Chameleon

www. chameleoncloud.org

CHAMELEON:
A NEW ECOSYSTEM FOR
EXPERIMENTAL COMPUTER SCIENCE

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

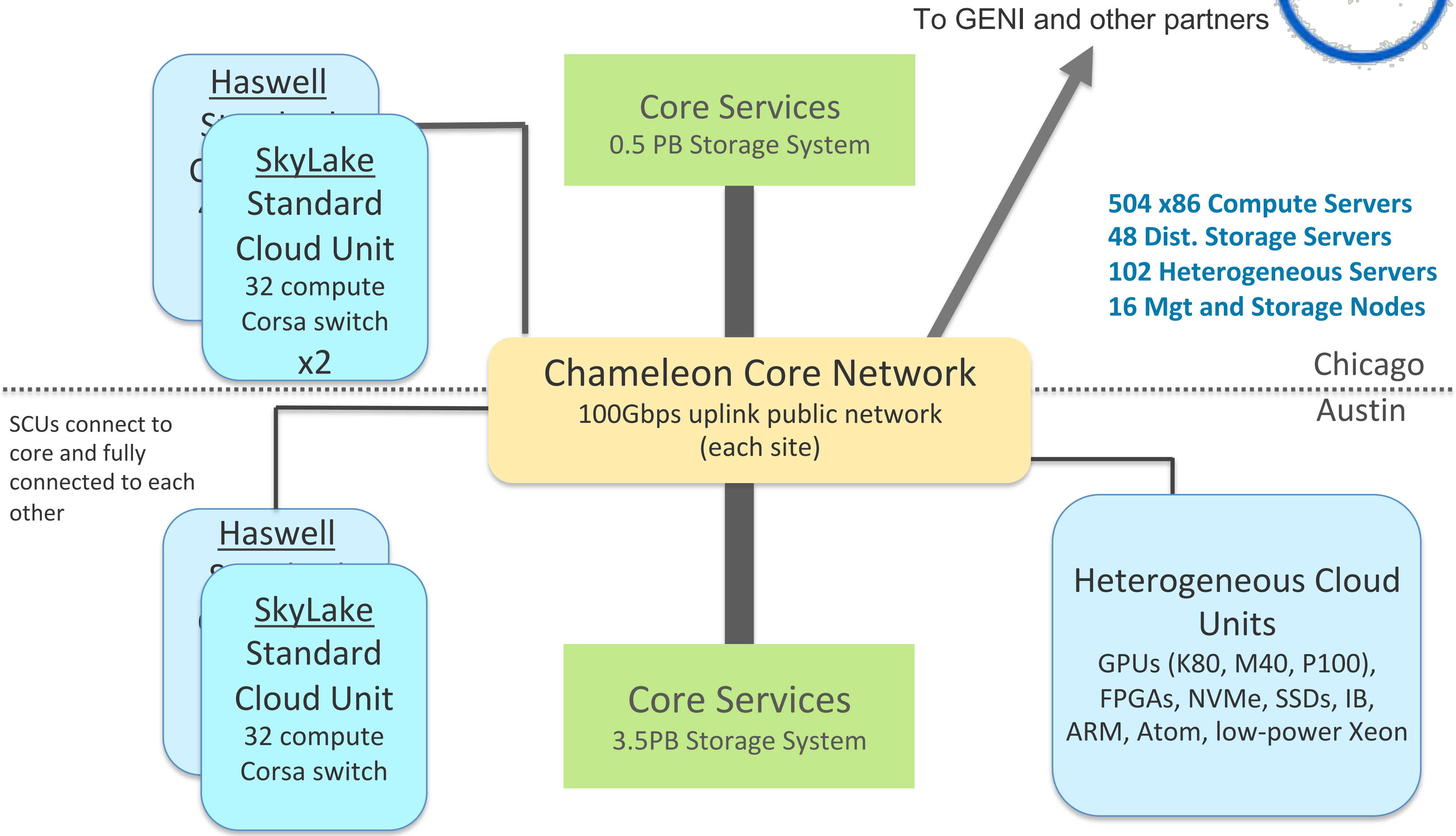
November 8, 2018
Boston University seminar

THE UNIVERSITY OF CHICAGO
TACC
NORTHWESTERN UNIVERSITY
renci
NSF

CHAMELEON IN A NUTSHELL

- **Deeply reconfigurable:** “As close as possible to having it in your lab”
  - Deep reconfigurability (bare metal) and isolation
  - Power on/off, reboot from custom kernel, serial console access, etc.
  - But also – modest KVM cloud for ease of use
- **Combining large-scale and diversity:** “Big Data, Big Compute research”
  - **Large-scale**: ~660 nodes (~15,000 cores), 5 PB of storage distributed over 2 sites connected with 100G network…
  - …and **diverse:** ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
  - **Coming soon**: more storage, more accelerators
- Blueprint for a **sustainable** production testbed: “cost-effective to deploy, operate, and enhance”
  - Powered by OpenStack with bare metal reconfiguration (Ironic)
  - Chameleon team contribution recognized as official OpenStack component
- **Open, collaborative, production** testbed for **Computer Science Research**
  - Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
  - Currently 2,700+ users, 450+ projects, 100+ institutions

CHAMELEON HARDWARE



To GENI and other partners

Haswell

SkyLake

Standard Cloud Unit

32 compute

Corsa switch

x2

Core Services

0.5 PB Storage System

504 x86 Compute Servers
48 Dist. Storage Servers
102 Heterogeneous Servers
16 Mgt and Storage Nodes

Chameleon Core Network

100Gbps uplink public network (each site)

Chicago

Austin

SCUs connect to core and fully connected to each other

Haswell

SkyLake

Standard Cloud Unit

32 compute

Corsa switch

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 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
  - 2 SkyLake Standard Cloud Units (32 node racks); Corsa (DP2400 & DP2200) switches, 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, GPUs (22 nodes), FPGAs (4 nodes)
  - ARM microservers (24) and Atom microservers (8), low-power Xeons (8)
- Coming soon: more nodes (CascadeLake), and more accelerators

REQUIREMENTS FOR EXPERIMENTAL WORKFLOW

discover res

allocate

configure and

monitor

- Fine-
- Com
- Up-to
- Versio
- Verifi

Isolation

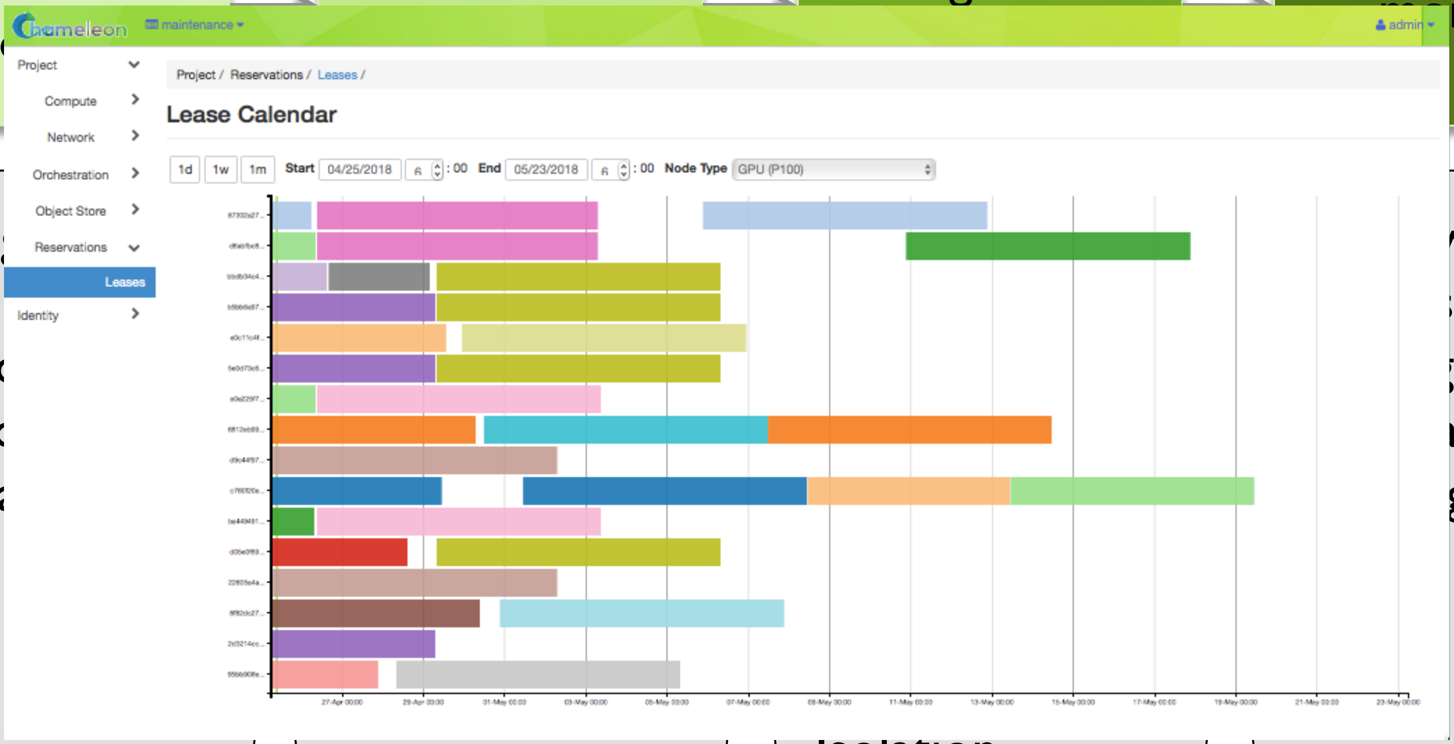
ware

s

grained

ation

gate and



Project / Reservations / Leases /

Lease Calendar

Start 04/25/2018 6 :00 End 05/23/2018 6 :00 Node Type GPU (P100)

Chameleon www.chameleoncloud.org

BUILDING CHI (CHAMELEON INFRASTRUCTURE)

- Requirements for core functionality (proposal stage)
  - Interviews with ~20 research groups
- Architecture**:** **discover**, **provision**, **configure**, and **monitor**
- Technology Evaluation and Risk Analysis
  - Many options: Grid'5000, Nimbus, LosF, OpenStack
  - Final round: Grid'5000 and OpenStack
- Criteria: sustainability as design criterion
  - ***Does it fit our purpose?*** Feature coverage, incl. ease of use
  - ***Can we customize it?*** Open-source, configurable, extendable
  - ***Can we rely on it?*** Stable, scalable, supported
  - Can a CS testbed be built from commodity components?
- A mix of technologies with lots of tweaks (aka "special sauce")
  - Grid'5000 for resource discovery and hardware verification
  - OpenStack for the rest (using Blazar, Ironic, and core OpenStack services)
- Core functionality built in just 3 months after evaluation

WHAT IS OPENSTACK?

- Leading open-source IaaS implementation… and more

Traditional software

OpenStack

- Community: ~ 1,500-2,000 developers contributing to each release including many big companies contributing, e.g. Huawei, Red Hat
- Deployment base:
  - 2017 user surveys logged 1,000 unique deployments (~millions of end users)
  - 60 public cloud data centers, from e.g. Rackspace, OVH
  - Large-scale deployments, e.g. ~100sK cores at CERN

THE MISSING COMPONENT: OPENSTACK BLAZAR

- **Advanced reservation service** for OpenStack
- Originally developed 2013-2014 in the context of power management research
- From early 2015: adaptation for Chameleon
  - Improve stability, integration with Ironic
  - Dashboard improvements (Gantt chart)
  - Incremental operational improvements
- Fall 2016: revival
  - Joined forces with NTT and others working on capacity reservation for NFV
- **Official OpenStack project** in Sep 2017

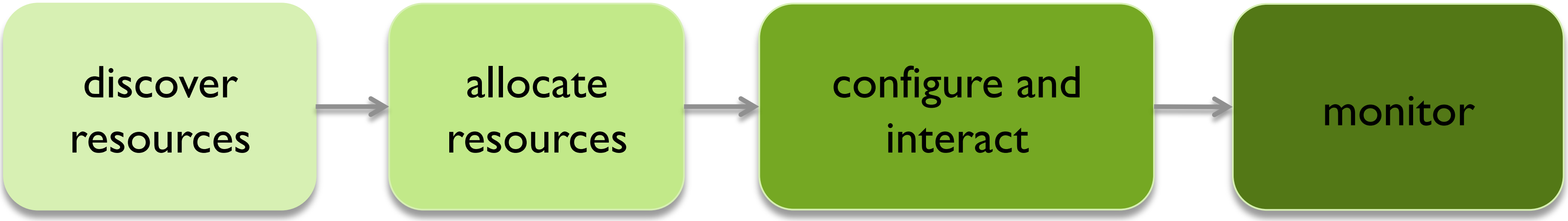
BLAZAR

an OpenStack Community Project

OPENSTACK: LESSONS LEARNED

- The good
  - Large community rapidly developing new features
  - Common requirements → shared effort
  - Commodity infrastructure for sustained use
  - Many users already familiar with OpenStack
- The bad
  - Large community rapidly developing new features
  - Complexity: need to understand core components
  - Some users assume Chameleon is like any OpenStack

SUPPORT FOR EXPERIMENTAL WORKFLOW



discover resources → allocate resources → configure and interact → monitor

Grid'5000 Resource Discovery

OpenStack:
- Nova
- *Blazar*
- Swift

OpenStack
- Ironic
- Neutron
- Glance
- Heat

Other
- Appliances++
- Snapshotting

Network Isolation

OpenStack
- Gnocchi

Agents, custom integration, etc.

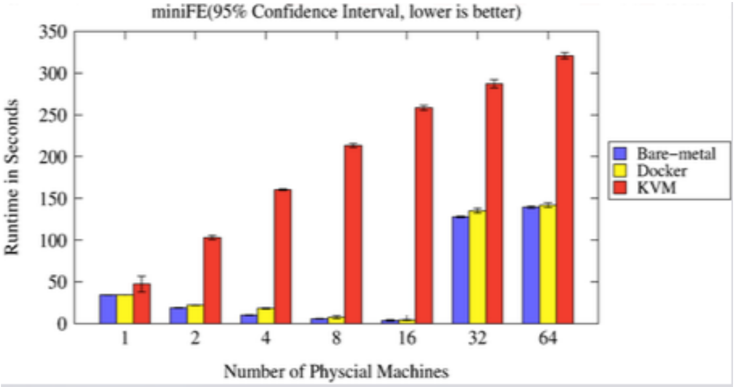
CHI = 65%*OpenStack + 10%*G5K + 25%*"special sauce"

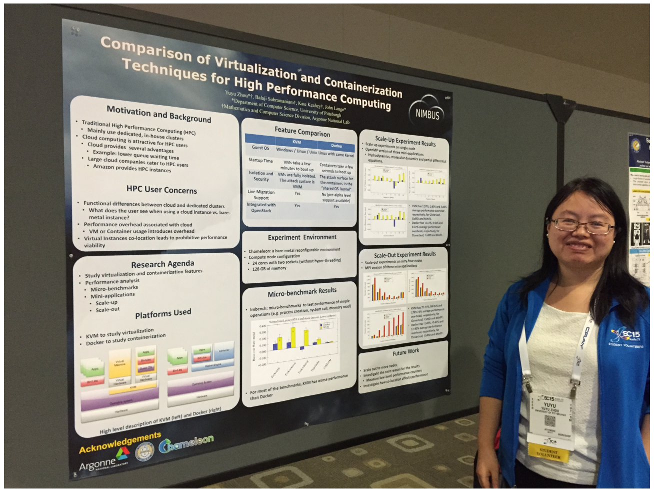
SPECIAL SAUCE, LATELY…

- Networking:
  - **Multi-tenant networking** allows users to provision isolated L2 VLANs and manage their own IP address space (since Fall 2017)
  - **Stitching** dynamic VLANs from Chameleon to external partners (ExoGENI, ScienceDMZs) (since Fall 2017)
  - VLANs + AL2S connection between UC and TACC for **100G experiments** (since Spring 2018)
  - **BYOC– Bring Your Own Controller**: isolated user controlled virtual OpenFlow switches (Summer 2018)
- New lease management features, multi-region configuration, power consumption metrics, whole disk image boot for ARM nodes, serial console access, etc.
- And many more…
  - Appliances, usability improvements, upgrades, etc.

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

Yuyu Zhou*†, Balaji Subramaniam†, Kate Keahey†, John Lange*
*Department of Computer Science, University of Pittsburgh
†Mathematics and Computer Science Division, Argonne National Lab

Motivation and Background
- Traditional High Performance Computing (HPC)
  - Mainly use dedicated, in-house clusters
- Cloud computing is attractive for HPC users
  - Cloud provides several advantages
    - Example: lower queue waiting time
  - Large cloud companies cater to HPC users
    - Amazon provides HPC instances

HPC User Concerns
- Functional differences between cloud and dedicated clusters
  - What does the user see when using a cloud instance vs. bare-metal instance?
- Performance overhead associated with cloud
  - VM or Container usage introduces overhead
- Virtual Instances co-location leads to prohibitive performance viability

Research Agenda
- Study virtualization and containerization features
- Performance analysis
  - Micro-benchmarks
  - Mini-applications
    - Scale-up
    - Scale-out

Platforms Used
- KVM to study virtualization
- Docker to study containerization

High level description of KVM (left) and Docker (right)

Feature Comparison

	KVM	Docker
Guest OS	Windows / Linux / Unix	Linux with same Kernel
Startup Time	VMs take a few minutes to boot up	Containers take a few seconds to boot up
Isolation and Security	VMs are fully isolated. The attack surface is VMM	The attack surface for the containers is the "shared OS kernel"
Live Migration Support	Yes	No (pre-alpha level support available)
Integrated with OpenStack	Yes	Yes

Experiment Environment
- Chameleon: a bare-metal reconfigurable environment
- Compute node configuration
  - 24 cores with two sockets (without hyper-threading)
  - 128 GB of memory

Micro-benchmark Results
- lmbench: micro-benchmarks to test performance of simple operations (e.g. process creation, system call, memory read)
- For most of the benchmarks, KVM has worse performance than Docker

Scale-Up Experiment Results
- Scale-up experiments on single node
- OpenMP version of three mini-applications
  - Hydrodynamics, molecular dynamics and partial differential equations.
- KVM has 1.57%, 1.60% and 2.00% average performance overhead, respectively, for CloverLeaf, CoMD and MiniFE.
- Docker has -0.17%, 0.50% and 0.07% average performance overhead, respectively, for CloverLeaf, CoMD and MiniFE.

Scale-Out Experiment Results
- Scale-out experiments on sixty-four nodes
- MPI version of three mini-applications
- KVM has 73.77%, 84.85% and 1789.74% average performance overhead, respectively, for CloverLeaf, CoMD and MiniFE
- Docker has -1.69%, -0.41% and 17.90% average performance overhead, respectively, for CloverLeaf, CoMD and MiniFE

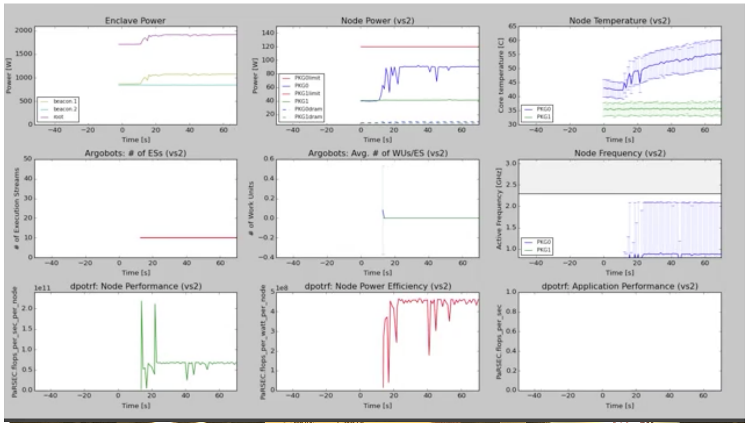
Future Work
- Scale out to more nodes
- Investigate the root reason for the results
- Measure low-level performance counters
- Investigate how co-location affects performance

Acknowledgements

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, params
  - Hardware: accurate information and control over changes, performance counters, many cores
  - Access to same infrastructure for multiple collaborators



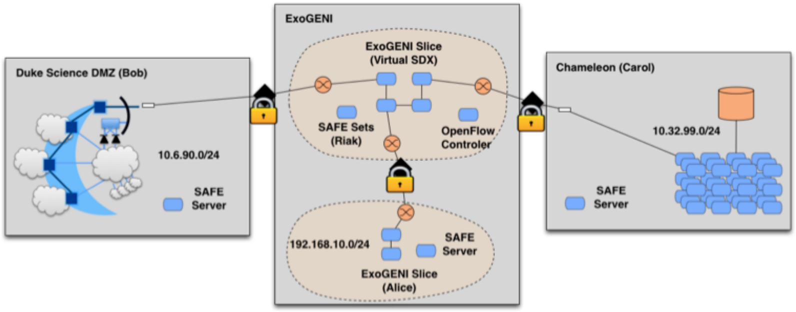
HPPAC'16 paper: "Systemwide Power Management with Argo"

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

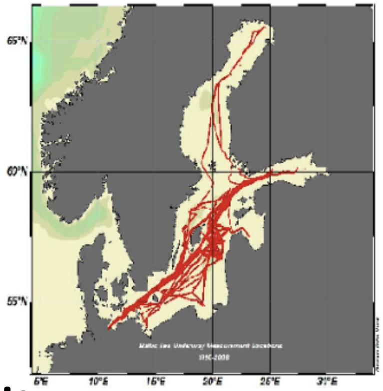
CREATING DYNAMIC SUPERFACILITIES

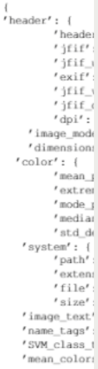
- NSF CICI SAFE, Paul Ruth, RENCI-UNC Chapel Hill
- Creating trusted facilities
  - Automating trusted facility creation
  - Virtual Software Defined Exchange (SDX)
  - Secure Authorization for Federated Environments (SAFE)
- Testbed requirements
  - Creation of dynamic VLANs and wide-area circuits
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



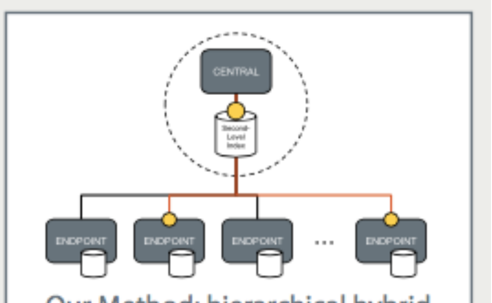
DATA SCIENCE RESEARCH

- 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 complex deployments





```
{
'header': {
        'header
        'jfif'
        'jfif_
        'exif'
        'jfif_
        'jfif_
        'dpi':
    'image_mode
    'dimensions
  'color': {
        'mean_
        'extre
        'mode_
        'median
        'std_de
   'system': {
        'path'
        'extens
        'file'
        'size'
   'image_text'
   'name_tags'
   'SVM_class_
   'mean_color
```

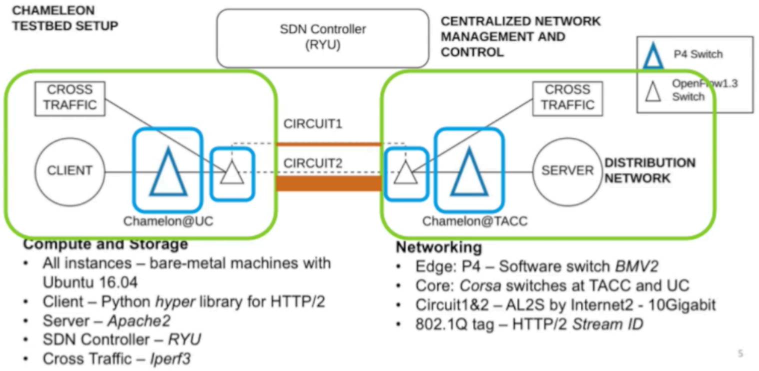


Our Method: hierarchical hybrid featuring "collapsed" second-level index (SLI)

- 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

ADAPTIVE BITRATE STREAMING

- Divyashri Bhat, UMass Amherst
- Research: application header based traffic engineering
- Testbed requirements:
  - Distributed testbed facility
  - BYOC – the ability to write an SDN controller specific to the experiment
  - Multiple connections between distributed sites



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*

BUILDING AN ECOSYSTEM

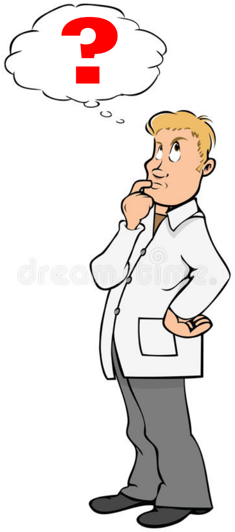
- Helping hardware providers interact
  - Bring Your Own Hardware (BYOH)
  - CHI-in-a-Box: deploy your own Chameleon
- Helping scientists interact
  - Leveraging the common denominator
  - Integrating tools for experiment management
  - Making reproducibility easier
  - Facilitating sharing

CHI-IN-A-BOX

- CHI-in-a-box: packaging a commodity-based testbed
- CHI-in-a-box scenarios
  - **Testbed extension:** join the Chameleon testbed: generalize and package + define operations models
  - **Part-time extension:** define and implement contribution models
  - **New testbed:** generalize policies
- Available since Summer 2018
- New Associate Site at Northwestern
  - Nodes with 100G network cards

REPRODUCIBILITY DILEMMA

Should I invest in making my experiments repeatable?



Should I invest in more new research instead?

- Reproducibility as side-effect: lowering the cost of repeatable research
  - Example: Linux "history" command
  - From a meandering scientific process to a recipe
- Documenting the process: interactive papers

REPEATABILITY MECHANISMS IN CHAMELEON

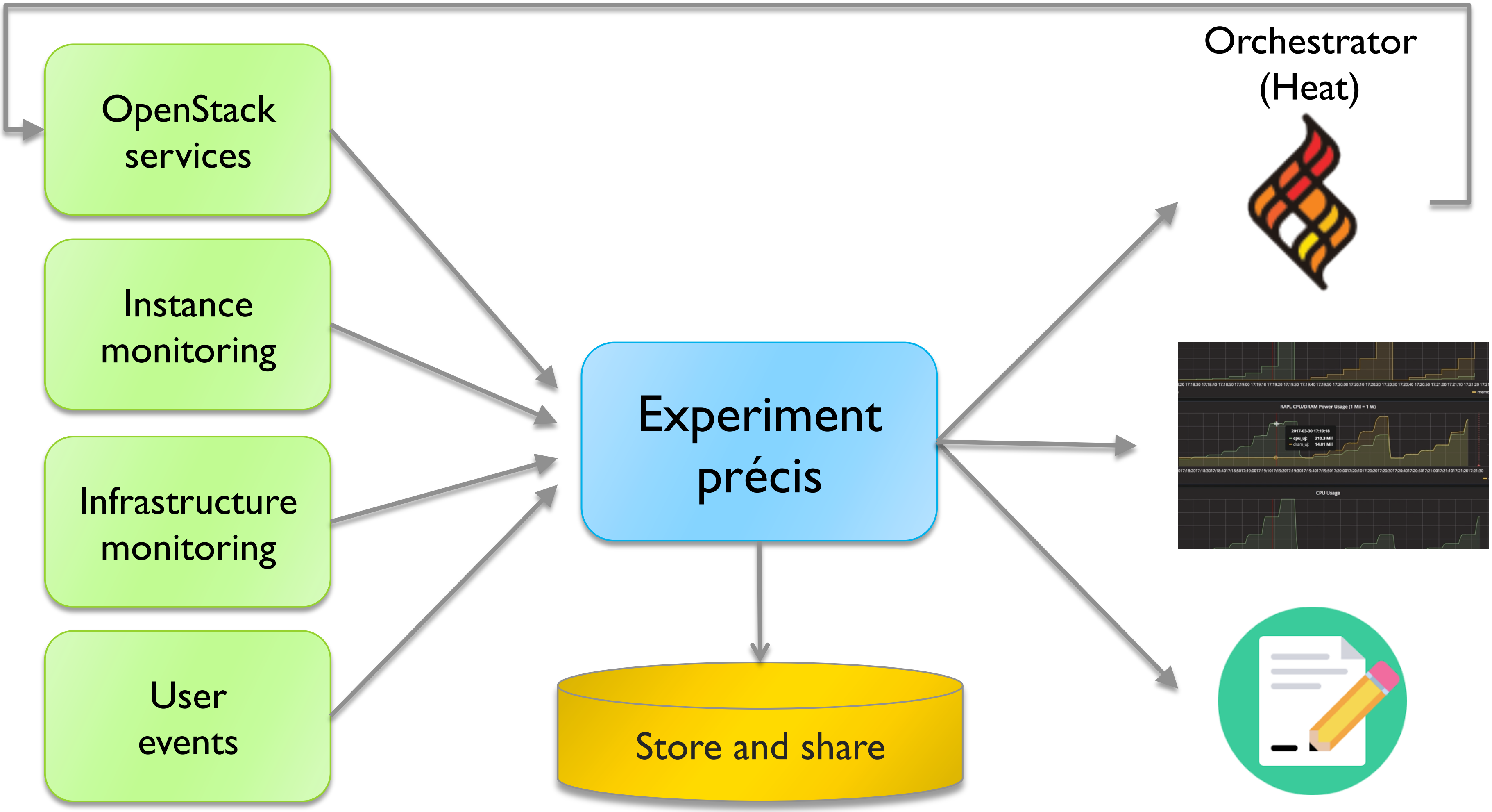
- Testbed versioning (collaboration with Grid’5000)
  - Based on representations and tools developed by G5K
  - >50 versions since public availability – and counting
  - Still working on: better firmware version management
- Appliance management
  - Configuration, versioning, publication
  - Appliance meta-data via the appliance catalog
  - Orchestration via OpenStack Heat
- Monitoring and logging
- **However… the user still has to keep track of this information**

KEEPING TRACK OF EXPERIMENTS

- Everything in a testbed is a recorded event
  - The resources you used
  - The appliance/image you deployed
  - The monitoring information your experiment generated
  - Plus any information you choose to share with us: e.g., “start power_exp_23” and “stop power_exp_23”

- **Experiment précis:** information about your experiment made available in a “consumable” form

REPEATABILITY: EXPERIMENT PRÉCIS



Orchestrator (Heat)

OpenStack services

Instance monitoring

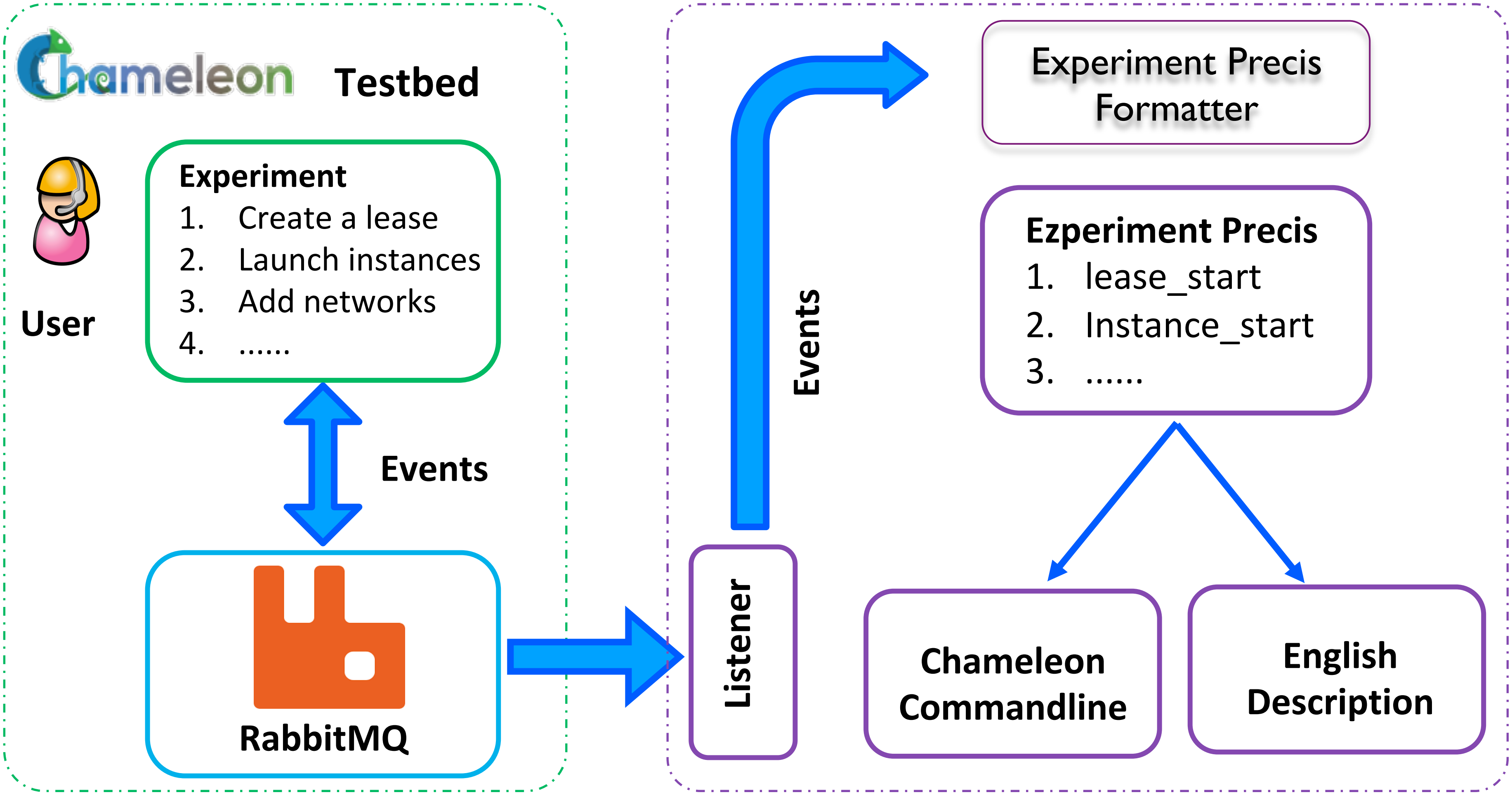
Infrastructure monitoring

User events

Experiment précis

Store and share

EXPERIMENT PRÉCIS IMPLEMENTATION



Chameleon Testbed

User

Experiment

1. Create a lease
2. Launch instances
3. Add networks
4. ……

Events

RabbitMQ

Listener

Events

Experiment Precis Formatter

Ezperiment Precis

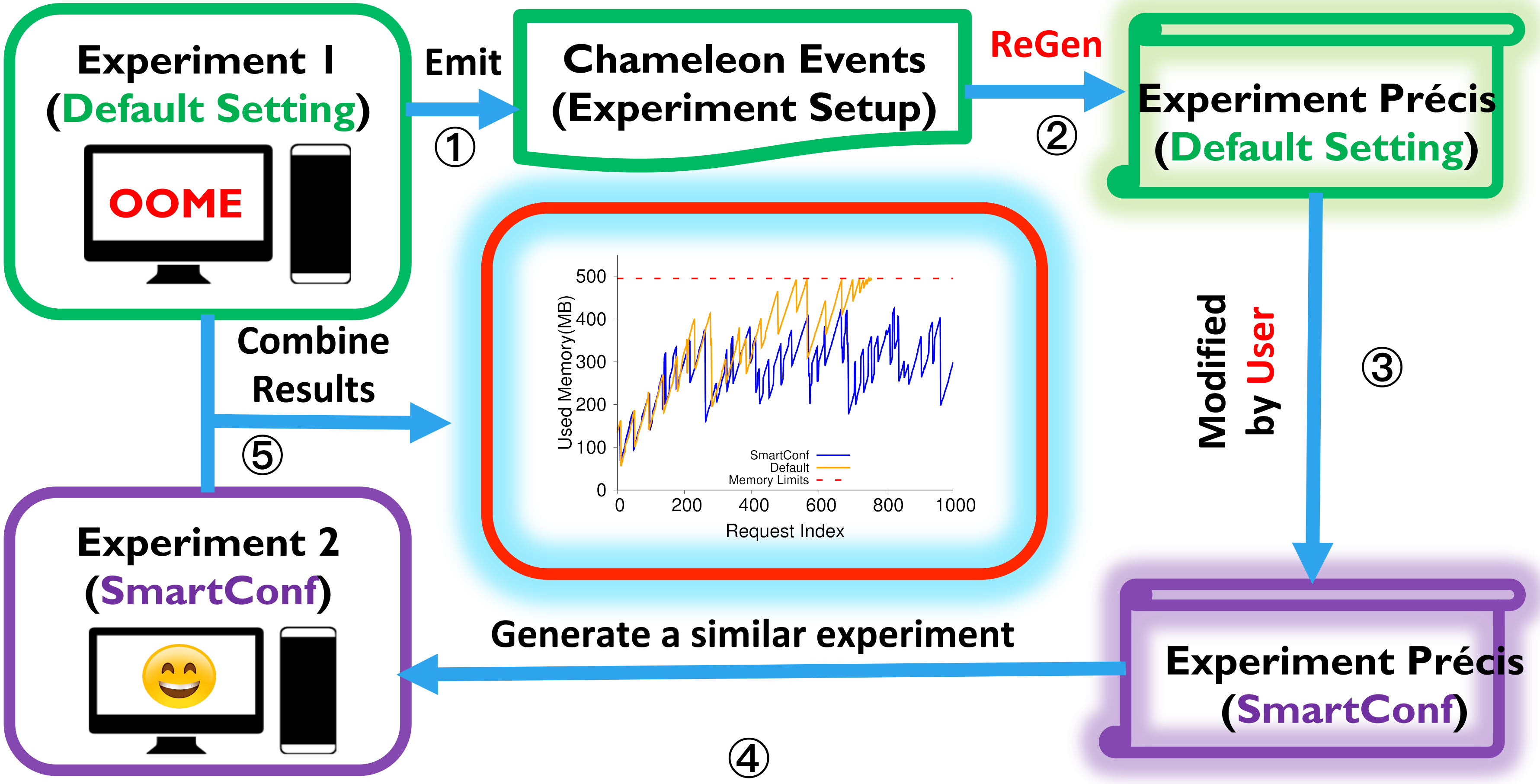
1. lease_start
2. Instance_start
3. ……

Chameleon Commandline

English Description

SC18 poster: "Reproducibility as Side-Effect"

EXPERIMENT PRÉCIS: A CASE STUDY



Experiment 1 (Default Setting)

OOME

Emit ①

Chameleon Events (Experiment Setup)

ReGen ②

Experiment Précis (Default Setting)

Modified by User ③

Experiment Précis (SmartConf)

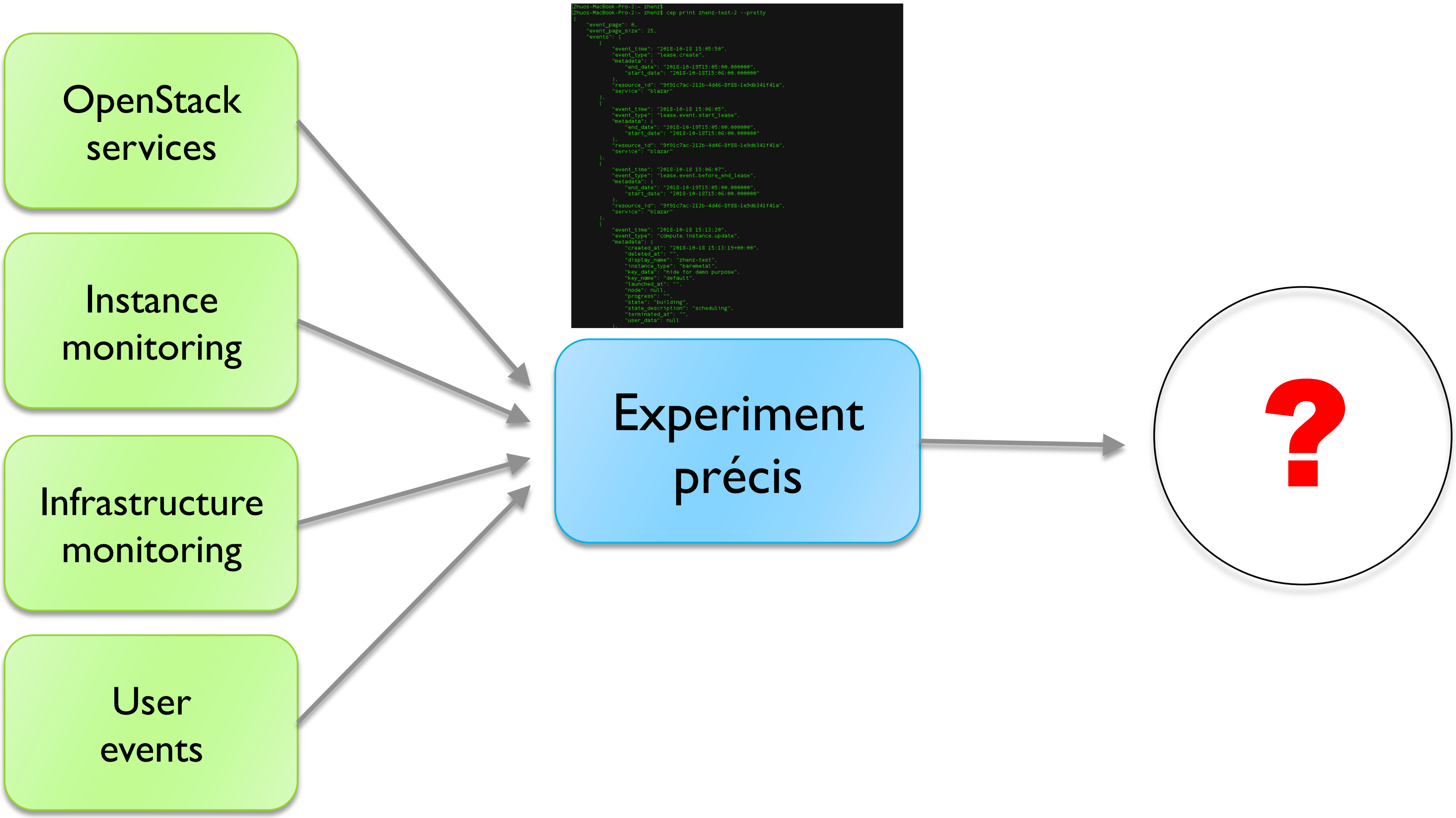
Generate a similar experiment ④

Experiment 2 (SmartConf)

Combine Results ⑤

Based on Wang et al., Understanding and Auto-Adjusting Performance-Sensitive Configurations. ASPLOS, 2018

REPEATABILITY: EXPERIMENT PRÉCIS

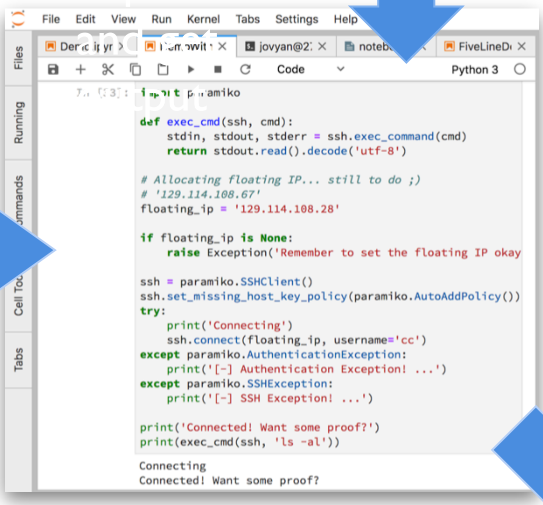


```
[Zhuos-MacBook-Pro-2:~ zhenz$
[Zhuos-MacBook-Pro-2:~ zhenz$ cep print zhenz-test-2 --pretty
{
    "event_page": 0,
    "event_page_size": 25,
    "events": [
        {
            "event_time": "2018-10-18 15:05:50",
            "event_type": "lease.create",
            "metadata": {
                "end_date": "2018-10-19T15:05:00.000000",
                "start_date": "2018-10-18T15:06:00.000000"
            },
            "resource_id": "9f91c7ac-212b-4d46-8f88-1e9db341f41a",
            "service": "blazar"
        },
        {
            "event_time": "2018-10-18 15:06:05",
            "event_type": "lease.event.start_lease",
            "metadata": {
                "end_date": "2018-10-19T15:05:00.000000",
                "start_date": "2018-10-18T15:06:00.000000"
            },
            "resource_id": "9f91c7ac-212b-4d46-8f88-1e9db341f41a",
            "service": "blazar"
        },
        {
            "event_time": "2018-10-18 15:06:07",
            "event_type": "lease.event.before_end_lease",
            "metadata": {
                "end_date": "2018-10-19T15:05:00.000000",
                "start_date": "2018-10-18T15:06:00.000000"
            },
            "resource_id": "9f91c7ac-212b-4d46-8f88-1e9db341f41a",
            "service": "blazar"
        },
        {
            "event_time": "2018-10-18 15:13:20",
            "event_type": "compute.instance.update",
            "metadata": {
                "created_at": "2018-10-18 15:13:19+00:00",
                "deleted_at": "",
                "display_name": "zhenz-test",
                "instance_type": "baremetal",
                "key_data": "hide for demo purpose",
                "key_name": "default",
                "launched_at": "",
                "node": null,
                "progress": "",
                "state": "building",
                "state_description": "scheduling",
                "terminated_at": "",
                "user_data": null
            },
```

WHAT DOES IT MEAN TO DOCUMENT A PROCESS?

- Requirements
  - Human readable/modifiable format
  - Integrates well with ALL aspects of experiment management
  - Bit by bit replay – allows for bit by bit modification (and introspection) as well – element of interactivity
  - Support story telling: allows you to explain your experiment design and methodology choices
  - Has a direct relationship to the actual paper that gets written
  - Can be version controlled and easily shared
  - Sustainable, a popular open source choice
- Implementation options
  - Orchestrators: Heat, the dashboard, and Flame
  - Notebooks: Jupyter, Nextjournal

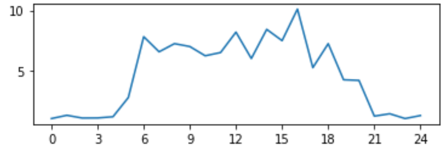
JUPYTER ON CHAMELEON



```
import paramiko

def exec_cmd(ssh, cmd):
    stdin, stdout, stderr = ssh.exec_command(cmd)
    return stdout.read().decode('utf-8')

# Allocating floating IP... still to do ;)
# '129.114.108.67'
floating_ip = '129.114.108.28'

if floating_ip is None:
    raise Exception('Remember to set the floating IP okay

ssh = paramiko.SSHClient()
ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
try:
    print('Connecting')
    ssh.connect(floating_ip, username='cc')
except paramiko.AuthenticationException:
    print('[-] Authentication Exception! ...')
except paramiko.SSHException:
    print('[-] SSH Exception! ...')

print('Connected! Want some proof?')
print(exec_cmd(ssh, 'ls -al'))
```

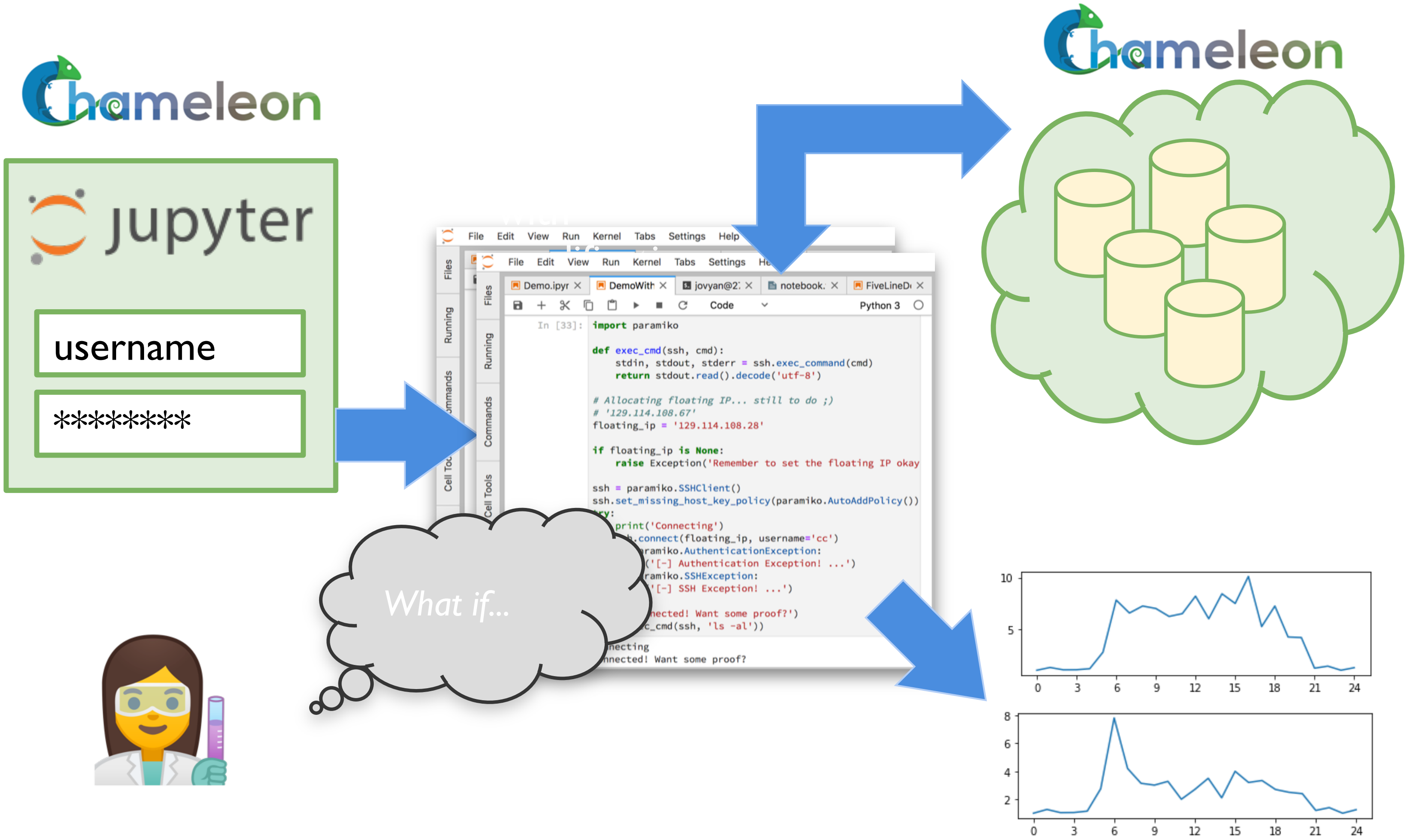
```
Connecting
Connected! Want some proof?
```

LaTeX



publishing

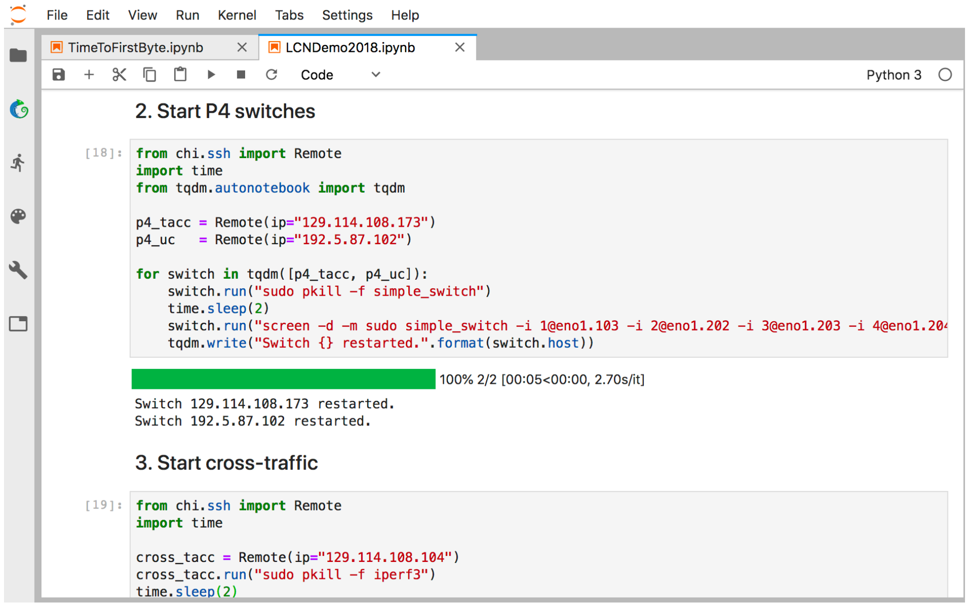
JUPYTER ON CHAMELEON



```
In [33]: import paramiko

         def exec_cmd(ssh, cmd):
             stdin, stdout, stderr = ssh.exec_command(cmd)
             return stdout.read().decode('utf-8')

         # Allocating floating IP... still to do ;)
         # '129.114.108.67'
         floating_ip = '129.114.108.28'

         if floating_ip is None:
             raise Exception('Remember to set the floating IP okay

         ssh = paramiko.SSHClient()
         ssh.set_missing_host_key_policy(paramiko.AutoAddPolicy())
         try:
             print('Connecting')
             h.connect(floating_ip, username='cc')
            aramiko.AuthenticationException:
             ('[-] Authentication Exception! ...')
            aramiko.SSHException:
             '[-] SSH Exception! ...')

             nected! Want some proof?')
             c_cmd(ssh, 'ls -al'))

         necting
         nnected! Want some proof?
```

CHAMELEON JUPYTER INTEGRATION

- Storytelling with Jupyter
  - text, process, results
- Jupyter as an interface to Chameleon
  - All the main testbed functions
  - Jupyter.chameleoncloud.org
  - “Hello World” template
  - Save and share via our object store
- Create and modify your experiment bit by bit
- Screencast of a complex experiment
  - https://vimeo.com/297210055



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@eno1.20
          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)
```

HOW DO I GET STARTED?

- Go to www.chameleoncloud.org
- Click the big orange Get started button
  - Create account
  - Create or join a project/allocation (10,000 SUs)
  - Follow the documentation to start a lease
- Keep in touch and let us know how we can help!

PARTING THOUGHTS

- Chameleon: rapidly evolving testbed
  - Changes as the research frontier changes
- Testbeds are not just experimentation platforms
  - Ecosystem: a meeting place of users sharing resources and research
  - Common/shared platform is a “common denominator” that can eliminate much complexity that goes into sharing and reproducibility: it allows you to do something interesting and powerful and then share it
- Get engaged – come to the User Meeting!
  - https://www.chameleoncloud.org/user-meeting-2019/

Questions?

www.chameleoncloud.org

keahey@anl.gov