www. chameleoncloud.org

CHAMELEON: TAKING SCIENCE FROM CLOUD TO EDGE

Kate Keahey

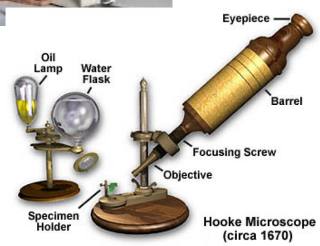
Mathematics and CS Division, Argonne National Laboratory

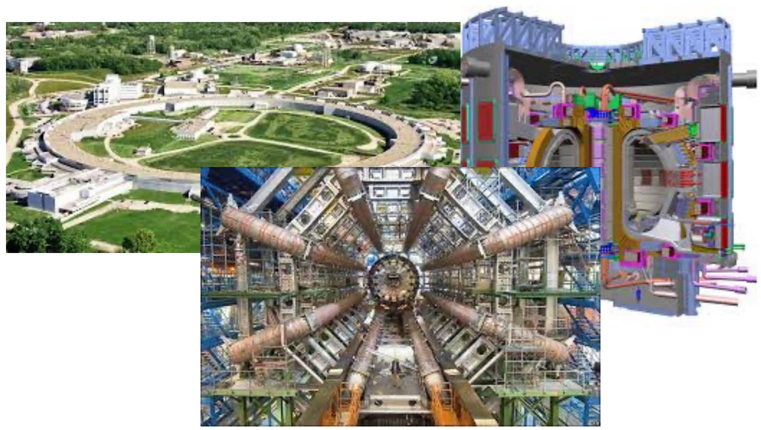
CASE, University of Chicago

keahey@anl.gov

October 21st, 2021
IEEE Cloud Summit 2021

SCIENTIFIC INSTRUMENTS



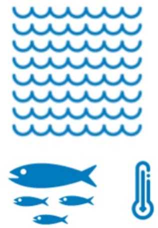


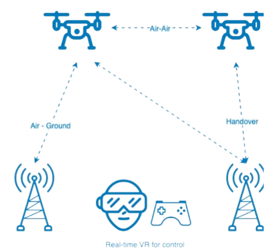
What scientific instruments do Computer Scientists need?

Innovative and diverse hardware, breadth of deployment, freedom to touch and measure every aspect of configuration and behavior.

Constantly evolving!

THE EMERGENCE OF IOT/EDGE





Challenges in connectivity, scale, security, dynamicity, resilience, data and information workflows, management – and many others!

CHAMELEON IN A NUTSHELL

- Chameleons like to change: a testbed that adapts itself to your experimental needs
  - **Deep reconfigurability** (bare metal) and isolation + KVM cloud (different cost/isolation trade-off)
  - Capabilities: 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 originally distributed over 2 sites (**UC, TACC**) connected with 100G network
  - Diverse: ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
  - **CHI-in-a-Box** sites at Northwestern, in preparation: NCAR, IIT, and other places
- 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

400+
Papers published

45
Countries

Over
6,000
Users

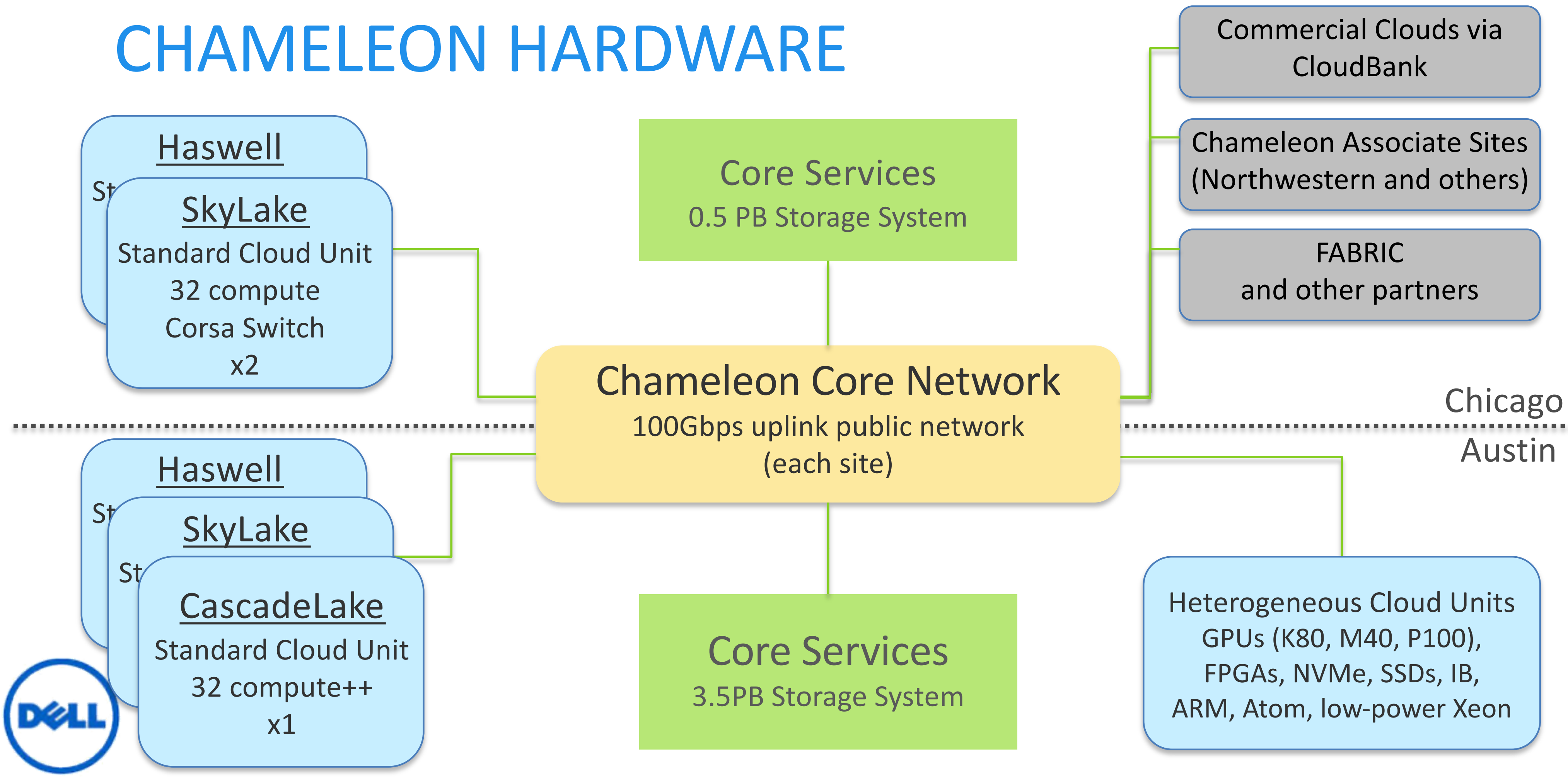
750+
Unique projects

160+
Institutions

6+
Years Old

and 3 more years to grow!

CHAMELEON HARDWARE



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

CHI EXPERIMENTAL WORKFLOW

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
- Deeply reconfigurable
- Appliance catalog
- Snapshotting
- Orchestration
- Jupyter integration
- Networks: stitching and BYOC

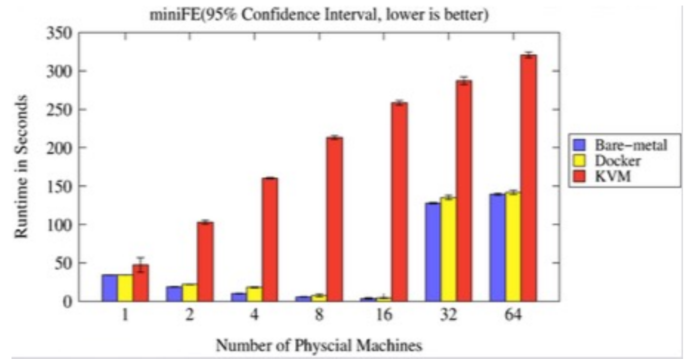
monitor
- Hardware metrics
- Fine-grained data
- Aggregate
- Archive

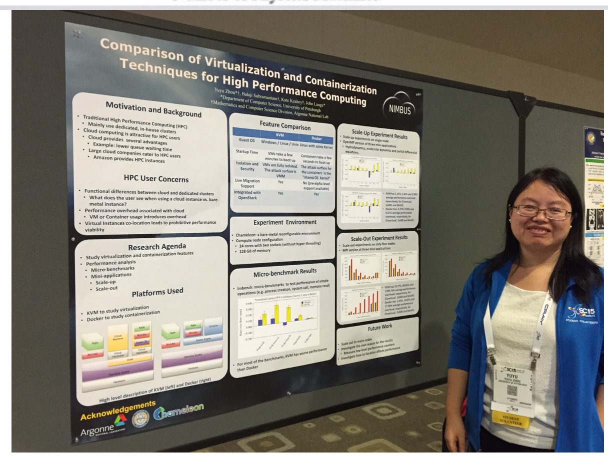
Authentication via federated identity, accessed via GUI, CLI and python/Jupyter

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

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



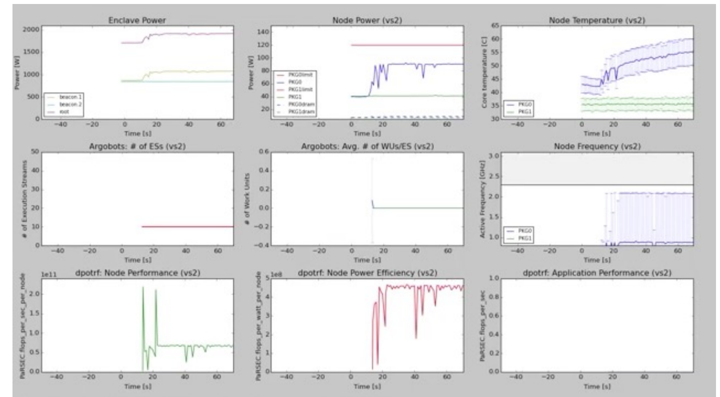


SC15 Poster: "Comparison of Virtualization and Containerization Techniques for HPC"

EXASCALE OPERATING SYSTEMS

- Swann Perarnau, ANL
- Research: exascale operating systems
- Testbed requirements:
  - Bare metal reconfiguration
  - Fast 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"

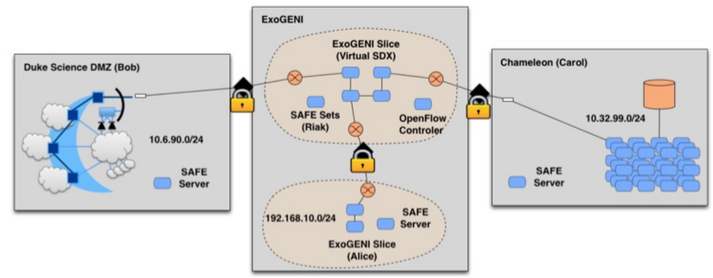


CLASSIFYING CYBERSECURITY ATTACKS

- 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

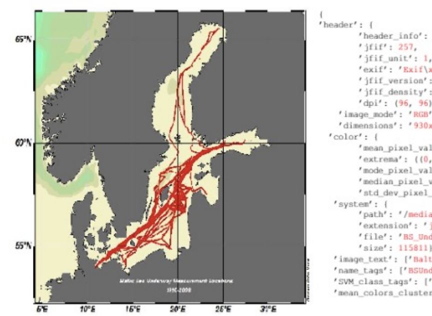
CREATING DYNAMIC SUPERFACILITIES

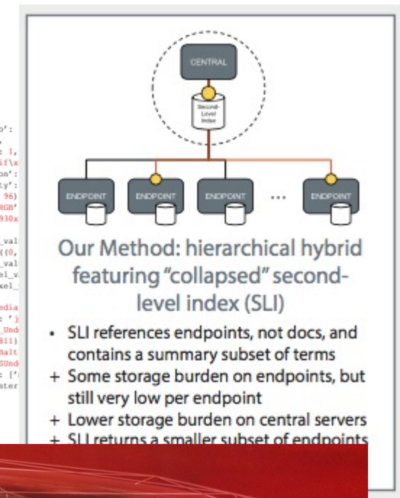
- NSF CICI SAFE, Paul Ruth, RENCI-UNC Chapel Hill
- Creating trusted facilities
  - Automating trusted facility creation
  - Virtual Software Defined Exchange (SDX)
  - Secure Authorization for Federated Environments (SAFE)
- Testbed requirements
  - Creation of dynamic VLANs and wide-area circuits
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



DATA SCIENCE RESEARCH

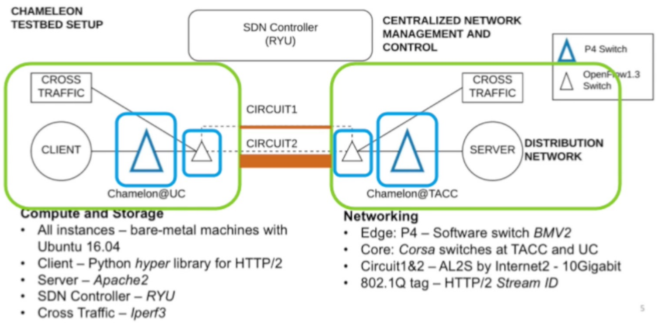
- ACM Student Research Competition semi-finalists:
  - Blue Keleher, University of Maryland
  - Emily Herron, Mercer University
- Searching and image extraction in research repositories
- Testbed requirements:
  - Access to distributed storage in various configurations
  - State of the art GPUs
  - Easy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



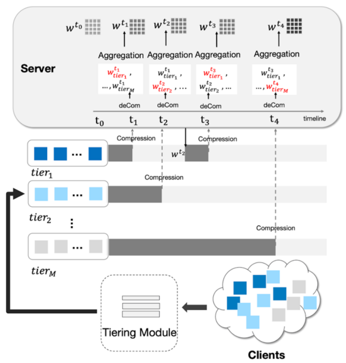
LCN'18: "Application-based QoS support with P4 and OpenFlow"

POWER CAPPING

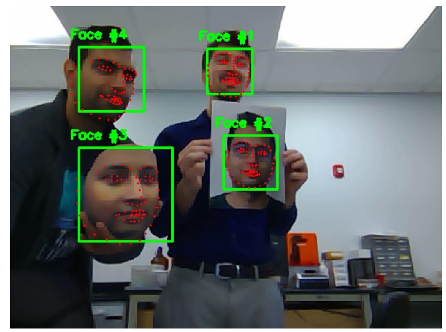
- Harper Zhang, University of Chicago
- Research: hierarchical, distributed, dynamic power management system for dependent applications
- Testbed requirements:
  - Support for large-scale experiments
  - Complex appliances and orchestration (NFS appliance)
  - RAPL/power management interface
- Finalist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

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

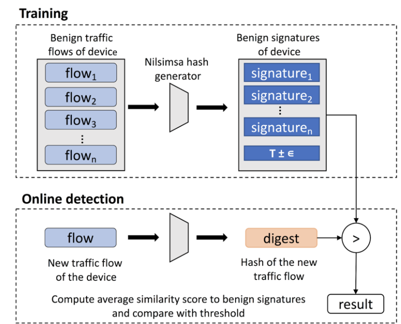
FROM CLOUD TO EDGE WITH CHAMELEON



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"

WHAT DOES AN EDGE TESTBED LOOK LIKE?

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**
  - Chameleon@Edge Community Workshop in September 2021
    https://chameleoncloud.org/chiedge-community-workshop/

WHAT DOES AN EDGE TESTBED LOOK LIKE?

In-network processing

Network/compute heterogeneity

Network Function Virtualization

Network slicing

Intelligent edge algorithms

Edge to cloud workflows

IoT and wireless multi-tenancy

Latency-aware job placement

Data management for edge

Power management

Operating system for edge

Edge security and privacy

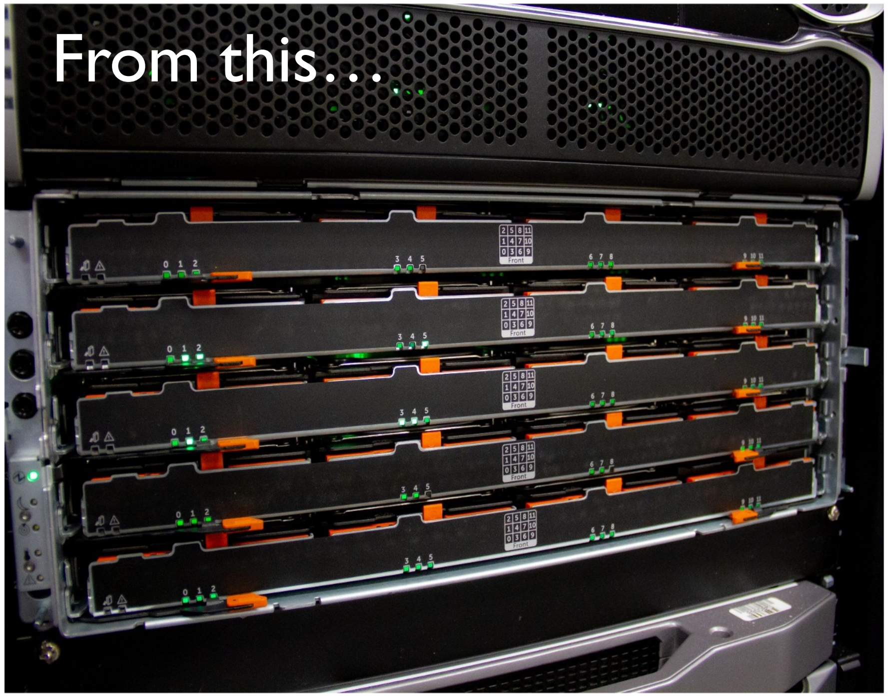
Reliability and Availability

CHI@Edge

chameleon-owned devices

user-owned devices

BUILDING CHI@EDGE



CHI@EDGE EXPERIMENTAL WORKFLOW (PREVIEW)

discover resources → **allocate resources** → **configure and interact** → **monitor**

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, accessed via GUI, CLI and python/Jupyter

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
- CHI@Edge in a Box – in development

AUTONOMOUS CARS WITH CHI@EDGE

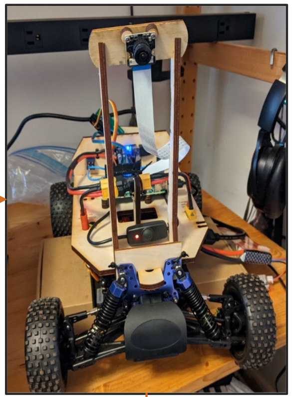
- Goal:
  - Teach machine learning and systems concepts using remote autonomous cars
- Challenges:
  - Control the cars remotely: manual workflows require lots of teacher effort
  - Iterate on code while learning and exploring
  - Collect, store, and process large datasets
- CHI@Edge:
  - Car reservations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

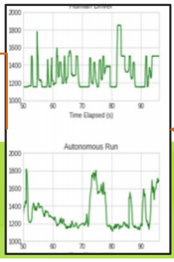
Rick Anderson
Virtual Worlds, Director
Rutgers University

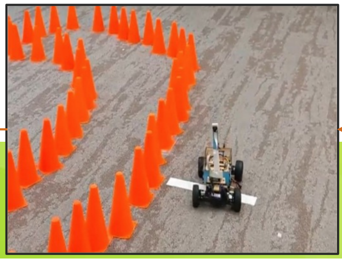
jupyterlab

```
#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\"")
```

Chameleon



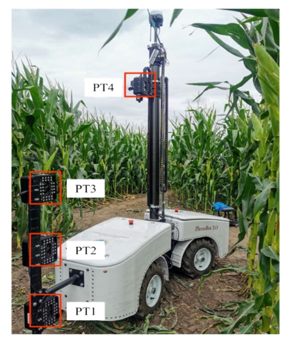




ARA: WIRELESS LIVING LAB FOR SMART & CONNECTED RURAL COMMUNITIES

- 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

Hongwei Zhang, ARA PI
Iowa State University



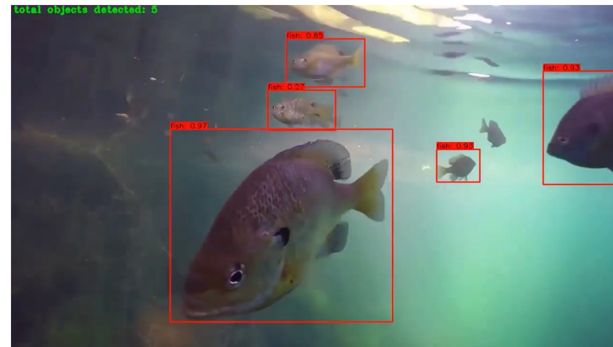
Location and Interior view of ISU Beef Nutrition Research Farm

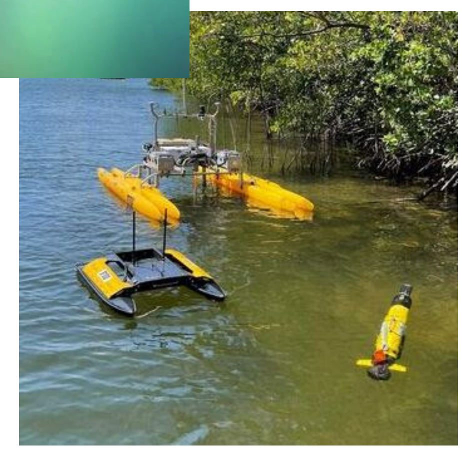
EDGE FOR MARINE BIOLOGY

- 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

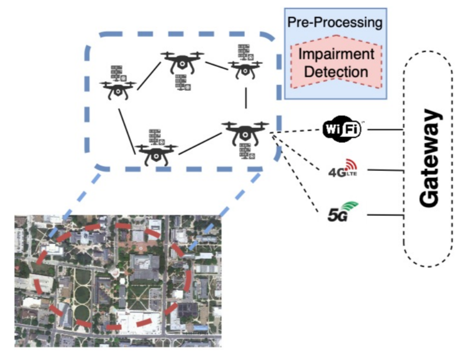


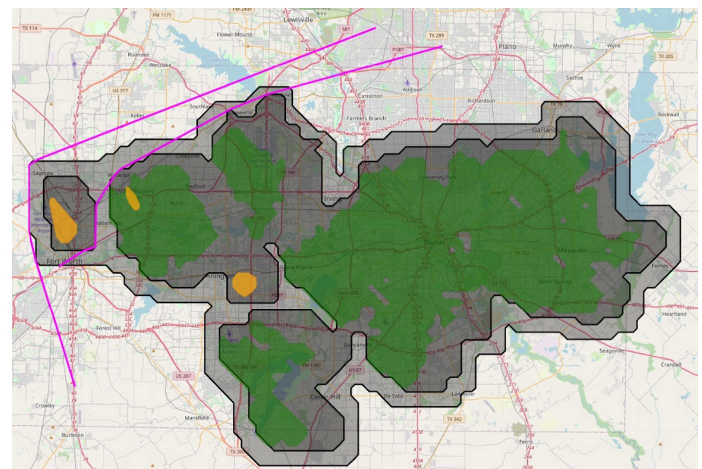



FLYNET: AN 'ON-THE-FLY' PROGRAMMABLE END-TO-END NETWORK-CENTRIC PLATFORM

- 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

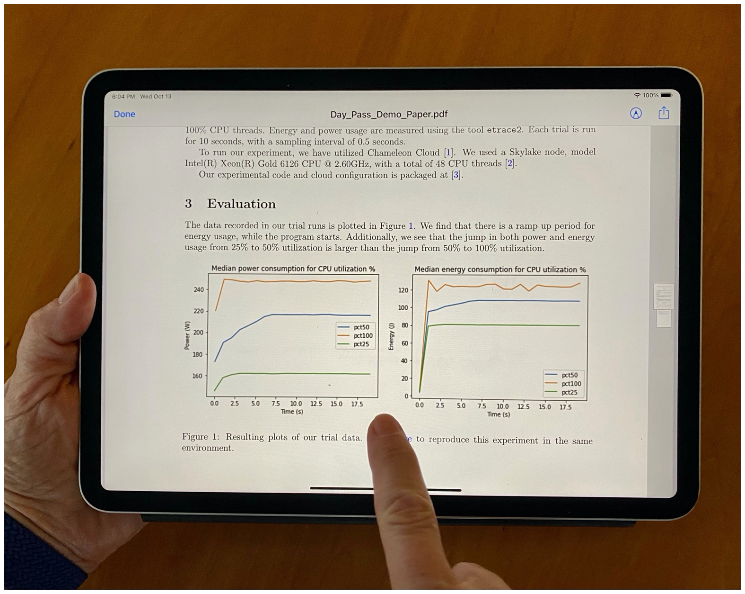


Mike Zink FlyNet PI
U of Mass, Amherst



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

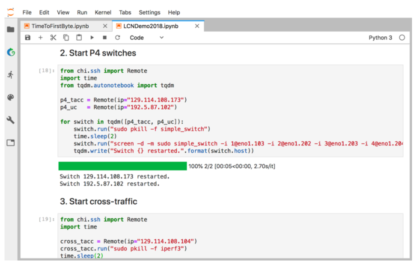
WHAT IS MISSING?

- 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

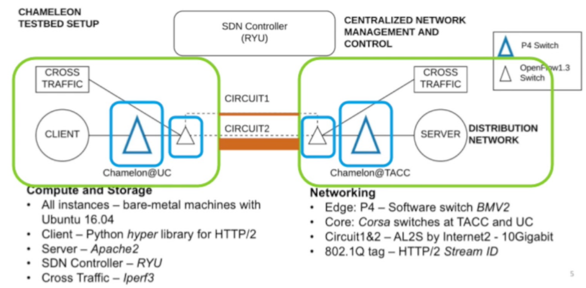
PACKAGING SHARABLE EXPERIMENTS

Literate Programming with Jupyter



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

+



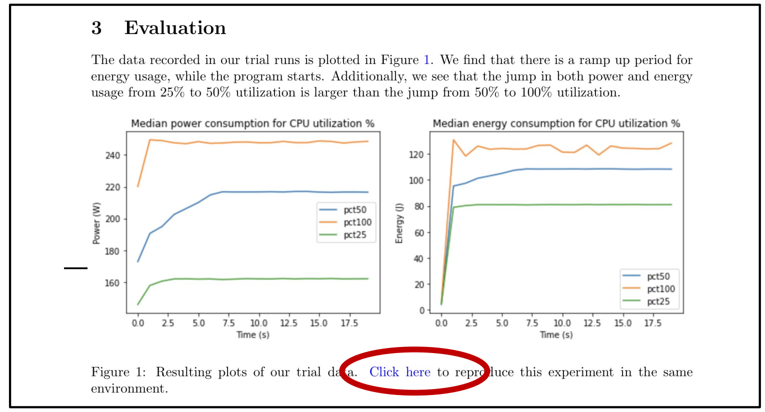
Complex Experimental containers

- 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)

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

TESTBED ACCESS WITH CHAMELEON DAYPASS

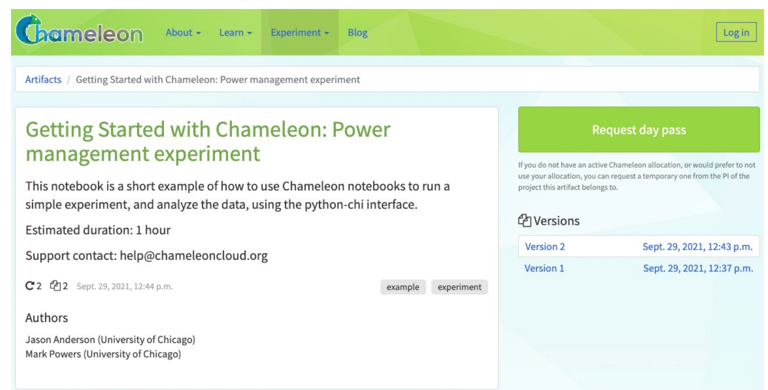
- Authors create a subproject with multiple short-term leases that are long enough to reproduce the experiment
- Readers click through data of a published experiment, request a daypass, and reproduce either the experiment or data analysis



3 Evaluation

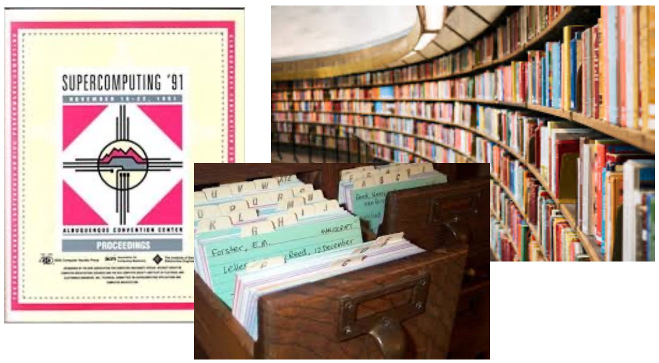
The data recorded in our trial runs is plotted in Figure 1. We find that there is a ramp up period for energy usage, while the program starts. Additionally, we see that the jump in both power and energy usage from 25% to 50% utilization is larger than the jump from 50% to 100% utilization.

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

PARTING THOUGHTS

- Constantly in motion: scientific instruments are laying down the pavement as science walks on it
- **Testbed evolution**: from cloud to edge
  - Before: expensive **provider-owned** hardware as the main draw
  - Now: **user-owned** inexpensive hardware using testbed **sharing and connecting** mechanisms
  - Testbeds == effective **sharing and connecting** mechanism + residual resources
- Sharing your research digitally is more important than ever!
  - Make it easy with Chameleon: public platform, environments as images, packaging, access, and sharing mechanisms at the ready
  - Biggest benefit in emergent area == real incentives

Think Big!

(with the help of a small reptile)

www.chameleoncloud.org