www. chameleoncloud.org

CHAMELEON: AN INNOVATION PLATFORM FOR COMPUTER SCIENCE RESEARCH AND EDUCATION

Kate Keahey

Mathematics and CS Division, Argonne National Laboratory

CS CASE, University of Chicago

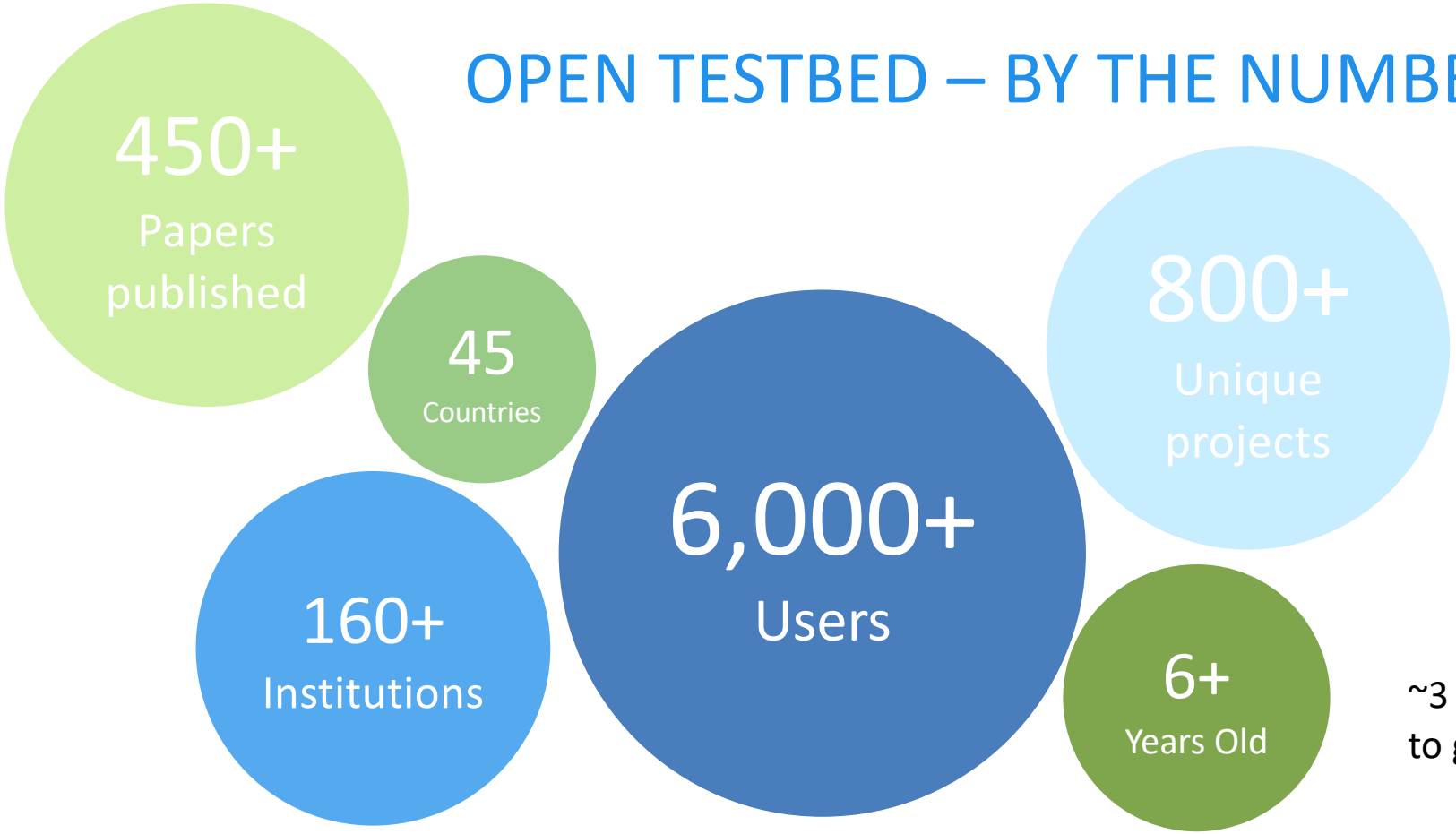
keahey@anl.gov

March 15th, 2022
CS/NERSC Data Seminar

CHAMELEON IN A NUTSHELL

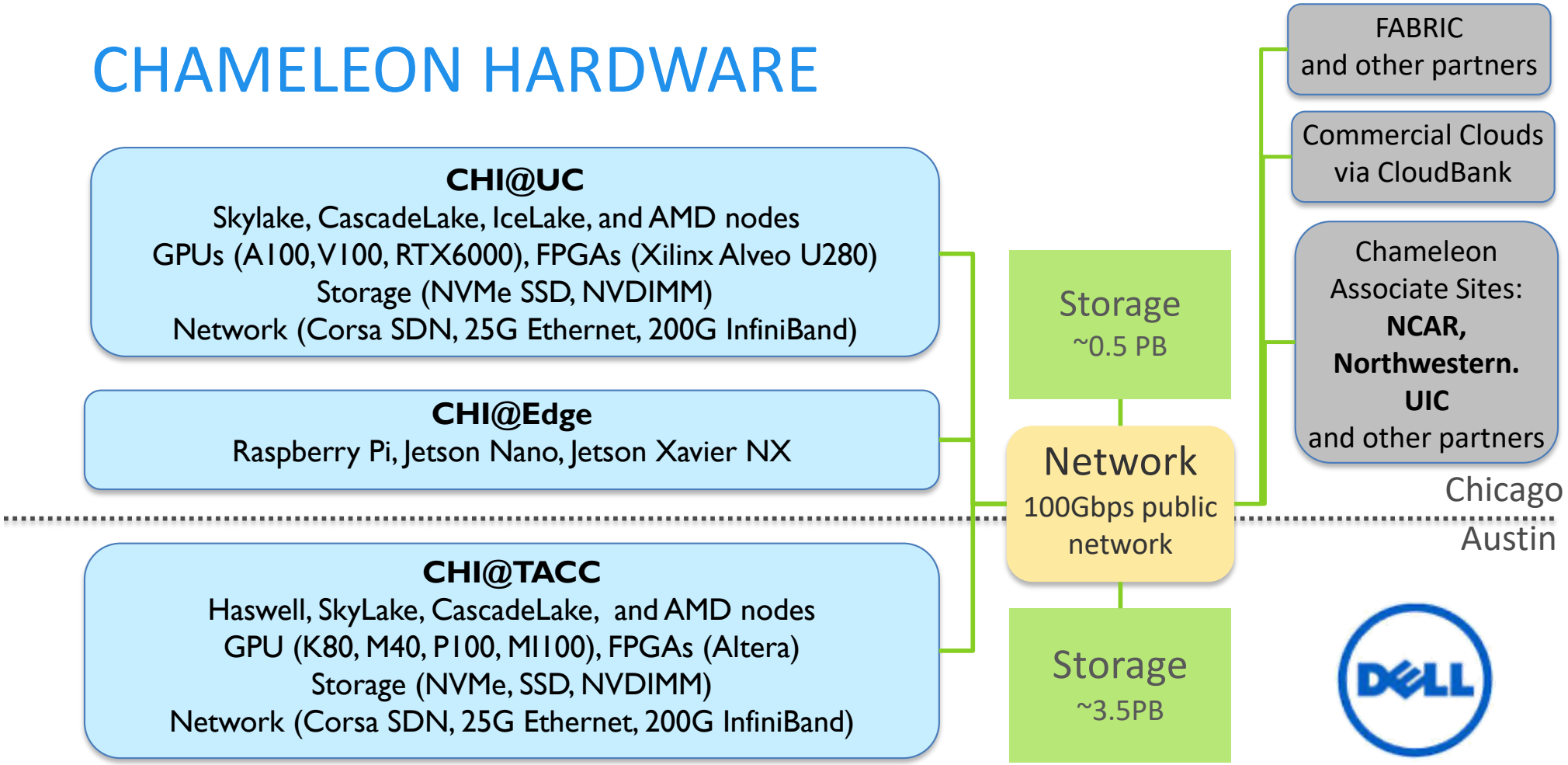
- Chameleons like to change: a testbed that adapts itself to your experimental needs
  - **Bare metal reconfigurability/isolation** + (alternative) KVM cloud (different cost/isolation trade-off)
  - Capabilities: power on/off, reboot, custom kernel, serial console access, etc.
- Balance: diversity and scale – from large to small
  - **Large to small:** from **supercomputing datacenters** (UC, TACC) with 100G network to **edge devices**
  - **Diverse:** FPGAs, GPUs, NVMe, NVDIMMs, Corsa switches, edge devices via CHI@Edge, etc.
  - **Distributed:** CHI-in-a-Box sites at **NCAR, Northwestern, and UIC**
- Cloud++: CHameleon Infrastructure (CHI) via mainstream cloud tech
  - Powered by **OpenStack** with bare metal reconfiguration (Ironic) + “special sauce” (50/50 split)
  - Blazar contribution recognized as official OpenStack component
- Reproducibility, repeatability, and sharing
  - **Jupyter integration** for imperative and non-transactional experiment packaging, **Chameleon daypass** for easy access, **Trovi** for sharing and finding experiments, integration with **Zenodo** for publishing

OPEN TESTBED – BY THE NUMBERS



~3 more years to grow!

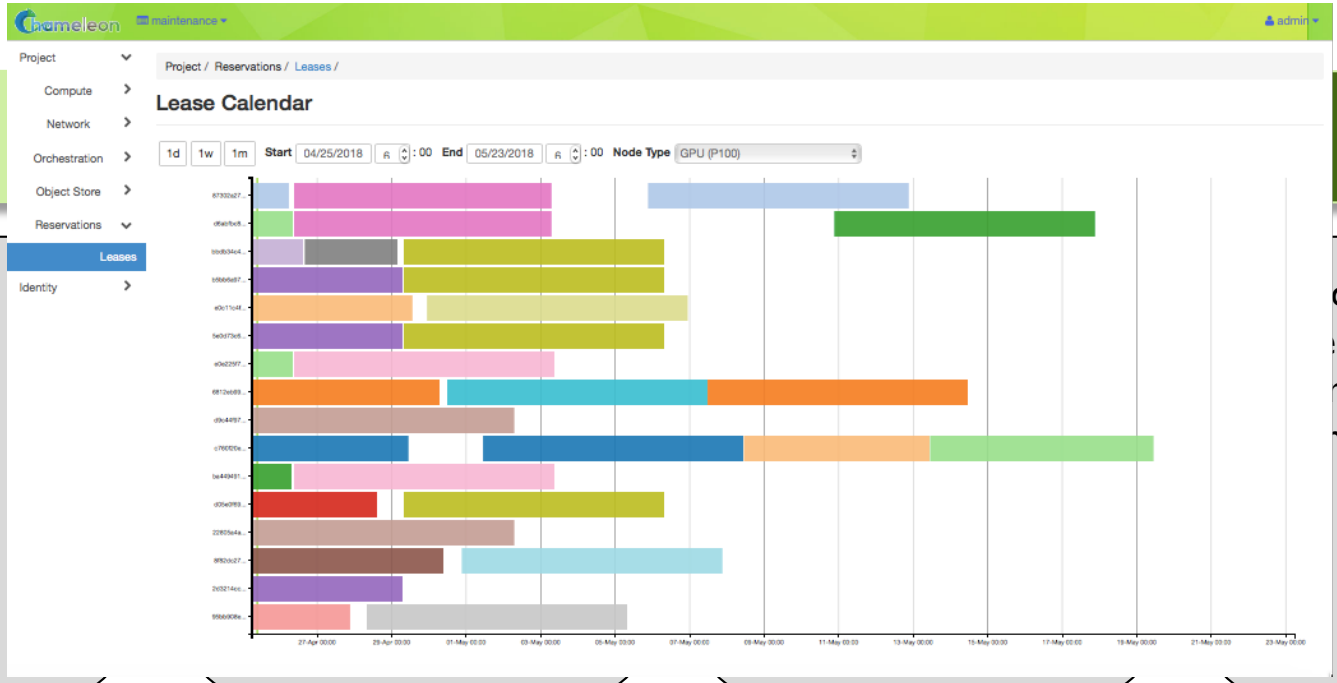
CHAMELEON HARDWARE



CHI EXPERIMENTAL WORKFLOW

discover resources

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



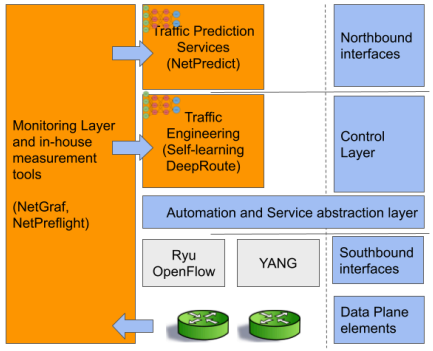
monitor

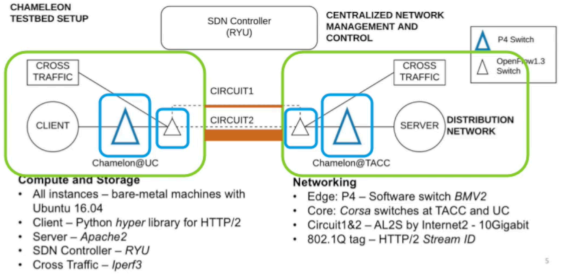
dware metrics
-grained data
regate
hive

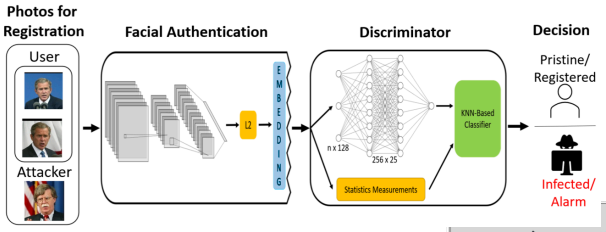
Authentication via federated identity, accessed via GUI, CLI and ***python/Jupyter (python-chi)***

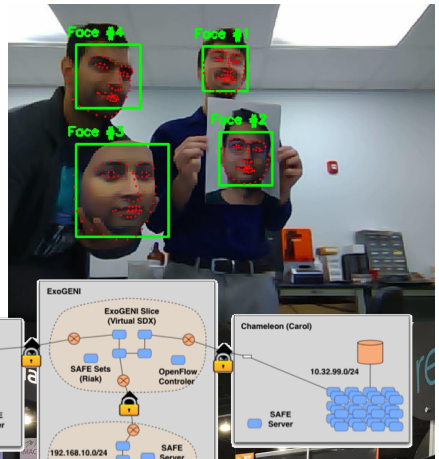
Paper: "Lessons Learned from the Chameleon Testbed", USENIX ATC 2020

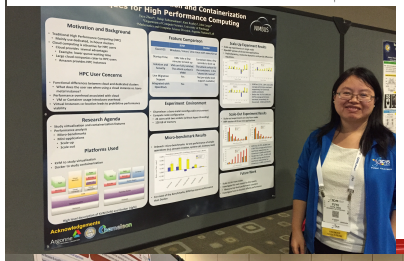
LEAVING NO EXPERIMENT BEHIND



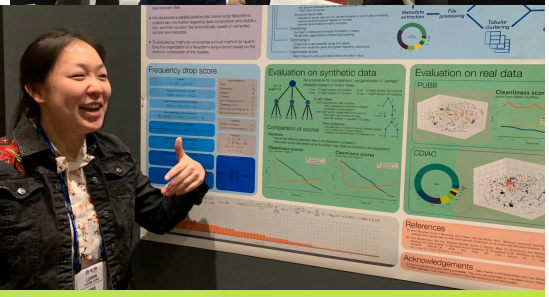


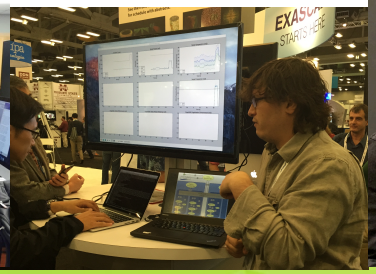






Supporting research projects in architecture, operating systems design, virtualization, power management, real-time analysis, security, storage systems, databases, networking, machine learning, neural networks, data science, and many others.

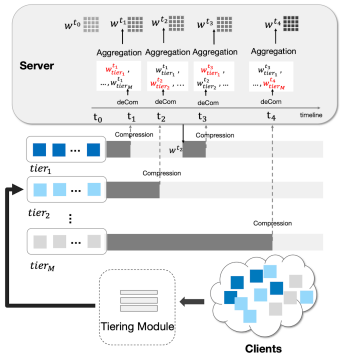




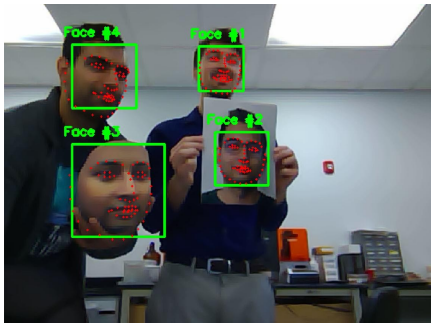
Chameleon www.chameleoncloud.org

Check out user experiment stories on our blog:
https://www.chameleoncloud.org/blog/category/user-experiments/

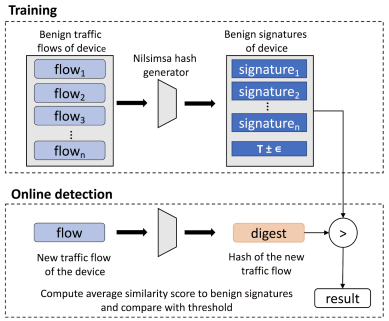
FROM CLOUD TO EDGE WITH CHAMELEON



federated learning



biometrics



network traffic fingerprinting for IoT devices

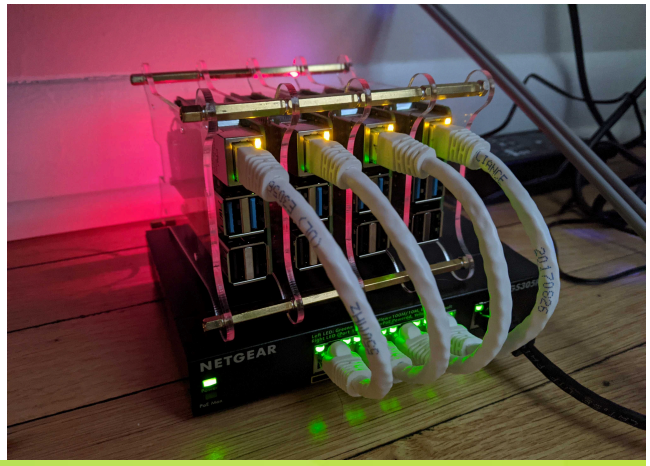
- Increasingly more Chameleon project applications working on IoT/edge
- Simulation/emulation don't always provide the answer: What are the impacts of this approach on power management on edge device? How will the performance transfer to edge? Can we measure the impact of distribution/networking for edge/cloud applications?
- Goal: "realistic edge to cloud experiments from one Jupyter notebook"

NEW IN P3: CHI@EDGE (PREVIEW)

A lot like a cloud!
All the features we know and love – but for edge!

Not at all like a cloud!
Location, location, location!
Not server-class!
IoT: cameras, actuators, SDRs!
And many other challenges!

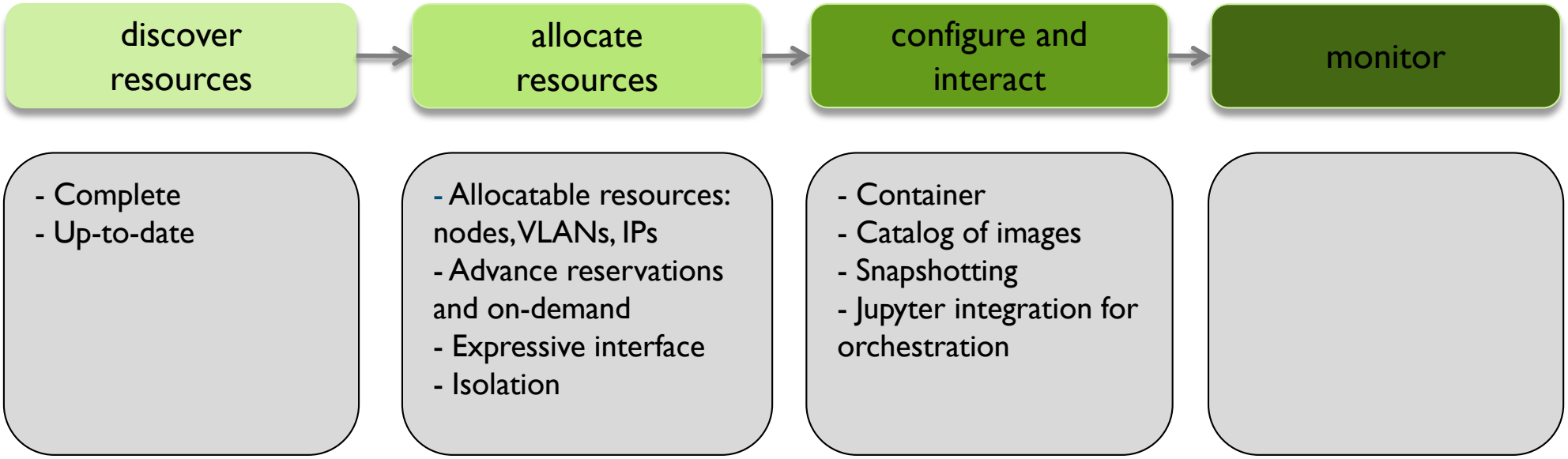
- CHI@Edge: all the features you know and love plus
  - Reconfiguration via container deployment
  - Support for peripherals based on an extensible plug-in model
  - **Mixed ownership** model via an SDK with devices, virtual site, and **restricted sharing**



Paper: "Chameleon@Edge Community Workshop Report" at https://www.chameleoncloud.org/about/papers-and-tech-reports/

FROM CLOUD TO EDGE

CHI@EDGE EXPERIMENTAL WORKFLOW (PREVIEW)



Authentication via federated identity, accessed via GUI, CLI and python/Jupyter (python-chi)

SHARING DEVICES THROUGH CHI@EDGE

- CHI@Edge SDK: fully automate the process of enrolling a device into CHI@Edge
- Support for **restricted leases**
  - You operate your device for your community and leverage our expertise on sharing
  - Your users get seamless access to the devices you operate for them + Chameleon + partnerships
- Access reasonable hardware properties e.g., GPUs
- Peripheral devices
  - Standard camera modules, GPIO, SDR
  - Extensible framework for integrating new devices
- In the process of migrating to CHI@Edge 2.0 now

AUTONOMOUS CARS WITH CHI@EDGE

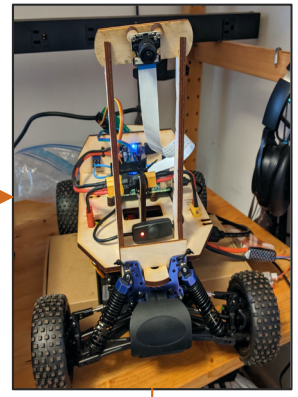
- Goal:
  - Teach machine learning and systems concepts using remote autonomous cars
- Challenges:
  - Control the cars remotely: manual workflows require lots of teacher effort
  - Iterate on code while learning and exploring
  - Collect, store, and process large datasets
- CHI@Edge:
  - Car reservations
  - Access through JupyterHub
  - Provides consistent network connection
  - Deploy code and collect results with repeatable workflows

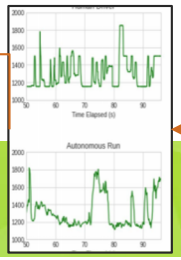
Rick Anderson
Virtual Worlds, Director
Rutgers University

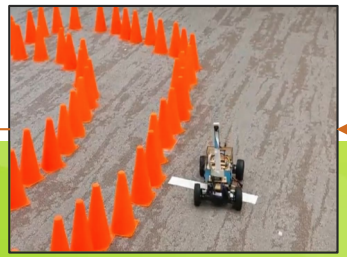
jupyterlab

chameleon

```
#let Chameleon know where to operate
chi.use_site("CHI@Edge")
chi.set("project_name", "CHI-??????")

#reserve a container lease
lease.add_device_reservation(reservations:=[], count = 2, device_model = "4")
container_lease = lease.create_lease("lease", reservations)
lease.wait_for_active(container_lease["id"])
print(f"Lease: {container_lease['name']} is available.")

#provision containers and append them to a hashmap
PORT = "7777"
DIR = "/var/www/html"
letter_list = [chr(ord('a')+i) for i in range(container_lease["reservations"][0]["max"])]
device_list = [container.create_container(name = f"container-{letter}",
                                          image = "id",
                                          image_driver = "glance",
                                          workdir = DIR,
                                          exposed_ports = [PORT],
                                          command = ["python3", "-m", "http.server", PORT],
                                          reservation_id = container_lease["reservations"][0]["id"])
               for letter in letter_list]

edge_device = dict(zip(letter_list, device_list))

container.execute("container-a", "python3 -c \"import this\"")
```



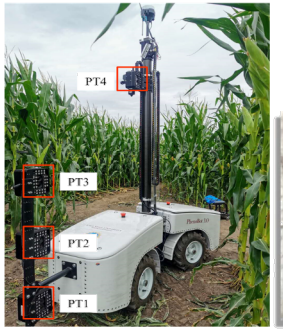




ARA: WIRELESS LIVING LAB FOR SMART & CONNECTED RURAL COMMUNITIES

Hongwei Zhang, ARA PI
Iowa State University

- ARA objectives
  - Enable research to achieve a factor of 10+ reduction in broadband cost and make rural broadband as affordable as urban broadband!
  - Support broadband use cases for rural communities
- ARA wireless living lab
  - Deploy advanced wireless platforms in Central Iowa (>600 square miles); capture systems and application and community contexts of rural broadband
  - Mainstream open-source platforms for living lab management and experimentation: OpenStack, CHI-in-a-Box & CHI@Edge, ONF (SD-RAN, SD-CORE, ONOS), srsRAN, OpenAirInterface etc
  - CHI@Edge: collaborating on spectrum reservations for management of wireless networks and CHI@Edge in a Box

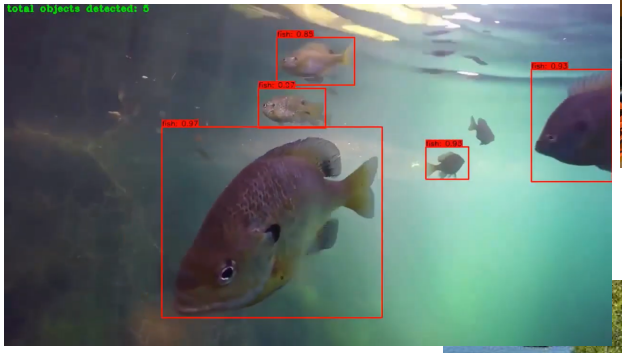


Location and Interior view of ISU Beef Nutrition Research Farm

EDGE FOR MARINE BIOLOGY

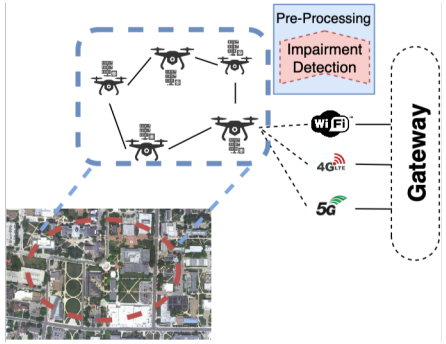
- Goal: map existing fish populations and thereby understand better how pollution impacts their habitat and the general Biscayne Bay ecosystem
- Challenges: What is the best cloud/edge strategy for collecting and analyzing data from the autonomous vehicle (AV)? How does the resolution of video data and quality of network connection influence them?
- CHI@Edge: using CHI@Edge for developing edge to cloud data processing workflows via Jupyter notebooks

Kevin Boswell, Leonardo Bobadilla, and Jonathan Tsen
Florida International University

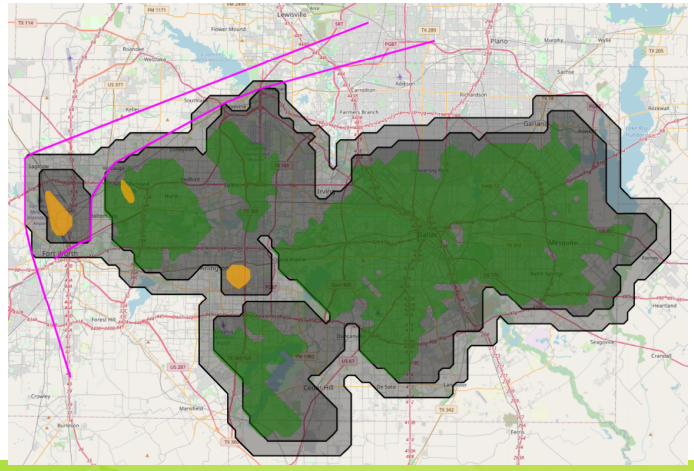


FLYNET: AN 'ON-THE-FLY' PROGRAMMABLE END-TO-END NETWORK-CENTRIC PLATFORM

- Architecture and tools that support edge computing devices in scientific workflows
- Critical for low latency and ultra-low latency applications: e.g., drone video analytics and route planning for drones
- Challenges: integration of compute and networking infrastructure, in-network processing, end-to-end monitoring, workflow management (Pegasus)
- CHI@Edge
  - Use for edge computing experiments
  - Provide experiments that can be reproduced by other researchers
  - FlyNet to provide tools to allow researcher to include CHI@Edge in their workflows

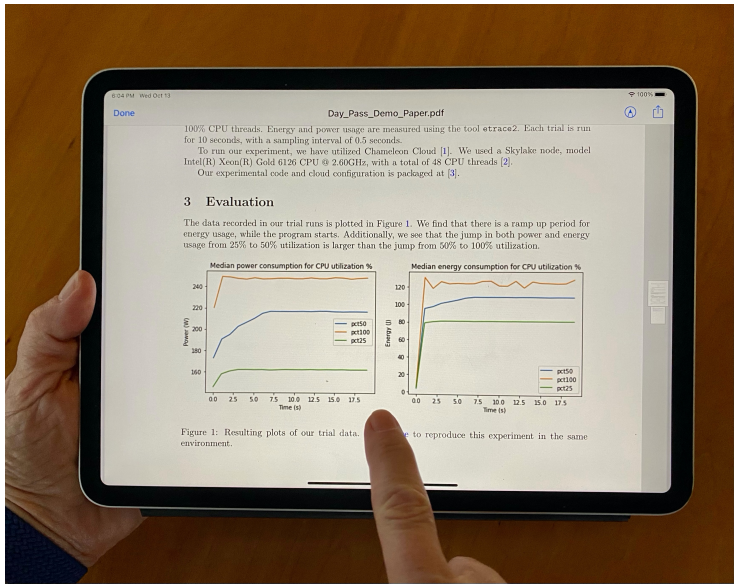


Mike Zink FlyNet PI
U of Mass, Amherst



PRACTICAL REPRODUCIBILITY

- Can experiments be as sharable as papers are today?
- Could it be as easy to provide conditions for reviewers to repeat experiments or data analysis in a paper as it is to organize a PC meeting?
- Can I simply integrate somebody's model into my research instead of reinventing the wheel?
- Can I have so much fun playing with somebody's experiment that discover a new result?
- Can I develop exercises for my class based on most recent research results?



The existence of powerful open testbeds is a fundamental requirement for practical reproducibility

TESTBED AS SHARING PLATFORM

- **Instruments held in common** are a reproducibility imperative
  - Hardware and hardware versions: >105 versions over 5 years
  - Expressive allocation
- Sharing via **cloud pattern**
  - Disk images, orchestration templates, and other artifacts
  - Chameleon >130,000 images, >35,000 orchestration templates and counting
- Testbed as “player” for environments

Paper: “The Silver Lining”, IEEE Internet Computing 2020

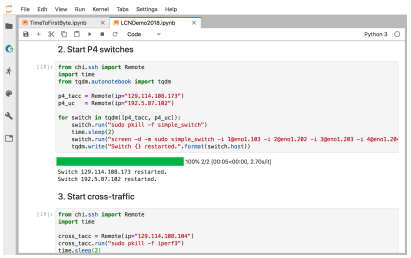
WHAT IS MISSING?

- Packaging: complete, imperative, non-transactional, integrated (literate programming)
- Get access for reproducibility
- Discover/find experiments through various channels

- Package experiment in a way that is cost-effective but also user-friendly
- Give access for reproducibility
- Share work in progress; publish and advertise completed work

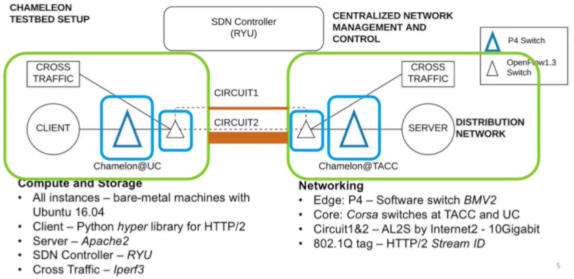
PACKAGING SHAREABLE EXPERIMENTS

Literate Programming with Jupyter



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

+



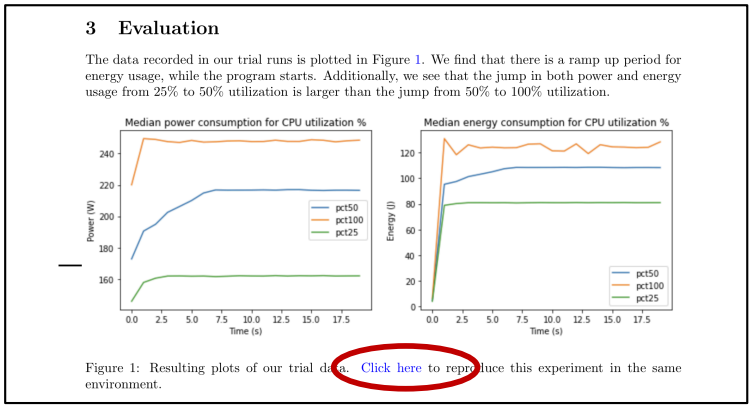
Complex Experimental containers
via programmable interfaces

- Programmable infrastructure: Chameleon "experimental containers" via Jupyter notebooks
  - JupyterLab for our users: use jupyter.chameleoncloud.org with Chameleon credentials
  - Programmable testbed interface with python-chi
  - Example: https://vimeo.com/297210055 (see LCN'18) and others on our youtube channel

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

TESTBED ACCESS WITH CHAMELEON DAYPASS

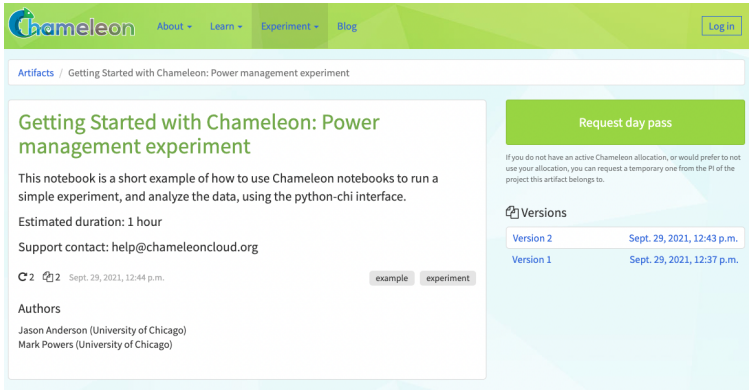
- Authors create a subproject with multiple short-term leases that are long enough to reproduce the experiment
- Readers click through data of a published experiment, request a daypass, and reproduce either the experiment or data analysis



3 Evaluation

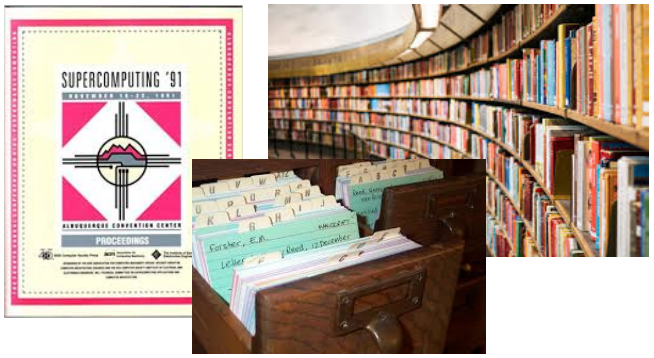
The data recorded in our trial runs is plotted in Figure 1. We find that there is a ramp up period for energy usage, while the program starts. Additionally, we see that the jump in both power and energy usage from 25% to 50% utilization is larger than the jump from 50% to 100% utilization.

Figure 1: Resulting plots of our trial data. Click here to reproduce this experiment in the same environment.



SHARING AND FINDING EXPERIMENTS

Familiar research sharing ecosystem



Digital research sharing ecosystem

Chameleon jupyterhub

?

- Digital publishing with Zenodo: make your experiments citable via Digital Object Identifiers (DOIs)
- Trovi: sharing work in progress
  - BINs to collect all the artifacts, fine-grained sharing, versioning
  - Portal to browse, filter, and find interesting experiments
  - Integrated with Jupyter/Chameleon, Swift, Zenodo, and github (in progress)

zenodo

NEW FRONTIER: COMPOSABLE SYSTEMS

- Challenges: how can we make the most efficient use of all the heterogenous aspects of today's infrastructure? How can we map workloads inside a datacenter most efficiently onto available hardware? What are the availability bottlenecks and how can we scale efficiently across multiple dimensions to avoid them?
- How can Chameleon users leverage and/or experiment with composable systems?
- Can Chameleon itself be a composable system?
  - Can we dynamically present external resources as Chameleon resources?
  - Can we make Chameleon resources dynamically available through other systems?

COMPOSABLE SYSTEMS FOR OUR USERS

- Composable hardware: Liqid
  - Disaggregated pools of components that can be flexibly combined
    - 10x NVIDIA A100 GPUs
    - (2->8)x 2-socket AMD EPYC 7763 64-Core/256G RAM
    - 2x 3.5TB NVMe SSD
  - Fungible versus non-fungible presentation
- Alternative: nodes with GPUs, NVDIMMs, large variety of SSDs, etc. connected by RDMA

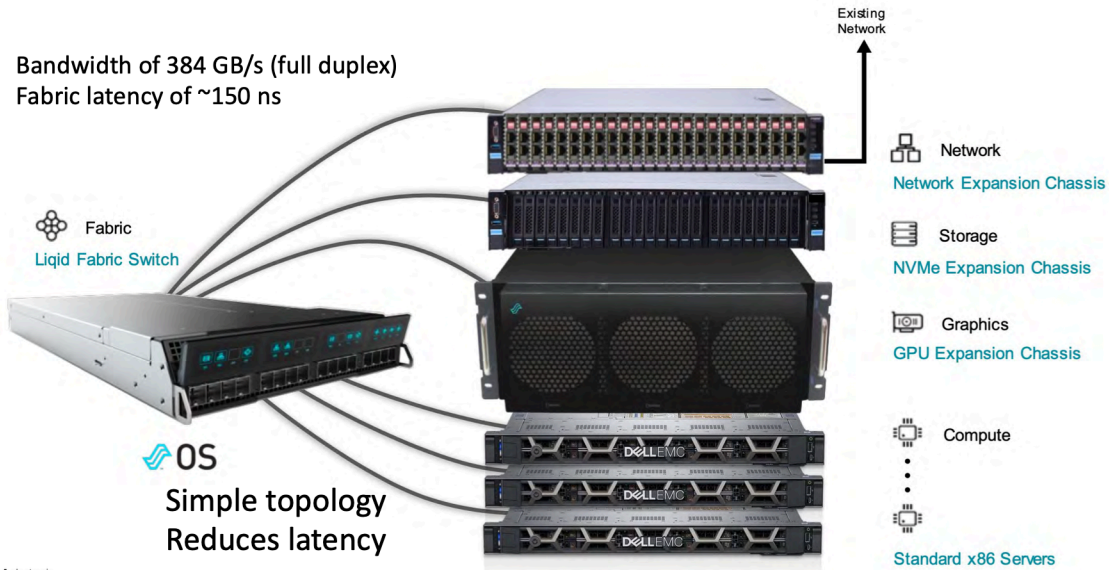
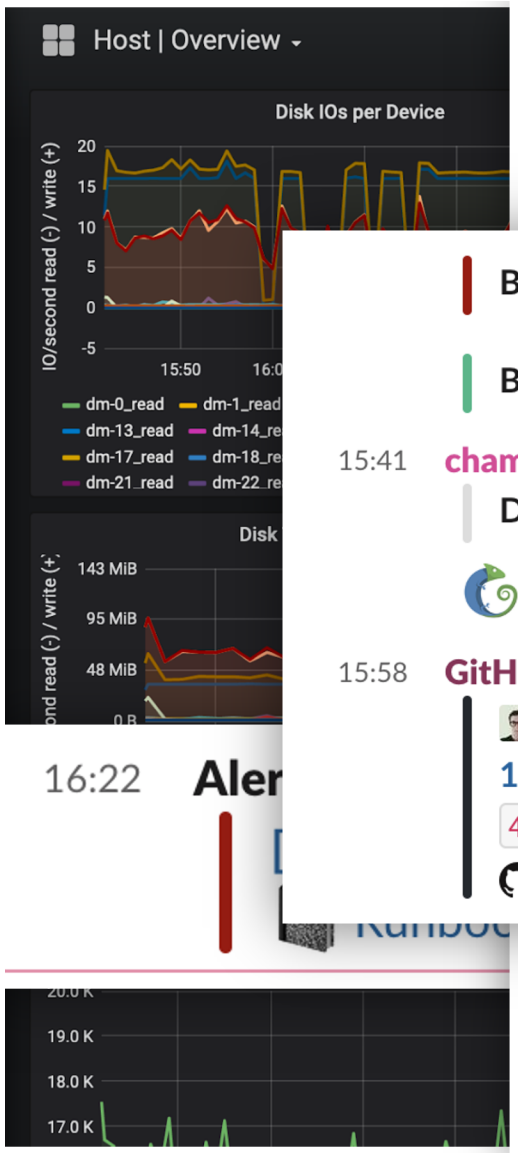


Image courtesy of Lance Long, Electronic Visualization Laboratory, University of Illinois Chicago, NSF Award #CNS-1828265

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

- CHI-in-a-box: packaging of CHameleon Infrastructure (CHI)
  - Packaging of a commodity-based testbed
  - Packages the system as well as the operations model
- Managing inventory with Doni
  - Bring Your Own Device (BYOD) model: allows administrators to dynamically enroll/de-enroll resources, define availability windows, and streamline operational tasks
  - Requires fixed POC (controller node) configuration on the site
- Packaging and operations strategies
  - Hub and spoke management, version-controlled site configuration management as code, containerization, streamlined configuration, monitoring, detection, and remediation tools
  - End-user services



[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
lize 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.

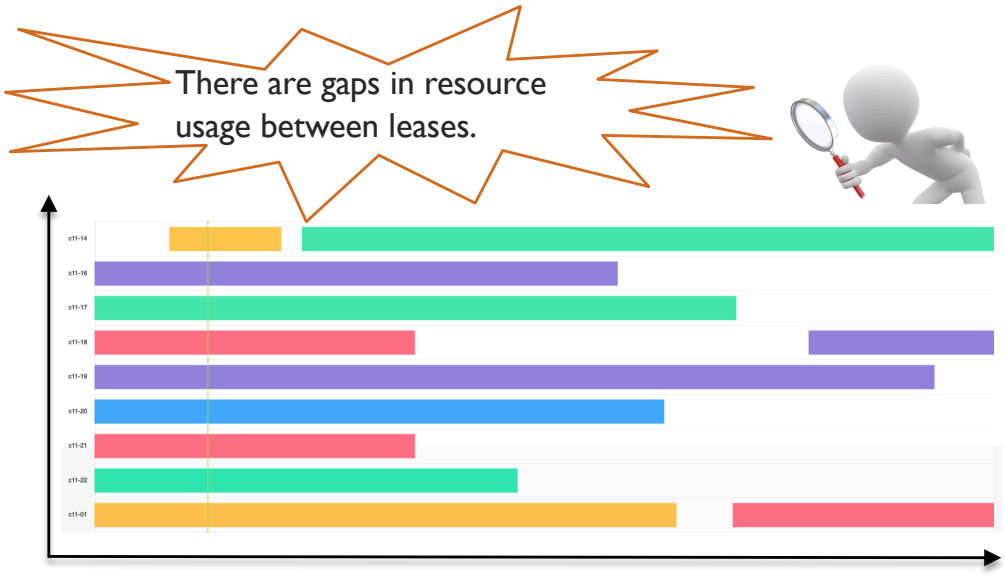
2. If there are more than `max_attempts` (3 at time of writing), then this node could have an issue with its IPMI interface and should be put into maintenance.

CHI-IN-A-BOX SCENARIOS AND ADOPTION

- CHI-in-a-box scenarios
  - **Independent Testbed:** package assumes independent account/project management, portal, and user support
  - **Chameleon Associate Sites:** sites join the Chameleon testbed full time, leverage central services and user support provided by the project
  - **Part-time Chameleon Associate Site:** like Associate Site but inventory is committed on part-time basis (BYOD)
- Adoption
  - Chameleon Associate Site at Northwestern (full support since 01/21), NCAR (since 01/22), UIC (since 02/11)
  - Independent Testbed through ARA (funded in 07/21, currently in construction)

YIN AND YANG: COMPOSABLE SYSTEMS

- Can we fill gaps in resources usage with HTC workflows?
  - Characteristics of a workload that can consume our resources
  - Mechanisms whereby they can be exported
  - Mechanisms that allow them to be consumed
- Collaboration with OSG



Chameleon host lease calendar

Paper: "Dynamically Negotiating Capacity Between On-demand and Batch Clusters", SC18
Paper: "Improving utilization of infrastructure clouds", CCGrid 2011

PARTING THOUGHTS

- Constantly in motion: scientific instruments are laying down the pavement as science walks on it
- **The great exodus:** from cloud to edge: from expensive **provider-owned** hardware to inexpensive **user-owned** hardware using testbed **sharing and connecting** mechanisms
- **Creating market for open science:** open access to **programmable resources** is the necessary condition – the sufficient conditions are in **programmability/packaging, sharing, and access management**
- **Composable systems:** adapting hardware to workloads with diverse characteristics – for our users and our system

Think Big!

(with the help of a small reptile)

www.chameleoncloud.org