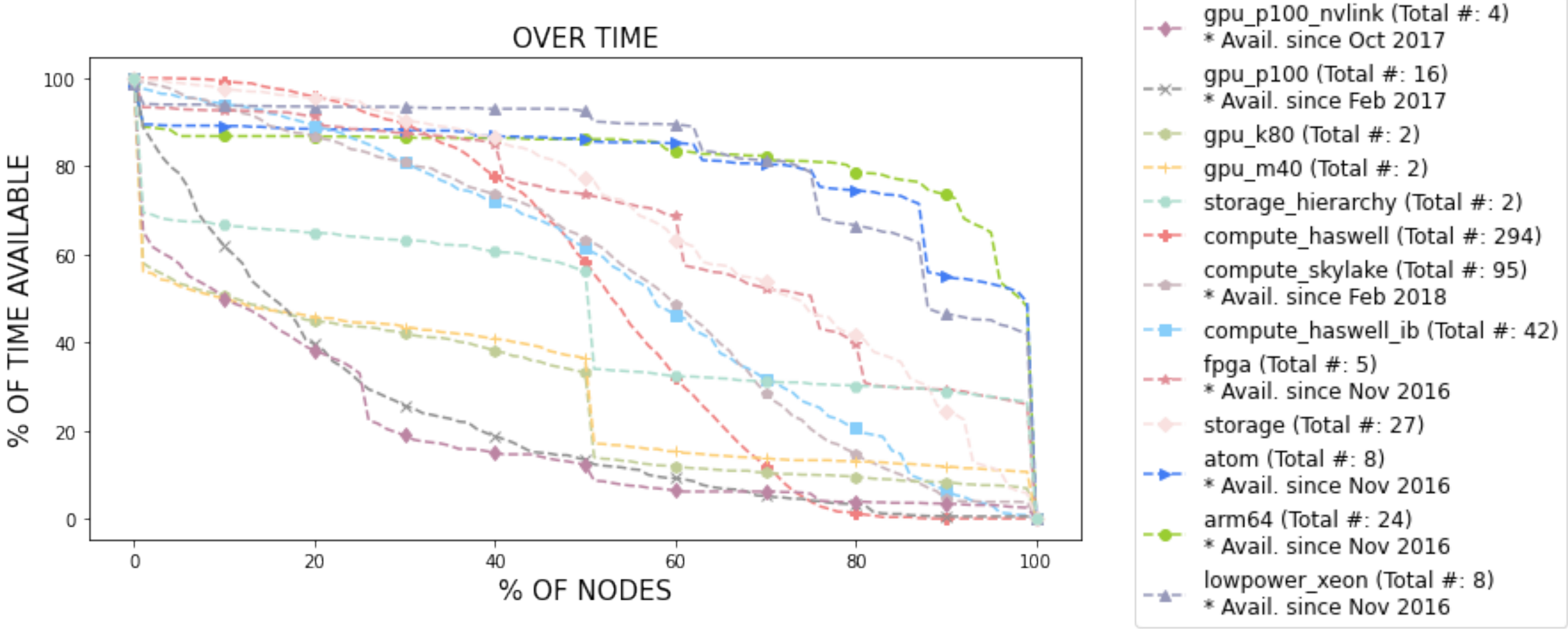
Core Services
3.5PB Storage System

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

CHAMELEON HARDWARE (DETAILS)

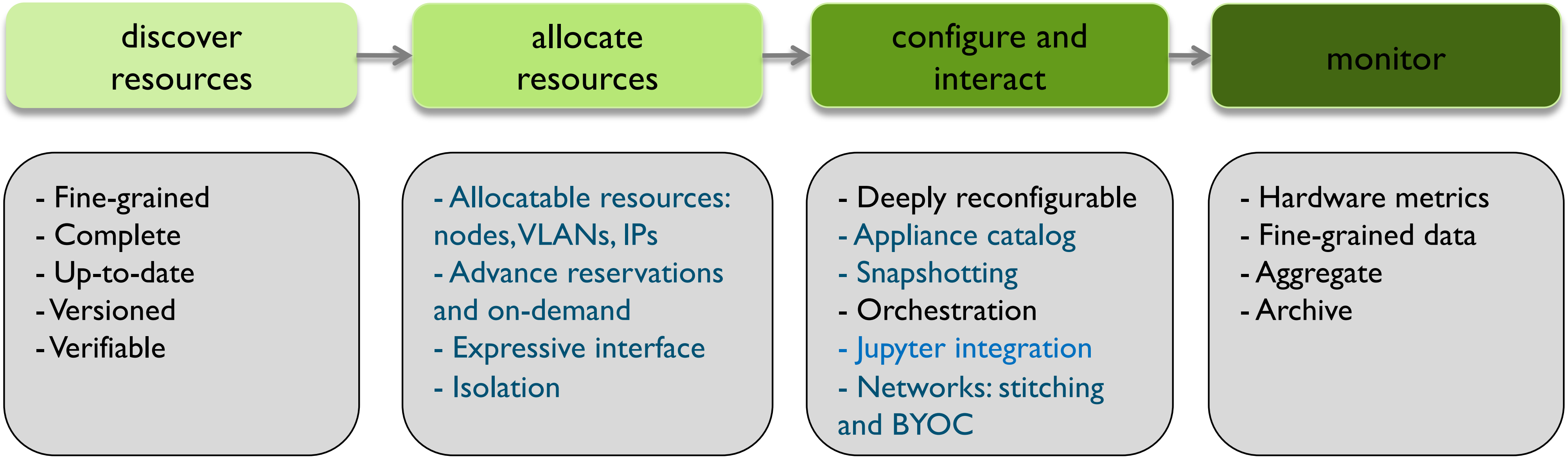
- "Start with large-scale homogenous partition"
  - 12 Haswell racks, each with 42 Dell R630 compute servers with dual-socket Intel Haswell processors (24 cores) & 128GB RAM and 4 Dell FX2 storage servers with 16 2TB drives each; Force10 s6000 OpenFlow-enabled switches 10Gb to hosts, 40Gb uplinks to Chameleon core network
  - 3 SkyLake racks (32 nodes each); Corsa (DP2400 & DP2200), 100Gb uplinks to core network
  - CascadeLake rack (32 nodes), 100Gb 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 (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

HARDWARE USAGE



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

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

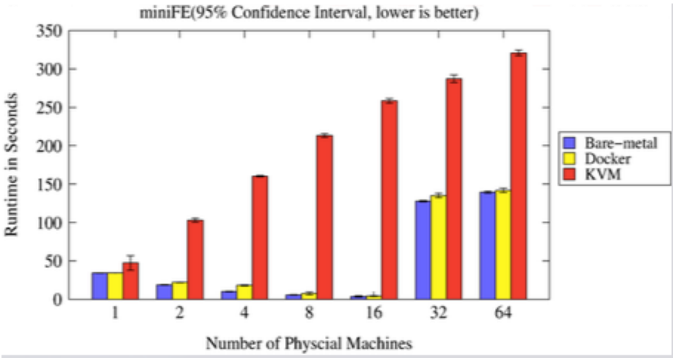
monitor

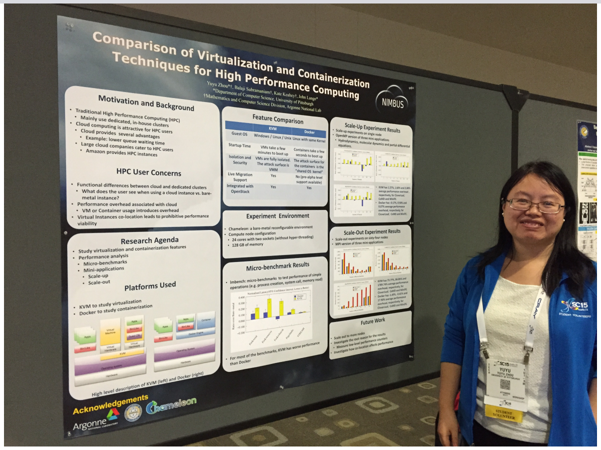
- Hardware metrics
- Fine-grained data
- Aggregate
- Archive

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



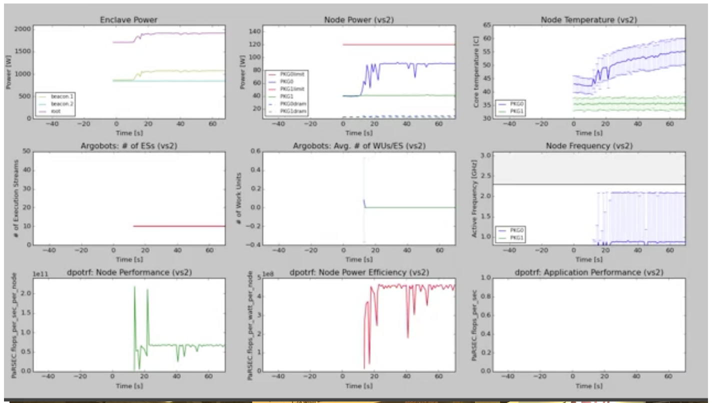

Comparison of Virtualization and Containerization Techniques for High Performance Computing

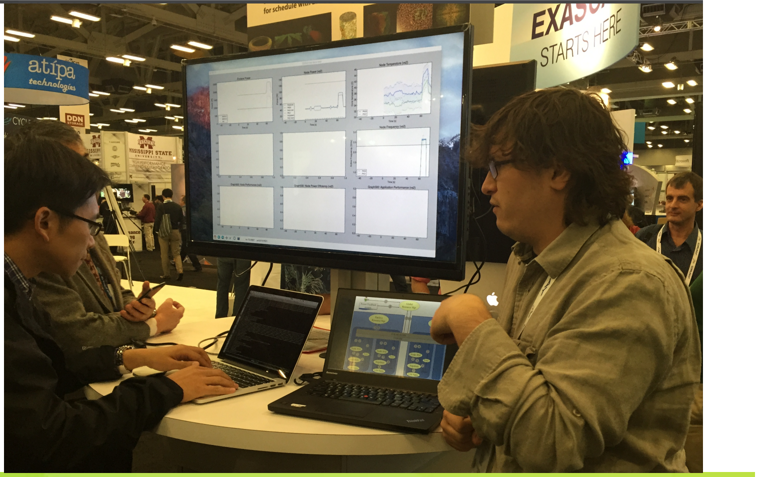
SC15 Poster: "Comparison of Virtualization and Containerization Techniques for HPC"

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"



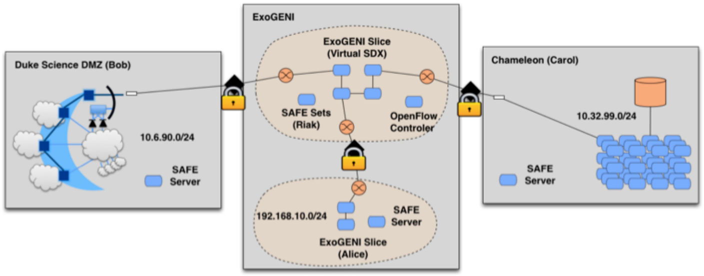


CLASSIFYING CYBERSECURITY ATTACKS

- 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

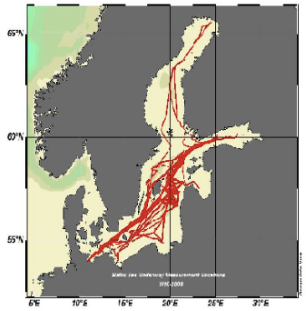
CREATING DYNAMIC SUPERFACILITIES

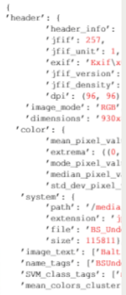
- NSF CICI SAFE, Paul Ruth, RENCI-UNC Chapel Hill
- Creating trusted facilities
  - Automating trusted facility creation
  - Virtual Software Defined Exchange (SDX)
  - Secure Authorization for Federated Environments (SAFE)
- Testbed requirements
  - Creation of dynamic VLANs and wide-area circuits
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



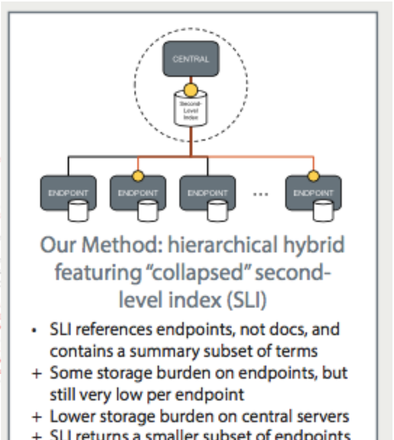
DATA SCIENCE RESEARCH

- 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





```
{
'header': {
        'header_info':
        'jfif': 257,
        'jfif_unit': 1,
        'exif': 'Exif\x
        'jfif_version':
        'jfif_density':
        'dpi': (96, 96)
    'image_mode': 'RGB'
    'dimensions': '930x
  'color': {
        'mean_pixel_val
        'extrema': ((0,
        'mode_pixel_val
        'median_pixel_v
        'std_dev_pixel_
   'system': {
        'path': '/media
        'extension': 'j
        'file': 'BS_Und
        'size': 115811}
   'image_text': ['Balt
   'name_tags': ['BSUnd
   'SVM_class_tags': ['
   'mean_colors_cluster
```



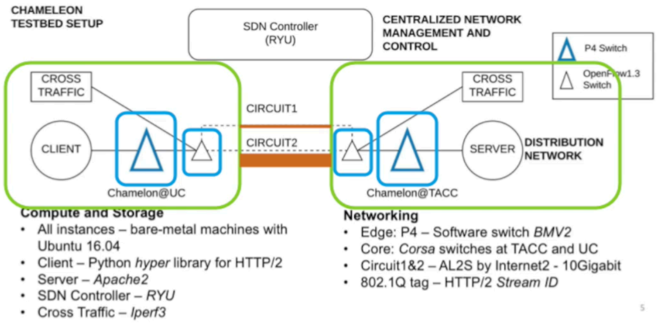
Our Method: hierarchical hybrid featuring "collapsed" second-level index (SLI)

- SLI references endpoints, not docs, and contains a summary subset of terms
- + Some storage burden on endpoints, but still very low per endpoint
- + Lower storage burden on central servers
- + SLI returns a smaller subset of endpoints

Chameleon 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



Compute and Storage

- All instances – bare-metal machines with Ubuntu 16.04
- Client – Python *hyper* library for HTTP/2
- Server – *Apache2*
- SDN Controller – *RYU*
- Cross Traffic – *Iperf3*

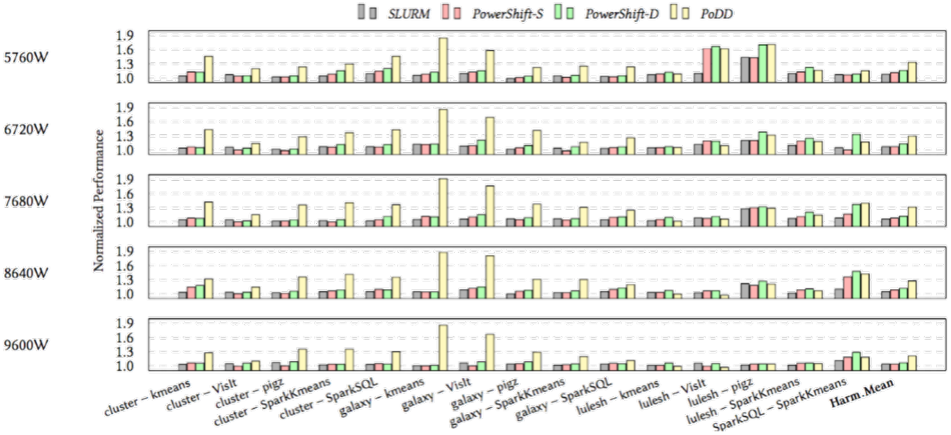
Networking

- Edge: P4 – Software switch *BMV2*
- Core: *Corsa* switches at TACC and UC
- Circuit1&2 – AL2S by Internet2 - 10Gigabit
- 802.1Q tag – HTTP/2 *Stream ID*

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

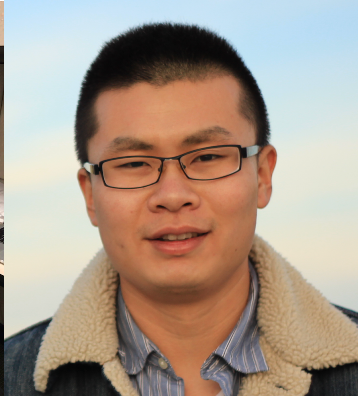
POWER CAPPING

- 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

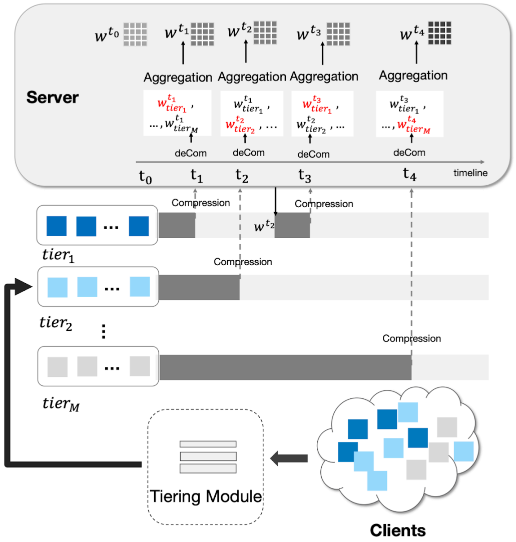


SC'19: "PoDD: Power-Capping Dependent Distributed Applications"

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



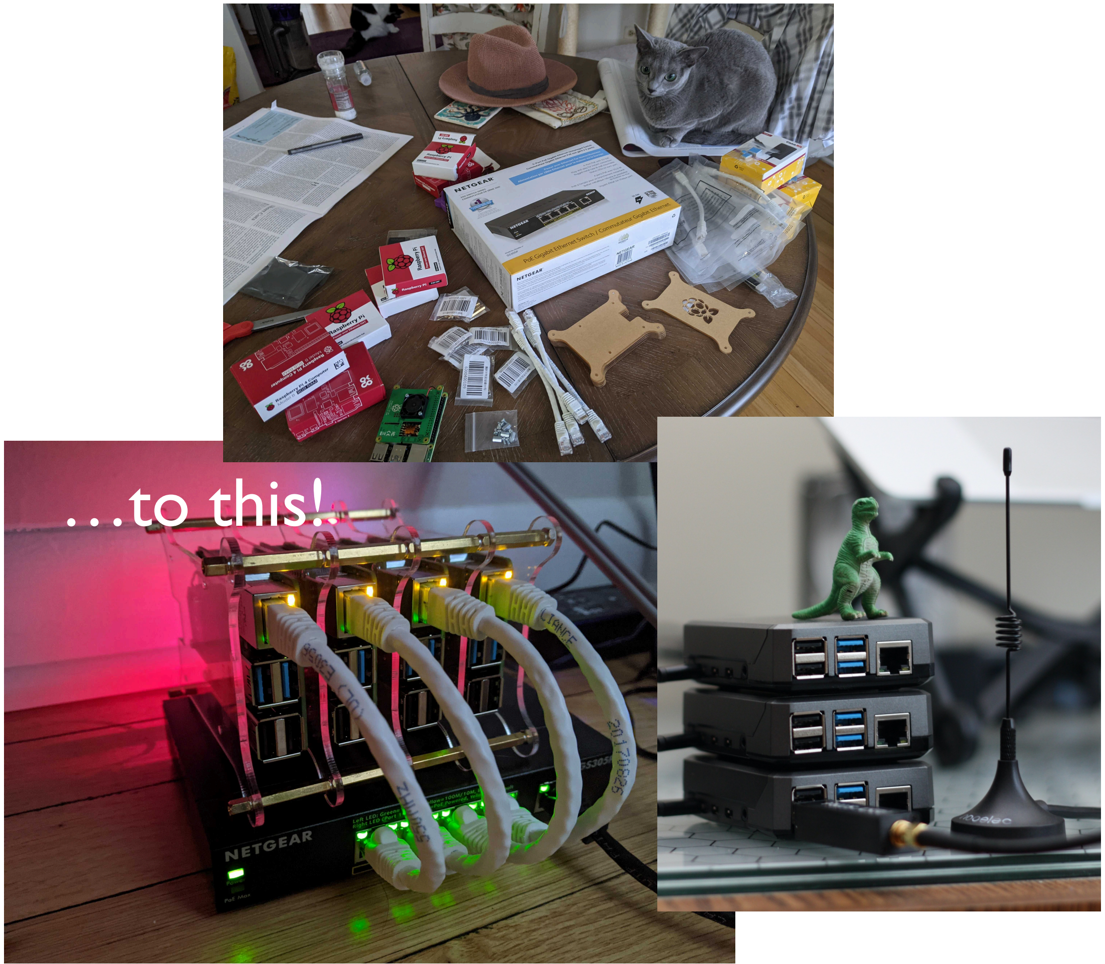
Paper: "FedAT:A Communication-Efficient Federated Learning Method with Asynchronous Tiers under Non-IID Data", October 2020

GIVING CHAMELEON AN EDGE

- What does an edge testbed look like?
  - Isolation: bare metal reconfiguration / virtual machines / **containers**
  - Sharing is caring: bring your own device (BYOD) model based on **CHI@Edge virtual site and SDK**
  - Practice makes perfect: **listen to users across a variety of experiments** and adjust
- How to build a testbed quickly
  - Leverage existing investment in (1) open source, and (2) Chameleon
  - Can we extend a cloud into the edge?
    - Familiar challenges: access management, connecting instances to the network securely, manage multiple tenant on a device and other sharing considerations
    - New challenges: remote locations, power/networking constraints, moving target

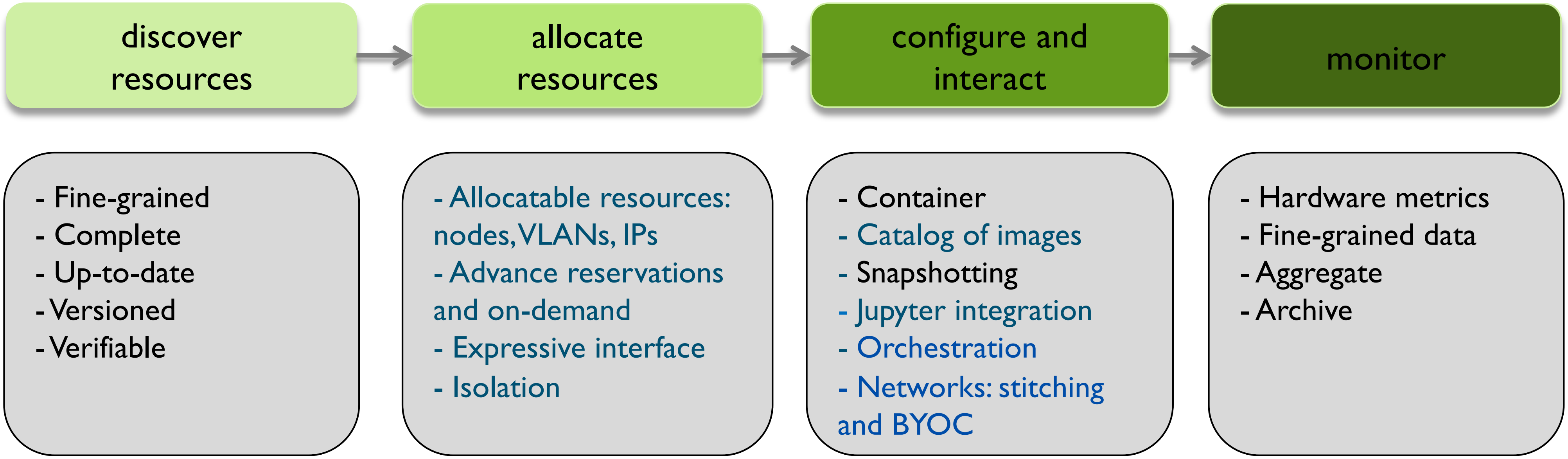
BUILDING CHI@EDGE

From this…



…to this!

CHI@EDGE EXPERIMENTAL WORKFLOW (PREVIEW)



discover resources → allocate resources → configure and interact → monitor

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

- Container
- Catalog of images
- Snapshotting
- Jupyter integration
- Orchestration
- Networks: stitching and BYOC

monitor

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

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)

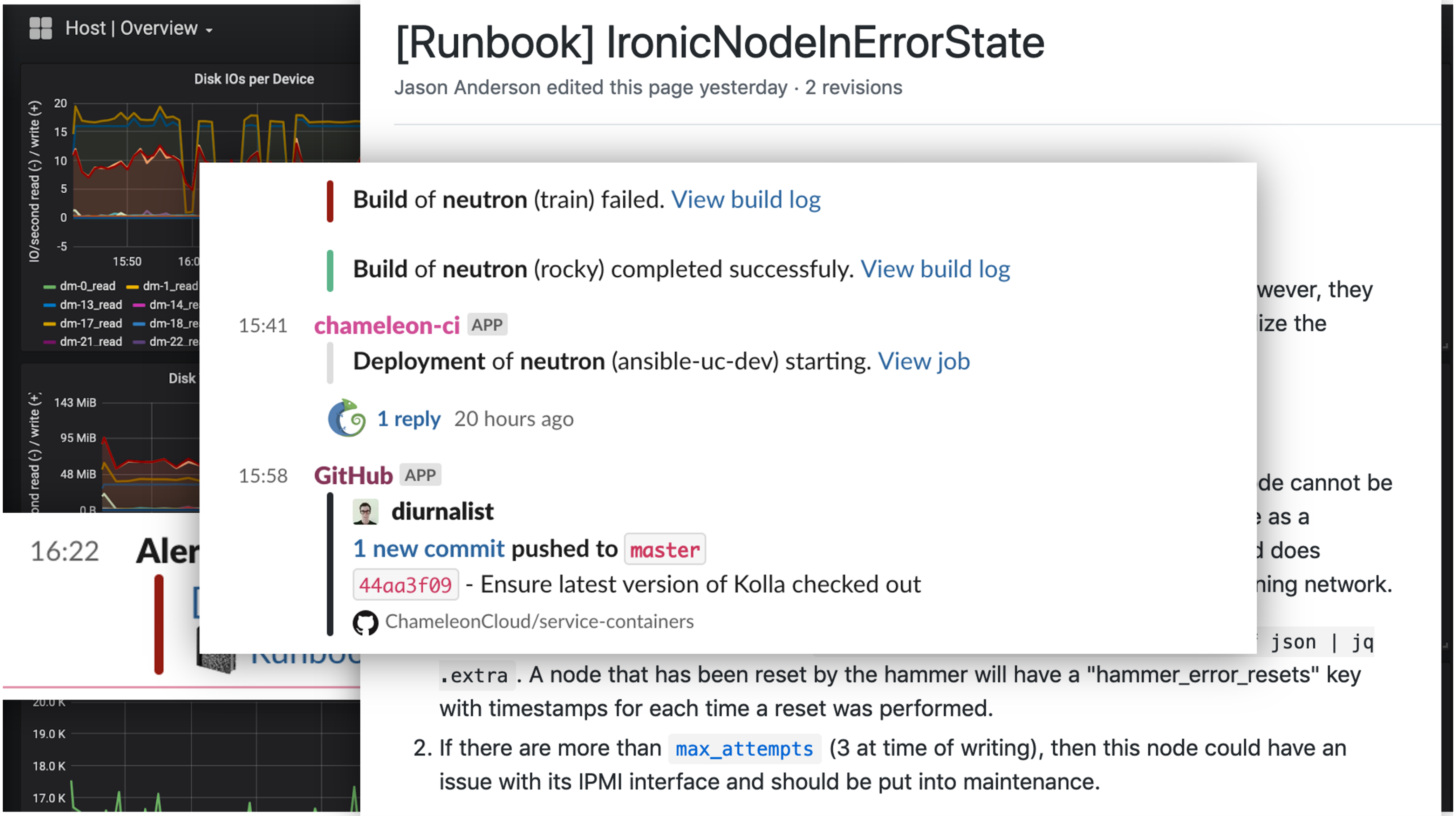
Mixed ownership model: bring your own device(s)!

JOIN US FOR THE SUMMER OF CHAMELEON!

- Early June: federated identity, GUI/CLI, pyton+Jupyter, public IP capability, homogeneous device pool (raspberry pi), no advance reservations or availability calendar, multi-tenant
- Late June: advance reservations, availability calendar, single-tenant, heterogeneous device pool (e.g., NVIDIA Nano, SDR)
- July: BYOD for full-time enrollment and with security attestations/SLAs
- Webinars, see https://chameleoncloud.org/learn/webinars/
- Chameleon-edge-users mailing list: https://groups.google.com/g/chameleon-edge-users?pli=1
- Help us build a better testbed!

FAMILIAR PLATFORM, LESSER COST

- 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: lessening cost per user
  - 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)
- Usability via portability and mainstream compatibility tools



16:22 Aler

Runboo

[Runbook] IronicNodeInErrorState

Jason Anderson edited this page yesterday · 2 revisions

wever, they
ize the

de cannot be
e as a
d 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.

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

Build of **neutron** (rocky) completed successfully. 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

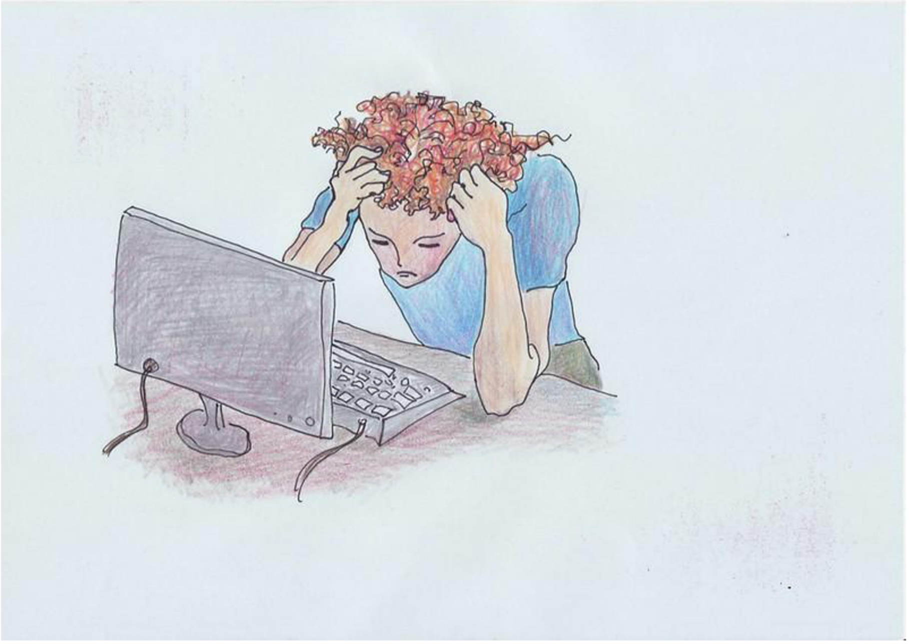
ChameleonCloud/service-containers

EXTENDING FOOTPRINT VIA MIXED OWNERSHIP

- 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: Bring Your Own Device (BYOD) like Chameleon extension but nodes can be added and taken away dynamically
- Adoption
  - New Chameleon Associate Site at Northwestern since fall 2018 – new networking features!
  - Chameleon Legacy Hardware Program

PRACTICAL REPRODUCIBILITY

- Towards a world where experiments are as sharable as papers today
- Goals
  - **Complete** packaging of an experiment – for reproducibility in the long run
  - **Easy to repeat** packaging – for repeatability in the short run
- Introducing variation
  - Extending impact: making it easier for others to **build on your research** (and cite it!)
  - Extending lifespan: making it **easier to adapt** for future environments (newer/different OS, updated hardware)
- Creating a market for experiments



Your reader today ...

... could be you tomorrow!

Flickr (harrycameron)

PRACTICAL REPRODUCIBILITY

- 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

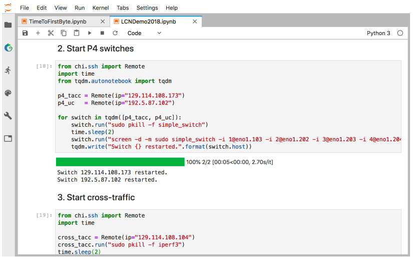
EXPERIMENT SHARING IN CHAMELEON

- 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

BENNY GOODMAN—VOL. 1 ('30s)/SHEP FIELD
BENNY GOODMAN—VOL. 2/MCKINNEY'S CO
BENNY CARTER/ALVINO REY/HARLAN LEON
GLENN MILLER—VOL. 1/ WILL BRADLEY/ ORRIN
ide up

PACKAGING SHARABLE EXPERIMENTS

Literate Programming with Jupyter



2. Start P4 switches

```
[18]: from chi.ssh import Remote
      import time
      from tqdm.autonotebook import tqdm

      p4_tacc = Remote(ip="129.114.108.173")
      p4_uc   = Remote(ip="192.5.87.102")

      for switch in tqdm([p4_tacc, p4_uc]):
          switch.run("sudo pkill -f simple_switch")
          time.sleep(2)
          switch.run("screen -d -m sudo simple_switch -i 1@eno1.103 -i 2@eno1.202 -i 3@eno1.203 -i 4@eno1.204
          tqdm.write("Switch {} restarted.".format(switch.host))
```

```
100% 2/2 [00:05<00:00, 2.70s/it]
Switch 129.114.108.173 restarted.
Switch 192.5.87.102 restarted.
```

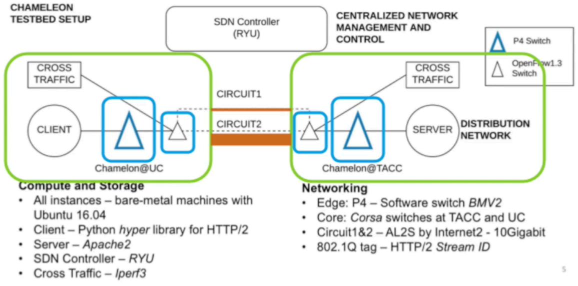
3. Start cross-traffic

```
[19]: from chi.ssh import Remote
      import time

      cross_tacc = Remote(ip="129.114.108.104")
      cross_tacc.run("sudo pkill -f iperf3")
      time.sleep(2)
```

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

+



Compute and Storage
- All instances – bare-metal machines with Ubuntu 16.04
- Client – Python *hyper* library for HTTP/2
- Server – *Apache2*
- SDN Controller – *RYU*
- Cross Traffic – *Iperf3*

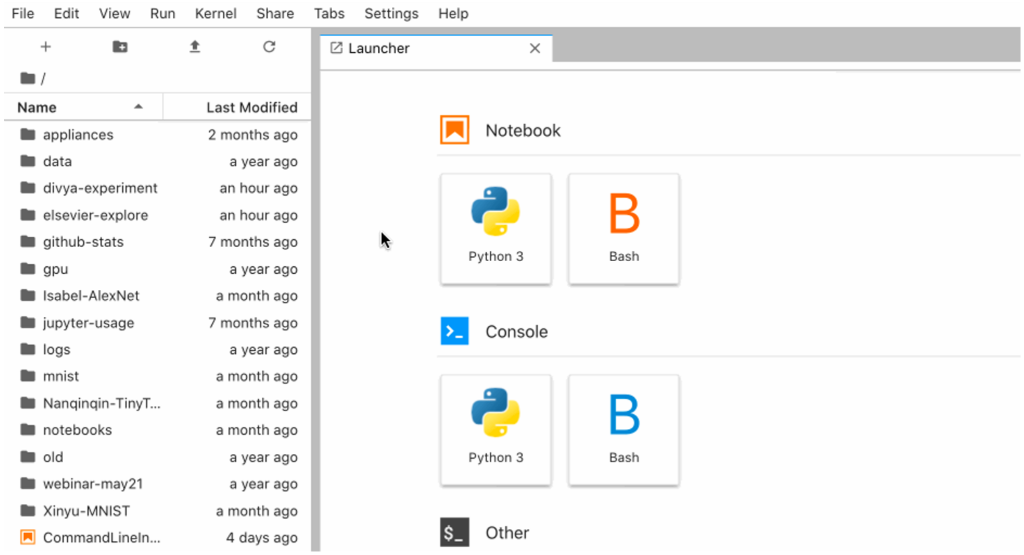
Networking
- Edge: P4 – Software switch *BMV2*
- Core: *Corsa* switches at TACC and UC
- Circuit1&2 – AL2S by Internet2 - 10Gigabit
- 802.1Q tag – HTTP/2 *Stream ID*

Complex Experimental containers

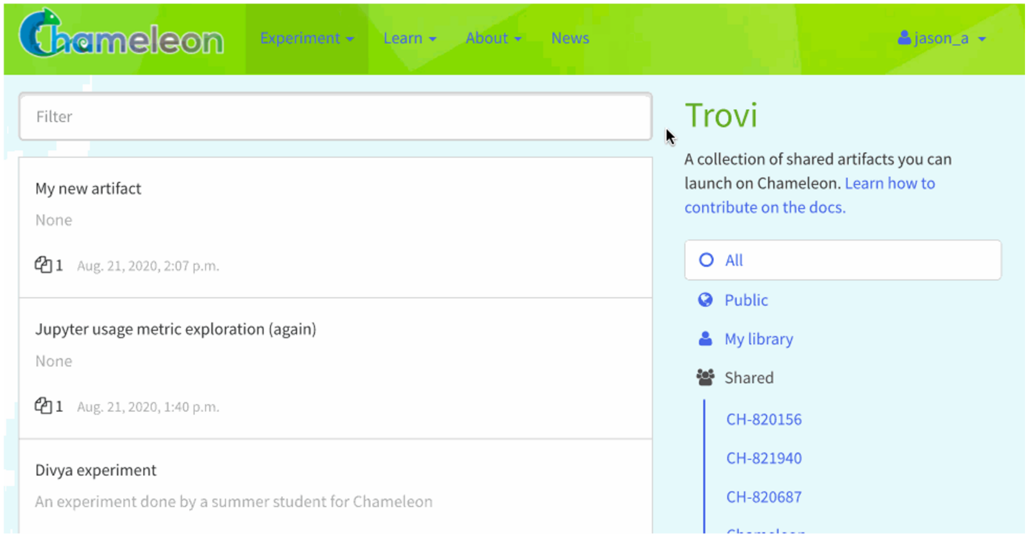
- Repeatability by default: Jupyter notebooks + Chameleon experimental containers
  - JupyterLab for our users: use jupyter.chameleoncloud.org with Chameleon credentials
  - Interface to the testbed in Python/bash + examples (see LCN’18: https://vimeo.com/297210055)
  - Especially for highly distributed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

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

TROVI: CHAMELEON'S EXPERIMENT PORTAL



Name	Last Modified
appliances	2 months ago
data	a year ago
divya-experiment	an hour ago
elsevier-explore	an hour ago
github-stats	7 months ago
gpu	a year ago
Isabel-AlexNet	a month ago
jupyter-usage	7 months ago
logs	a year ago
mnist	a month ago
Nanqinqin-TinyT...	a month ago
notebooks	a month ago
old	a year ago
webinar-may21	a year ago
Xinyu-MNIST	a month ago
CommandLineIn...	4 days ago



My new artifact

None

1 Aug. 21, 2020, 2:07 p.m.

Jupyter usage metric exploration (again)

None

1 Aug. 21, 2020, 1:40 p.m.

Divya experiment

An experiment done by a summer student for Chameleon

Trovi

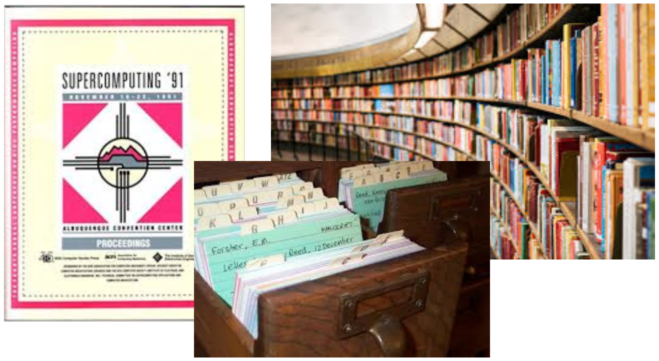
A collection of shared artifacts you can launch on Chameleon. Learn how to contribute on the docs.

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

Familiar research sharing ecosystem



Digital research sharing ecosystem

?

- 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

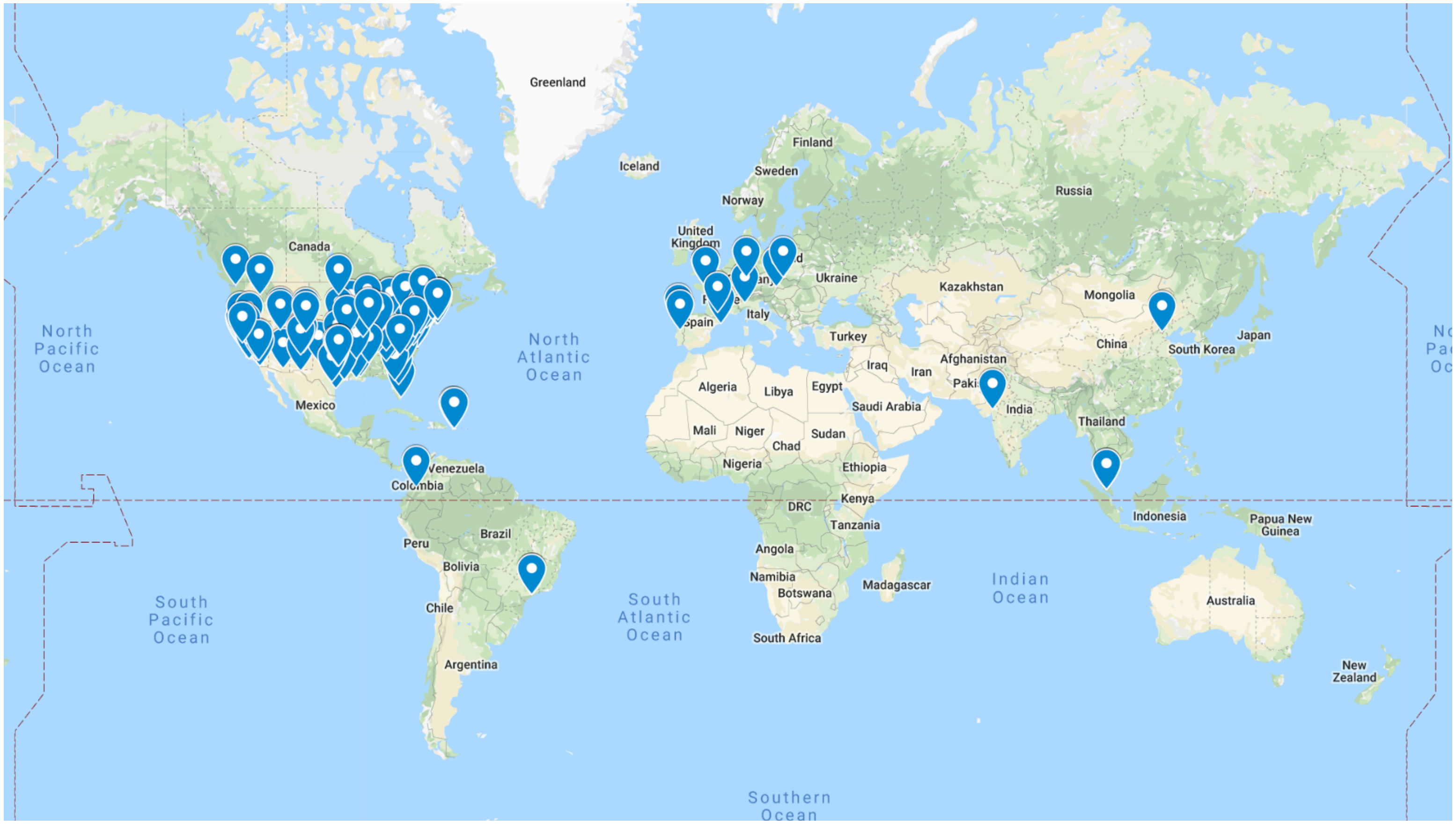
PARTING THOUGHTS

- Scientific instruments: laying down the pavement as science walks on it
  - CHI@Edge: extending our mission from cloud to edge
- Making systems experiment accessible, cheap and ubiquitous
  - Building on a mainstream open source project, investing in building operational tools
  - CHI-in-a-Box: you too can operate a systems testbed!
- 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.chameleoncloud.org

AN OPEN PLATFORM


Greenland
Iceland
Finland
Sweden
Norway
Russia
United Kingdom
Canada
Ukraine
Kazakhstan
Mongolia
Italy
Spain
Turkey
Japan
South Korea
China
North Pacific Ocean
North Atlantic Ocean
Iraq
Iran
Afghanistan
Algeria
Libya
Egypt
Saudi Arabia
India
Mexico
Thailand
Mali
Niger
Sudan
Chad
Nigeria
Ethiopia
Venezuela
Colombia
Kenya
DRC
Indonesia
Papua New Guinea
Tanzania
Brazil
Peru
Angola
Bolivia
Namibia
Botswana
Madagascar
Indian Ocean
South Pacific Ocean
South Atlantic Ocean
Australia
Chile
South Africa
Argentina
New Zealand
Southern Ocean