www. chameleoncloud.org

CHAMELEON: BUILDING A RECONFIGURABLE EXPERIMENTAL TESTBED FOR LARGE-SCALE CLOUD RESEARCH

Pierre Riteau, Chameleon Lead DevOps Engineer

priteau@uchicago.edu

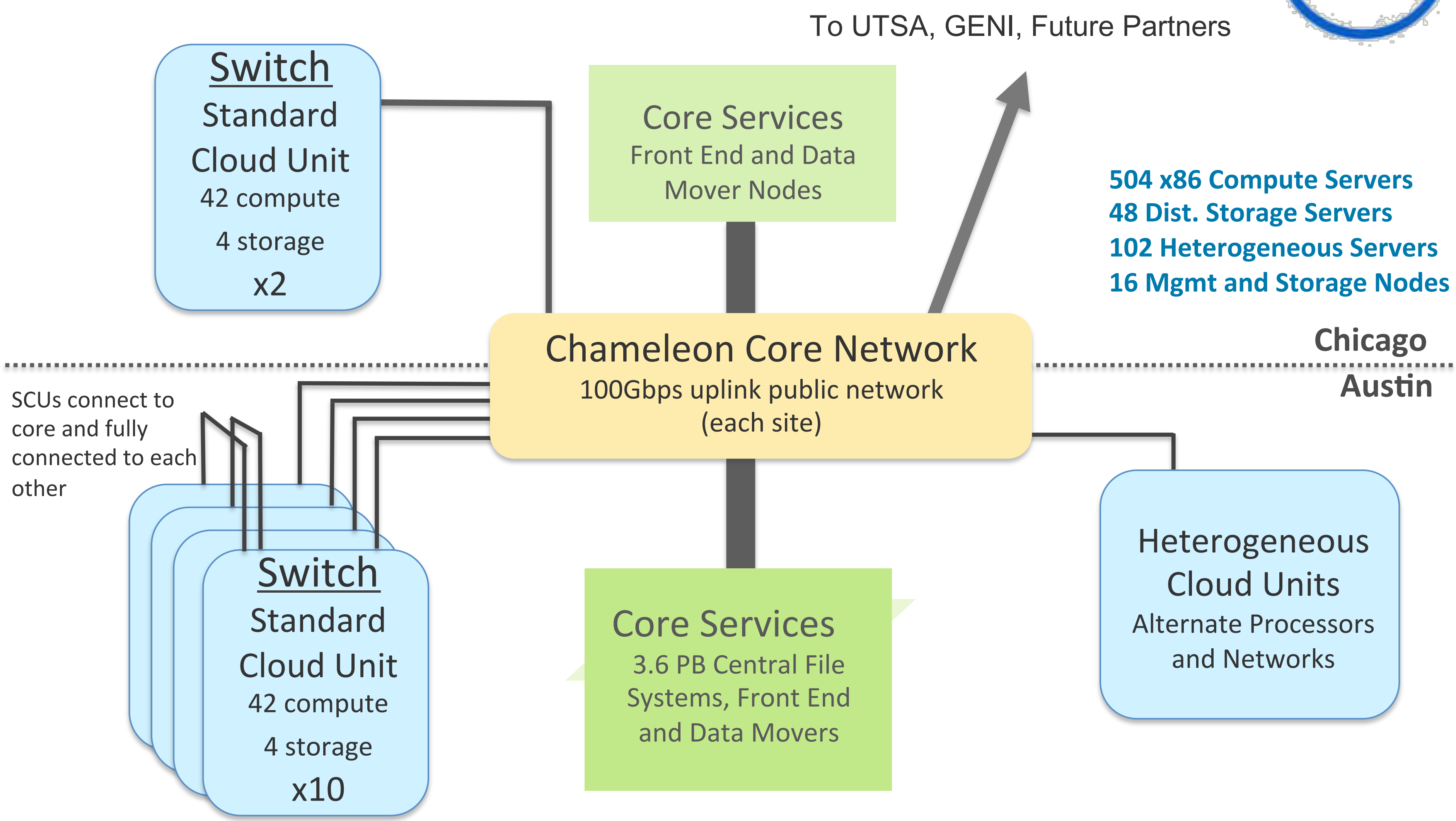
Grid'5000 Winter School 2016
February 5, 2016
Grenoble, France

TO AVOID ANY MISUNDERSTANDINGS

CHAMELEON DESIGN STRATEGY

- Large-scale: “Big Data, Big Compute, Big Instrument research”
  - ~650 nodes (~14,500 cores), 5 PB disk over two sites, 2 sites connected with 100G network
- Reconfigurable: “As close as possible to having it in your lab”
  - Bare metal reconfiguration, operated as a single instrument
  - Support for repeatable and reproducible experiments
- Connected: “One stop shopping for experimental needs”
  - Workload and Trace Archive
  - Partnerships with production clouds: CERN, OSDC, Rackspace, Google, and others
  - Partnerships with users
- Complementary: “Can’t do everything ourselves”
  - Complementing GENI, Grid’5000, and other experimental testbeds
- Sustainable: “Easy to maintain, easy to share”

CHAMELEON HARDWARE



CHAMELEON HARDWARE

- Standard Cloud Units (SCU) (deployed)
  - Each of the 12 Standard Cloud Units is a single 48U rack
  - 42 Dell R630 **compute servers**, each with dual-socket Intel Xeon (Haswell) processors (12 cores, 24 threads) and 128 GB of RAM
  - 4 Dell FX2 **storage servers**, each with a connected JBOD array of 16 2TB drives (total of 128 TB per SCU), 2 x 10 cores, and 64 GB of RAM
  - Allocations can be an entire SCU, multiple SCUs, or within a single SCU, or across SCUs (e.g., storage servers for Hadoop configurations)
  - 48 port Force10 S6000 **OpenFlow**-enabled switches 10Gb to hosts, 40Gb uplinks to Chameleon core network
  - Connectx3 **Infiniband network** in one rack at TACC
- Shared infrastructure (deployed)
  - 3.6 PB global storage, 100Gb Internet connection between sites
- Heterogeneous Cloud Units (to be procured in Y2)
  - ARM microservers, Atom microservers, SSDs, GPUs, FPGAs

CAPABILITIES AND SUPPORTED RESEARCH

Development of new models, algorithms, platforms, auto-scaling HA, etc., innovative application and educational uses

Persistent, reliable, shared clouds

Repeatable experiments in new models, algorithms, platforms, auto-scaling, high-availability, cloud federation, etc.

Isolated partition, Chameleon Appliances

Virtualization technology (e.g., SR-IOV, accelerators), systems, networking, infrastructure-level resource management, etc.

Isolated partition, full bare metal reconfiguration

IMPLEMENTING THE EXPERIMENTAL WORKFLOW

discover resources	provision resources	configure and interact	monitor
- Fine-grained - Complete - Up-to-date - Versioned - Verifiable	- Advance reservations & on-demand - Fine-grained allocations - Isolation	- Bare metal - Deeply reconfigurable - Multiple appliances to a lease - Snapshotting - Complex Appliances	- Hardware metrics - Fine-grained information - Aggregate and archive

BUILDING A TESTBED FROM SCRATCH

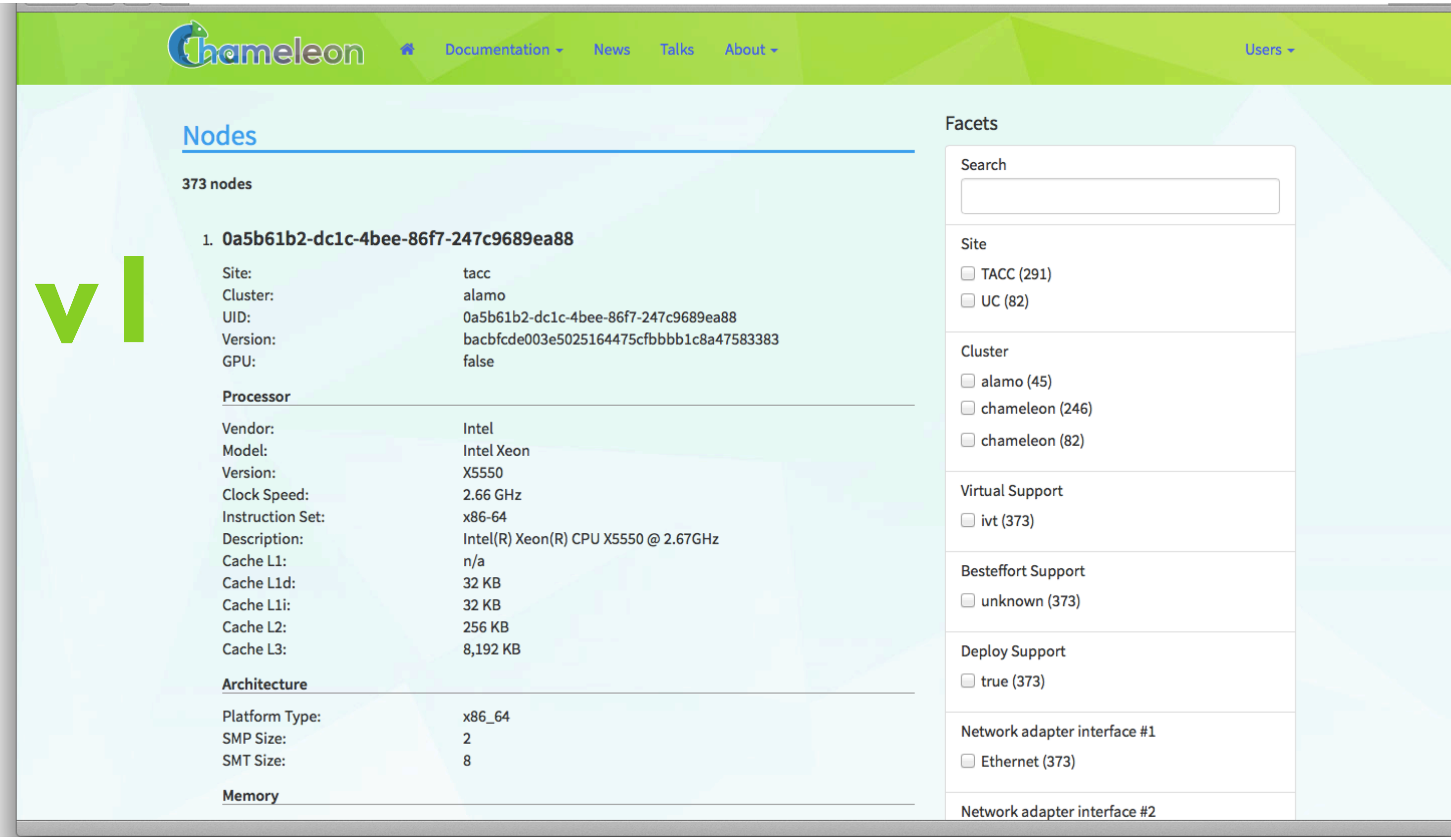
- Requirements (proposal stage)
- Architecture (project start)
- Technology Evaluation and Risk Analysis
  - Many options: G5K, Nimbus, LosF, OpenStack
  - Sustainability as design criterion: can a CS testbed be built from commodity components?
  - Technology evaluation: Grid'5000 and OpenStack
  - Architecture-based analysis and implementation proposals
- CHI = OpenStack + Grid'5000 + special sauce

CHI: DISCOVERING AND VERIFYING RESOURCES

- Fine-grained, up-to-date, and complete representation
- Both machine parsable and user friendly representations
- Testbed versioning
  - “What was the drive on the nodes I used 6 months ago?”
- Dynamically verifiable
  - Does reality correspond to description? (e.g., failure handling)

- Grid’5000 registry toolkit + Chameleon portal UI
  - Automated resource description, automated export to RM/Blazar
- g5k-checks (renamed **cc-checks** for consistency)
  - Can be run after boot, acquires information and compares it with resource catalog description

v1



Chameleon | Documentation | News | Talks | About | Users

Nodes

373 nodes

1. **0a5b61b2-dc1c-4bee-86f7-247c9689ea88**

Site:	tacc
Cluster:	alamo
UID:	0a5b61b2-dc1c-4bee-86f7-247c9689ea88
Version:	bacbfcde003e5025164475cfbbbb1c8a47583383
GPU:	false

Processor

Vendor:	Intel
Model:	Intel Xeon
Version:	X5550
Clock Speed:	2.66 GHz
Instruction Set:	x86-64
Description:	Intel(R) Xeon(R) CPU X5550 @ 2.67GHz
Cache L1:	n/a
Cache L1d:	32 KB
Cache L1i:	32 KB
Cache L2:	256 KB
Cache L3:	8,192 KB

Architecture

Platform Type:	x86_64
SMP Size:	2
SMT Size:	8

Memory

Facets

Search

Site
- TACC (291)
- UC (82)

Cluster
- alamo (45)
- chameleon (246)
- chameleon (82)

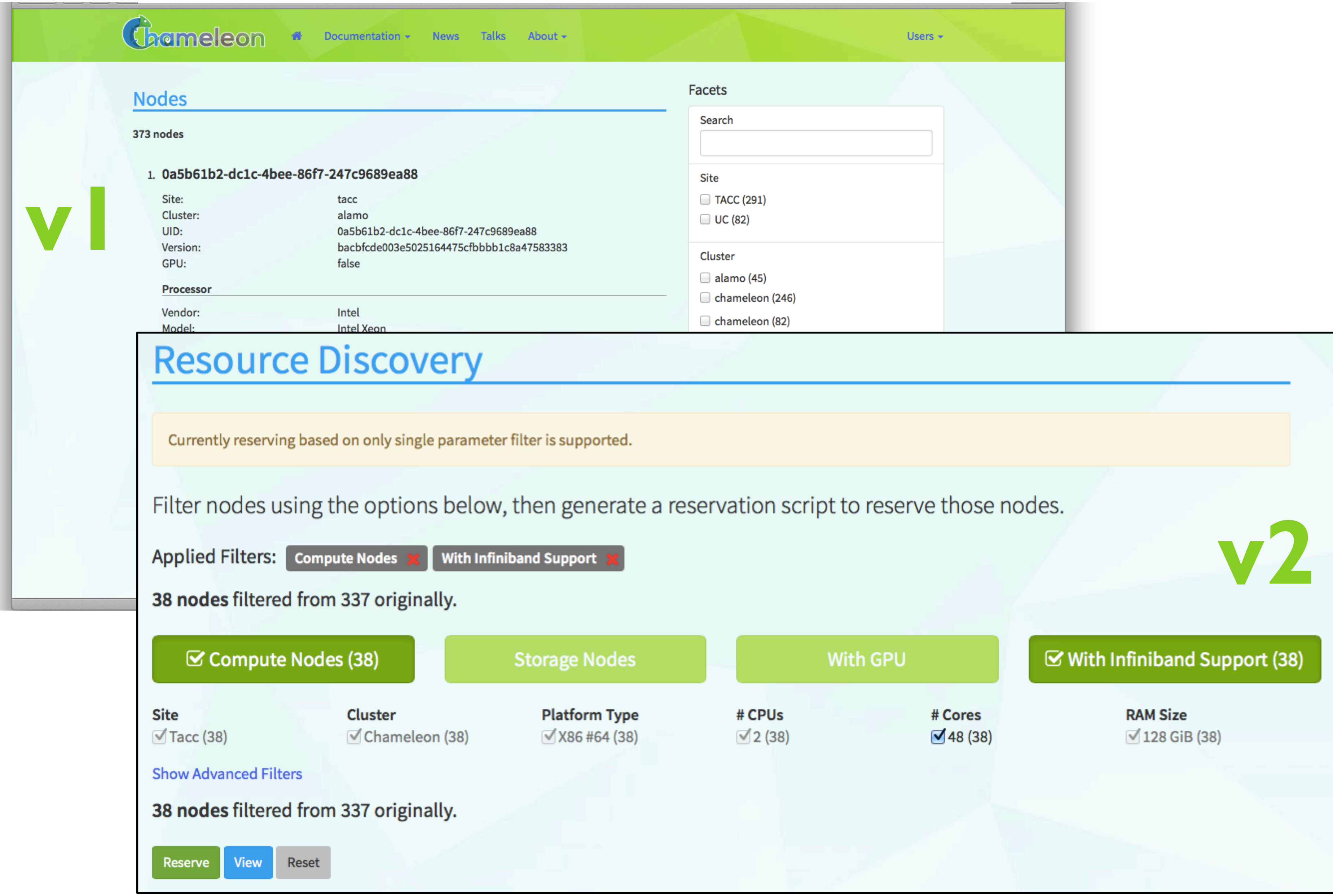
Virtual Support
- ivt (373)

Besteffort Support
- unknown (373)

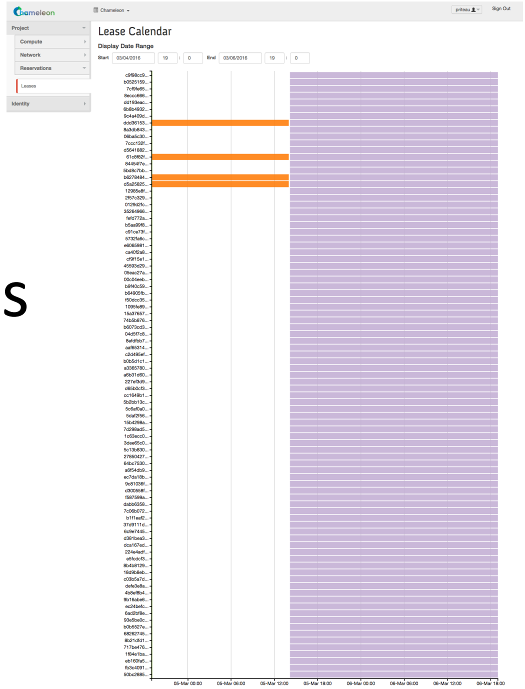
Deploy Support
- true (373)

Network adapter interface #1
- Ethernet (373)

Network adapter interface #2


Chameleon
Documentation
News
Talks
About
Users
Nodes
373 nodes
1. 0a5b61b2-dc1c-4bee-86f7-247c9689ea88
Site: tacc
Cluster: alamo
UID: 0a5b61b2-dc1c-4bee-86f7-247c9689ea88
Version: bacbfcde003e5025164475cfbbbb1c8a47583383
GPU: false
Processor
Vendor: Intel
Model: Intel Xeon
Facets
Search
Site
TACC (291)
UC (82)
Cluster
alamo (45)
chameleon (246)
chameleon (82)
v1
Resource Discovery
Currently reserving based on only single parameter filter is supported.
Filter nodes using the options below, then generate a reservation script to reserve those nodes.
Applied Filters: Compute Nodes With Infiniband Support
38 nodes filtered from 337 originally.
Compute Nodes (38)
Storage Nodes
With GPU
With Infiniband Support (38)
Site
Tacc (38)
Cluster
Chameleon (38)
Platform Type
X86 #64 (38)
CPUs
2 (38)
Cores
48 (38)
RAM Size
128 GiB (38)
Show Advanced Filters
38 nodes filtered from 337 originally.
Reserve
View
Reset
v2

CHI: PROVISIONING RESOURCES



- Resource leases
- Advance reservations (AR) and on-demand
  - AR facilitates allocating at large scale
- Fine-grain allocation of a range of resources
  - Different node types, switches, etc.
- Isolation between experiments
- Future extensions: match making, testbed allocation management

- OpenStack Nova/Blazar, contributions to Blazar
- Extensions to support Gantt chart displays and other features

CHI: CONFIGURE AND INTERACT

- Bare Metal
- Allow deep reconfigurability (access to console)
- Map multiple appliances to a lease
- Snapshotting for image sharing
- Efficient appliance deployment
- Handle complex appliances
  - Virtual clusters, cloud installations, etc.
- Interact: shape experimental conditions

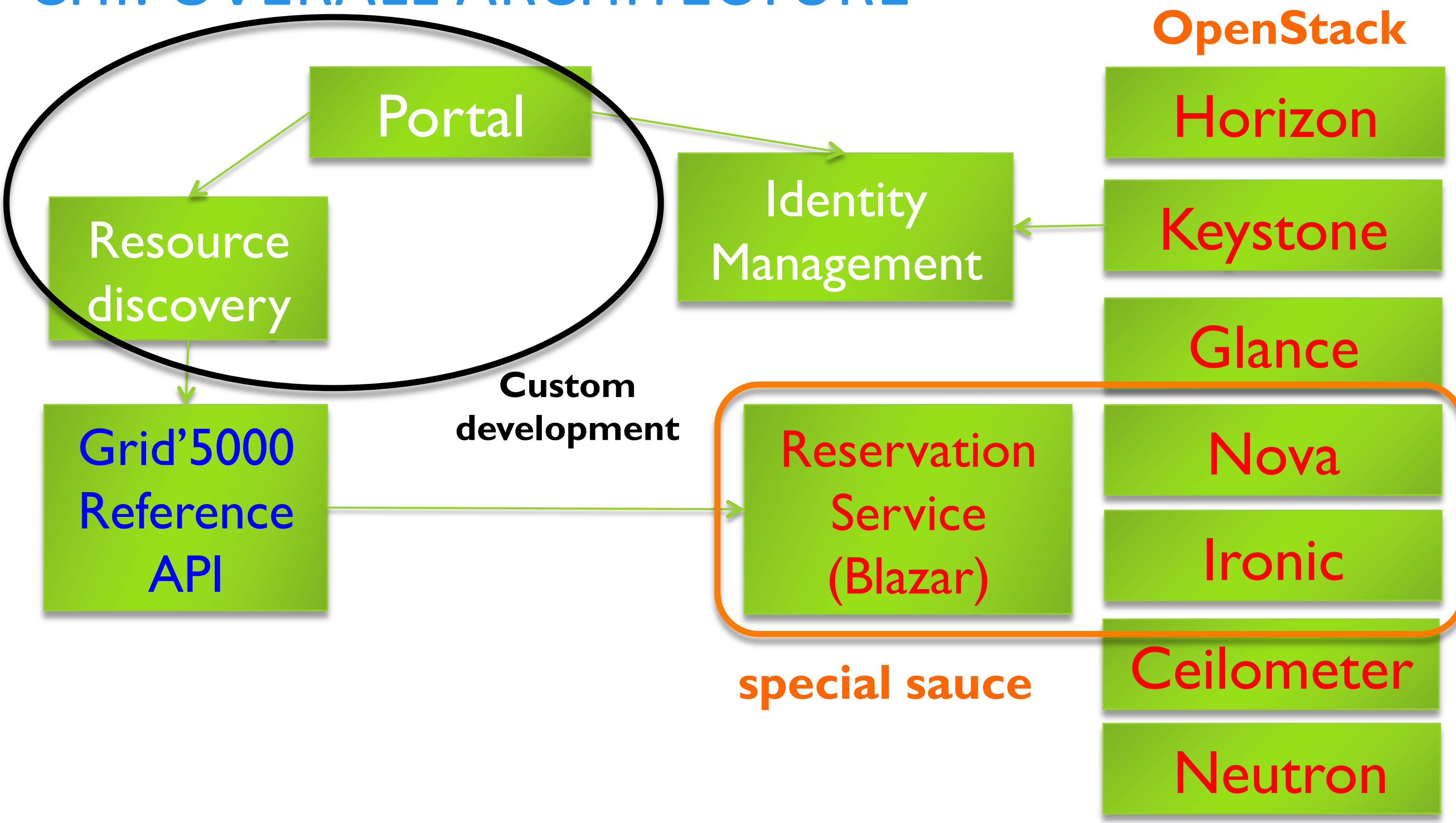
- OpenStack Ironic, Glance, and user-data / meta-data

CHI: INSTRUMENTATION AND MONITORING

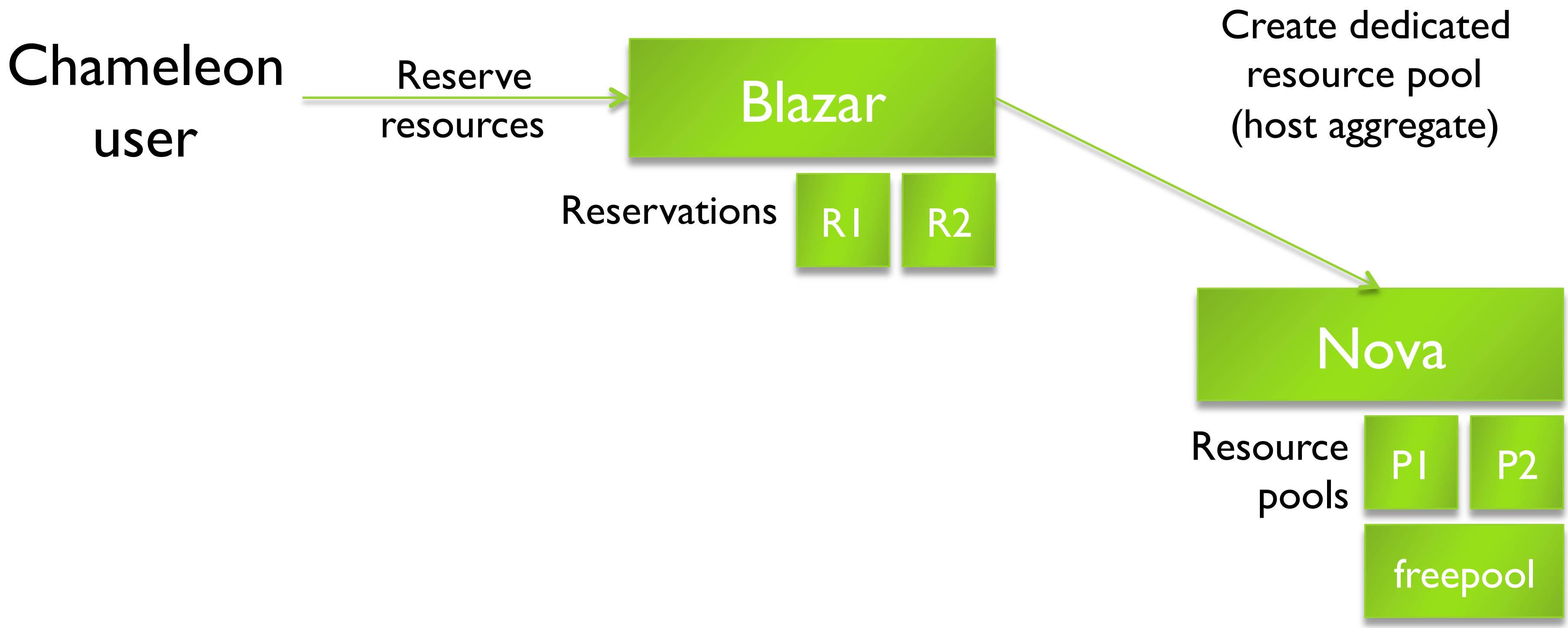
- Enables users to understand what happens during the experiment
- Instrumentation: high-resolution metrics
- Types of monitoring:
  - Infrastructure monitoring (e.g., PDUs)
  - User resource monitoring
  - Custom user metrics
- Aggregation and Archival
- Easily export data for specific experiments

- OpenStack Ceilometer + custom metrics

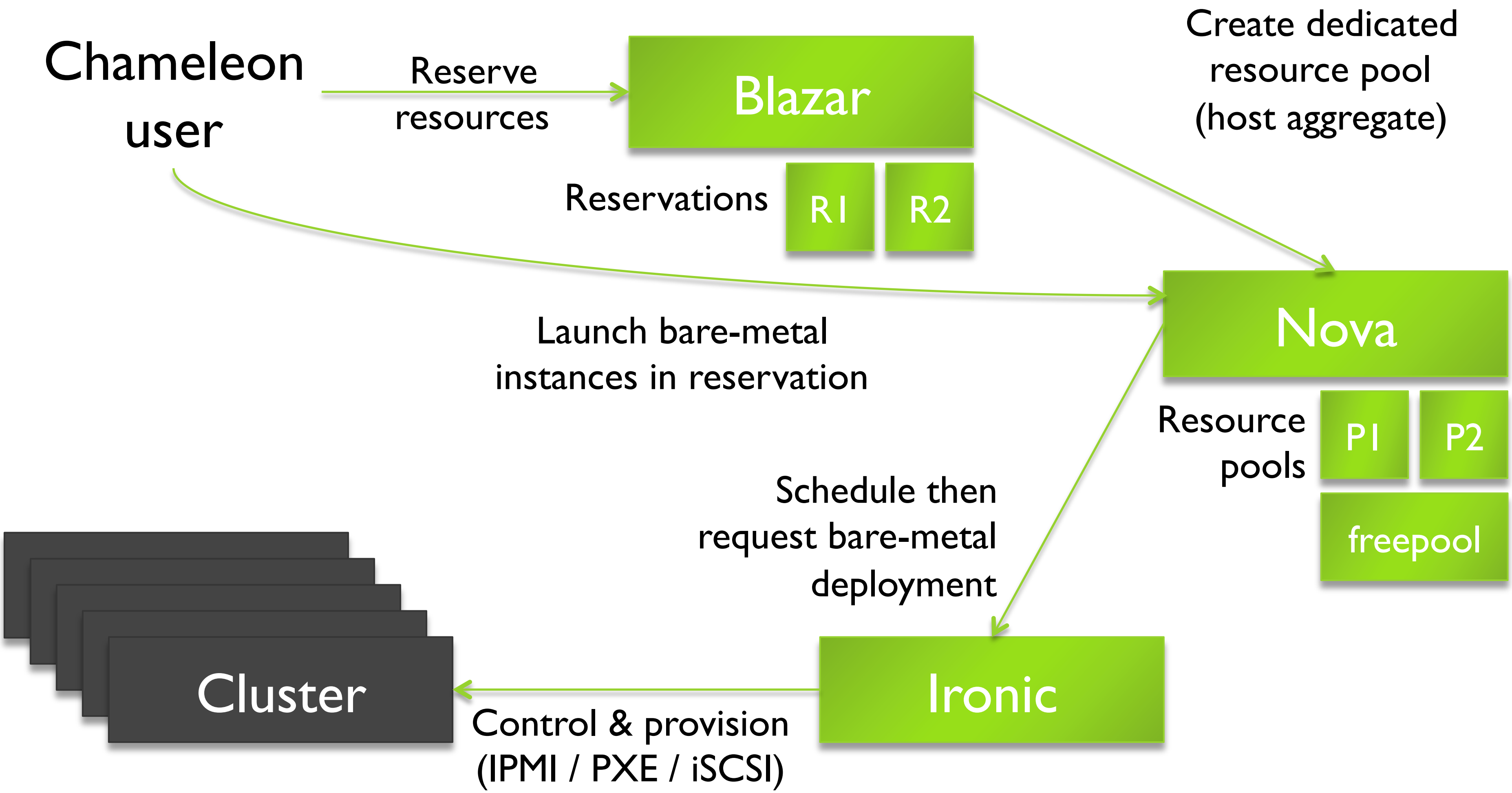
CHI: OVERALL ARCHITECTURE



HOW DOES IT WORK INTERNALLY?

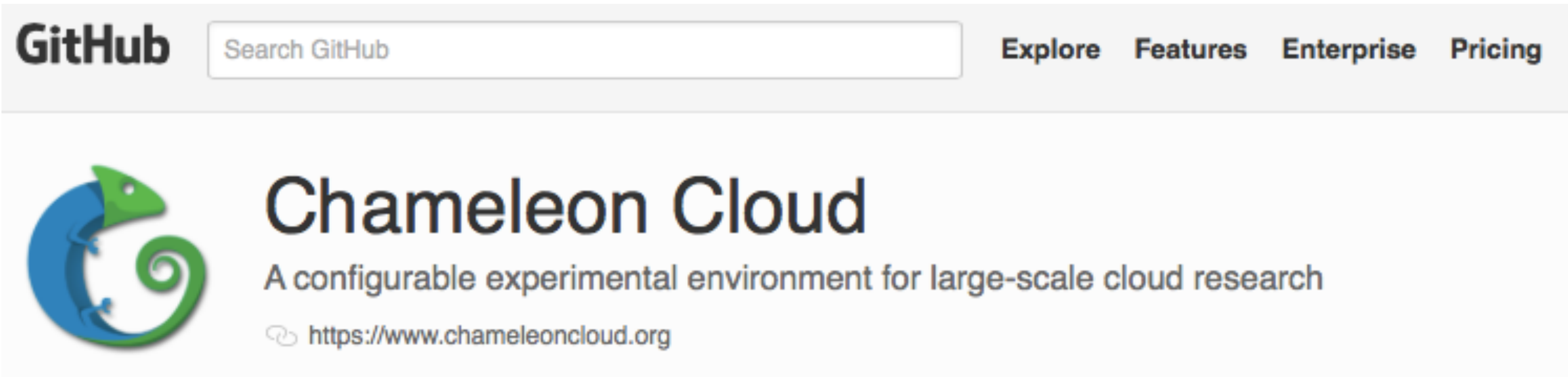


HOW DOES IT WORK INTERNALLY?



DEVELOPED IN THE OPEN

- https://github.com/ChameleonCloud



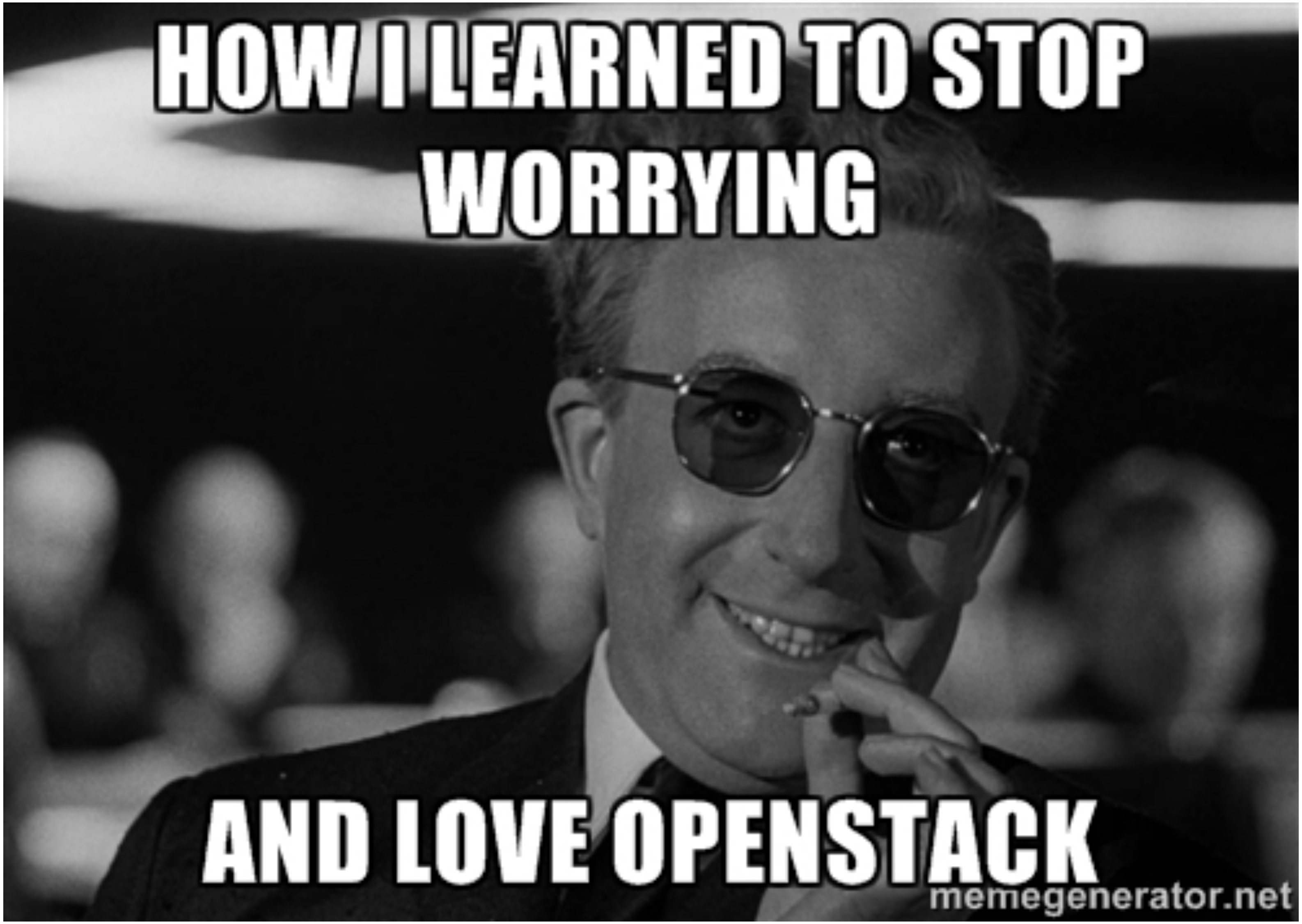
- OpenStack patches, Grid'5000 g5k-checks patches
- User portal, resource discovery, Horizon extensions
- Testbed configuration with Puppet (*not yet open*)
  - Aim is to provide a Chameleon-in-a-box!

CHAMELEON TIMELINE AND STATUS

- 10/2014: Project starts
- 12/2014: FutureGrid@Chameleon (OpenStack KVM)
- 04/2015: Chameleon Technology Preview on FutureGrid hardware
- 06/2015: Chameleon Early User on new hardware
- 07/2015: Chameleon Public availability (bare metal)
- 09/2015: Chameleon KVM OpenStack cloud available
- 10/2015: Interoperability with GENI (1st phase)
- Today: 600+ users/150+ projects
- 2016: Heterogeneous hardware available

IN THE PIPELINE…

- Y1 theme was "making things possible": focus on infrastructure
- Y2 theme is "from possible to easy": focus on users
- Outreach: webinars, tutorials, user stories
- Experiment management
  - Appliances: snapshotting, sharing, appliance marketplace, community
  - Experiment Blueprint: automation and preservation
- Functionality: from possible to easy
  - Better reconfiguration capabilities
  - Better networking capabilities
  - Better infrastructure monitoring (PDUs, etc.)
  - And others

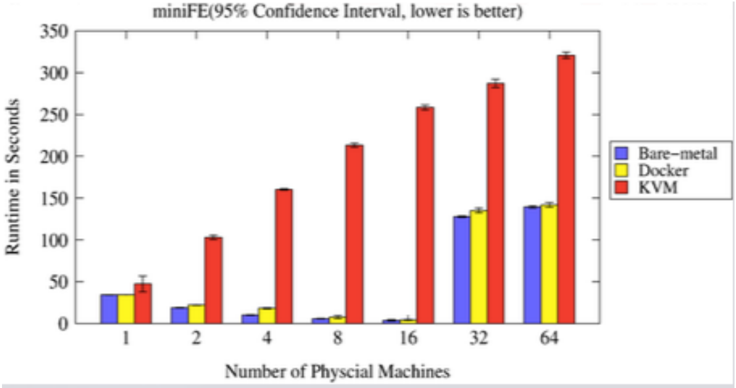

HOW I LEARNED TO STOP
WORRYING
AND LOVE OPENSTACK
memegenerator.net

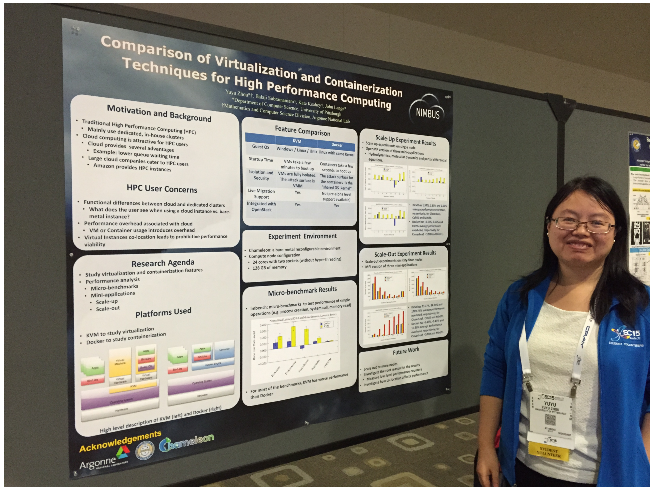
OPENSTACK: LESSONS LEARNED

- Operating OpenStack can be difficult
  - Forget about traditional UNIX admin: even bare metal needs OVS and IP namespaces
  - Thousands of configuration switches, many with little documentation
  - **Must read the code!**
  - Inter-dependent components ➔ checks all logs with debug enabled
- Upstream development mostly done on KVM
  - Less testing of Ironic ➔ bugs
- Lots of experimental projects with little upstream support
  - We were lucky as community interested in reviving Blazar
- Do not put too much hope in blueprints
  - Many abandoned or delayed for multiple releases
- Where to find help and possible fixes?
  - bugs.launchpad.net (bug reports) / review.openstack.org (patches)
  - Most developers available on IRC

VIRTUALIZATION OR CONTAINERIZATION?

- Yuyu Zhou, University of Pittsburgh
- Research: lightweight virtualization
- Testbed requirements:
  - Bare metal reconfiguration
  - Boot from custom kernel
  - Console access
  - Up-to-date hardware
  - Large scale experiments

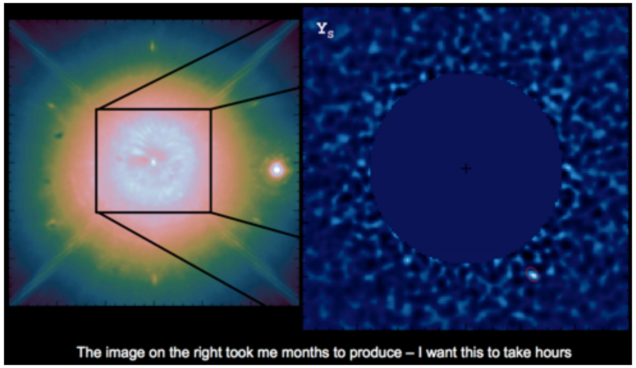


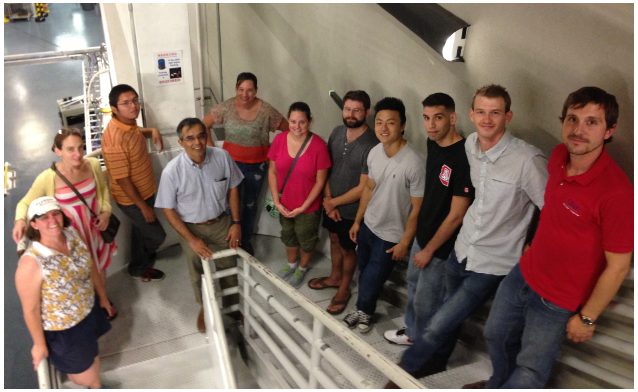


SC15 Poster: "Comparison of Virtualization and Containerization Techniques for HPC"

TEACHING CLOUD COMPUTING

- Nirav Merchant and Eric Lyons, University of Arizona
- ACIC2015: project-based learning course
  - Data mining to find exoplanets
  - Scaled analysis pipeline by Jared Males
  - Develop a VM/workflow management appliance and best practice that can be shared with broader community
- Testbed requirements:
  - Easy to use IaaS/KVM installation
  - Minimal startup time
  - Support distributed workers
  - Block store: make copies of many 100GB datasets





DEFENDING COMPUTING RESOURCES

- Led by Jessie Walker, University of Arkansas at Pine Bluff
- Working on detecting cyber attacks
  - Model and visualize multi-stage intrusion attacks (MAS)
  - Create custom Snort rules to monitor traffic and detect attacks
- Complex and expensive to buy and use their own hardware
- Limited by permissions needed to run cybersecurity attacks inside campuses
- Testbed requirements:
  - Virtual machines to simulate attacks in the cloud and run intrusion detection systems

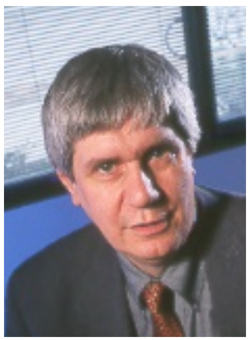
PARTING THOUGHTS

- From vision to reality with Express Delivery
  - Built from scratch within a year on a shoestring
  - Thanks to experience from other testbeds, esp. **Grid'5000**
  - Thanks to open-source code from other projects, esp. **OpenStack** and **Grid'5000**
  - Operational testbed: 600+ users/150+ projects
- Federation
  - Ongoing efforts with GENI
  - Grid'5000 too?

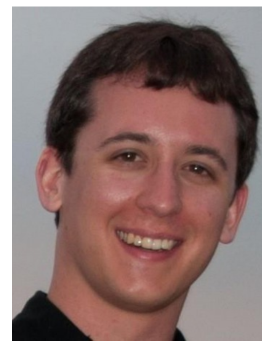
CHAMELEON TEAM

Kate Keahey
Chameleon PI
Science Director
Architect
University of Chicago

Paul Rad
Industry Liaison
Education and training
UTSA



Joe Mambretti
Programmable networks
Federation activities
Northwestern University



Pierre Riteau
DevOps Lead
University of Chicago

DK Panda
High-perf networking
Ohio State University

Dan Stanzione
Facilities Director
TACC

COME AND WORK WITH US!

- As a collaborator
  - Generalizing results: what would Kameleon or DISTEM look like in the Chameleon context?
  - Also projects in resource management for HPC&Cloud, elastic scaling platform
  - Summer internship opportunities
- As a co-worker
  - Programming postdoc or researching programmer