www. chameleoncloud.org

CHAMELEON: A DEEPLY RECONFIGURABLE, LARGE SCALE INSTRUMENT FOR COMPUTER SCIENCE EXPERIMENTATION

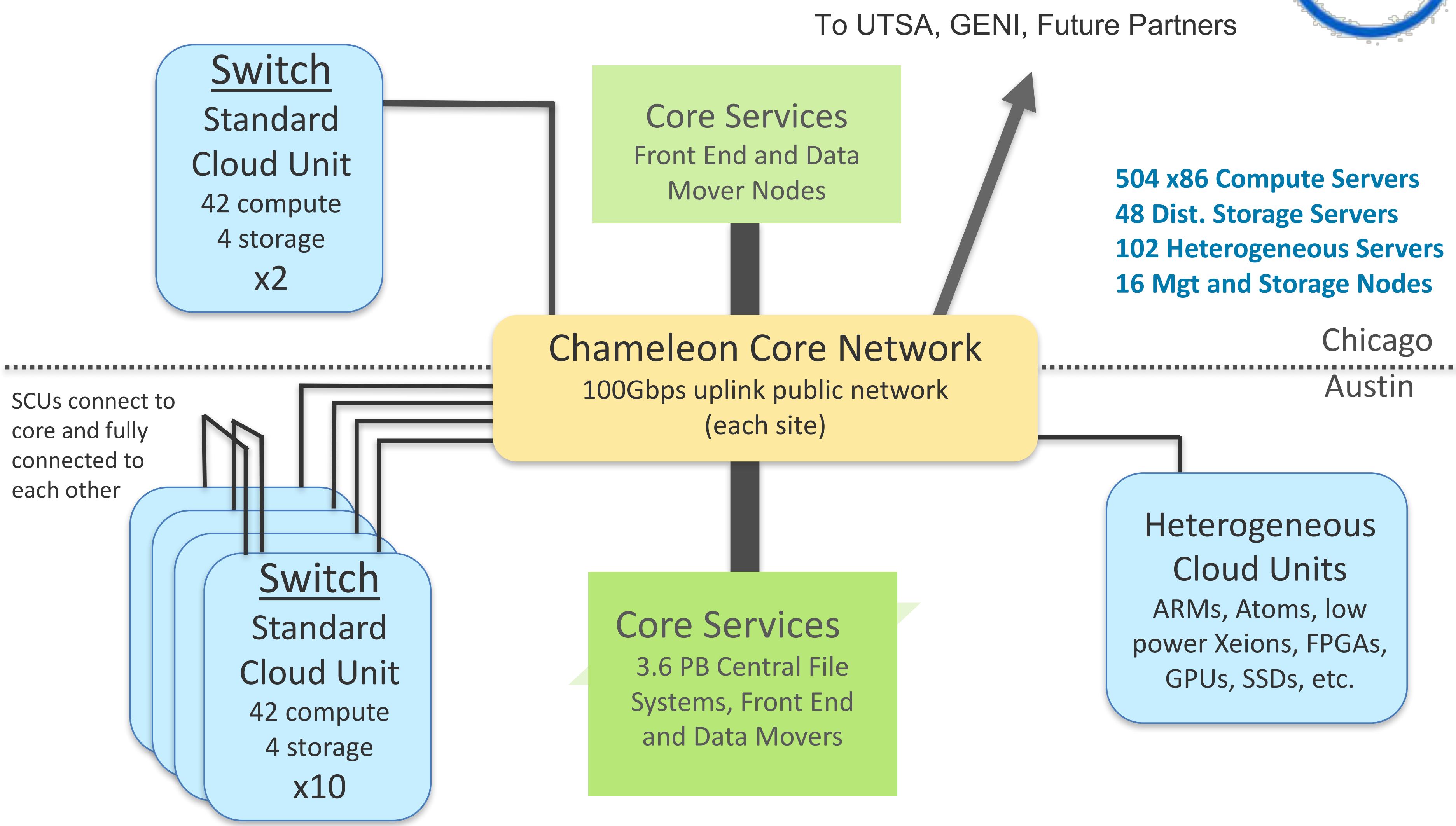
Kate Keahey
Mathematics and CS Division, Argonne National Laboratory
Computation Institute, University of Chicago
keahey@anl.gov

THE UNIVERSITY OF CHICAGO TACC NORTHWESTERN UNIVERSITY THE OHIO STATE UNIVERSITY UTSA NSF

CHAMELEON IN A NUTSHELL

- **Open** production testbed for **Computer Science Research**
  - Available since 07/2015,
  - Currently 1,600+ users, 300+ projects
- **Large-scale:** "Big Data, Big Compute research"
  - ~650 nodes (~15,000 cores), 5 PB of storage distributed over 2 sites connected with 100G network – and counting…
  - Operated as a single instrument
- **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.
- Blueprint for a **sustainable** production testbed: "cost-effective to deploy, operate, and enhance"
  - Powered by OpenStack with bare metal reconfiguration (Ironic)
- **Chameleon User Meeting,** Sept 13-14 2017
  - www.chameleoncloud.org/user-meeting-2017

CHAMELEON HARDWARE



CHAMELEON HARDWARE (DETAIL)

- "Start with large-scale homogenous partition"
  - 12 Standard Cloud Units (48 node racks)
  - Each rack has 42 Dell R630 compute servers, each with dual-socket Intel Haswell processors (24 cores) and 128GB of RAM
  - Each rack also has 4 Dell FX2 storage server (also Intel Haswells), each with a connected JBOD of 16 2TB drives (total of 128 TB per SCU)
  - 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)
  - 48 port Force10 s6000 OpenFlow-enabled switches 10Gb to hosts, 40Gb uplinks to Chameleon core network
- Shared infrastructure
  - 3.6 PB global storage, 100Gb Internet connection between sites
- "Graft on heterogeneous features"
  - Infiniband with SR-IOV support netw in one rack
  - High-memory, NVMe, SSDs, GPUs (18 nodes), FPGAs (4 nodes)
  - ARM microservers (24) and Atom microservers (8), low-power Xeons (8)

TESTBED CAPABILITIES

discover resources	provision resources	configure and interact	monitor
- Fine-grained - Complete - Up-to-date - Versioned - Verifiable	- Advance reservations & on-demand - Isolation - Fine-grained allocations	- Deeply reconfigurable - Appliance catalog - Snapshotting - Complex Appliances - Network Isolation	- Hardware metrics - Fine-grained information - Aggregate and archive

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

SOLUTION DISCOVERY AND EVALUATION

- 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 three implementation proposals managing risk areas
- Implementation of core capabilities (~3 months)
- Continuous discovery: users with fantastic solutions!

CHANGING AVAILABILITY OF CI

- Constant **risk/cost** versus **reward** evaluation
- Example: Nimbus/CloudStack/OpenStack
- Chameleon example: OpenStack upgrades
  - Reward: bug fixes, new features, lesser technical debt, etc.
  - Risk/cost: cost of change versus opportunity cost
- Motivators for change: access to serial console, multi-tenant networking, storage volume management
- Cost of change: customized upgrade and new feature evaluation

INCREASED REUSE AND AWARENESS OF CI -> INCREASED INTEROPERABILITY ACROSS FACILITIES

- Software familiarity: trained personnel, trained users, familiar model/interfaces
- Portability of images and deployment scripts (with small adaptations)
  - Example: experimenting on Chameleon -> production deployment on Roger (OpenStack cloud at NCSA)
  - Similar for Jetstream and other OpenStack couds
- Interoperability of solutions
  - Chameleon: integrating ExoGENI's implementation of a slice to provide interoperability with GENI

CAN COMMUNITY EFFORTS IN INTEGRATION, INTEROPERABILITY AND SUSTAINABILITY LEAD TO WELL DEFINED INTERFACES THAT FACILITATE ACCESS TO AND INCORPORATION OF NEW TECHNOLOGIES?

- Yes …and no
- Well-defined interfaces will help integrate solutions compatible with them
- …while at the same time preventing (or making it very hard or convoluted) to integrate groundbreaking solutions
- Sensitive to a moment in technology lifecycle

WHAT ARE THE MOST CRITICAL CI GAPS THAT YOU WOULD LIKE TO BE ADDRESSED?

- Experimental Science Universe
  - "active papers", support for repeatability and reproducibility, management of provenance, etc.
  - Chameleon: testbed versioning, appliances, experiment summaries (precis), replay mechanisms
- Software and data publication and sharing
  - Critical for narrowing the options
  - SoftwareX, JOSS, etc.
- Mechanisms to combine experimental infrastructure, cloud, and HPC
  - Currently silos: different support hard to explain to lay people
  - Can we combine these configuration methods?