www. chameleoncloud.org

MANAGING ALLOCATABLE RESOURCES

Kate Keahey, Pierre Riteau, Jason Anderson, and Zhuo Zhen

Argonne National Laboratory, University of Chicago

keahey@anl.gov

Cloud 2019
July 11, 2019

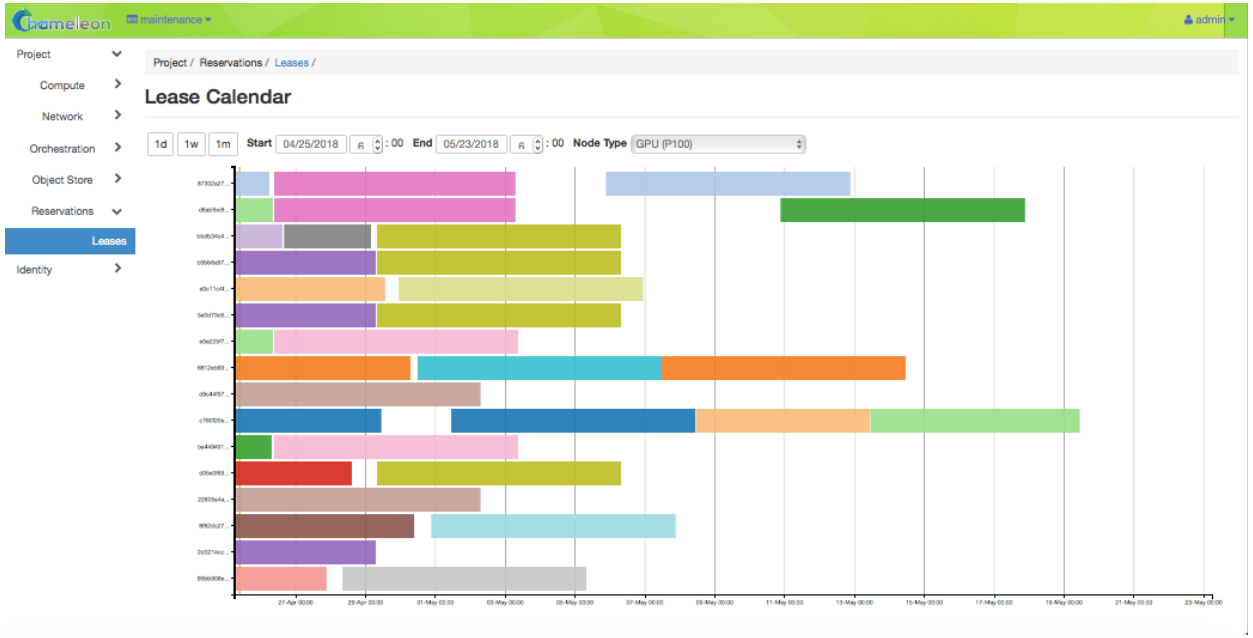
CHAMELEON IN A NUTSHELL

- We like to change: testbed that adapts itself to your experimental needs
  - Deep reconfigurability (bare metal) and isolation (CHI) – but also ease of use (KVM)
  - CHI: power on/off, reboot, custom kernel, serial console access, etc.
- We want to be all things to all people: balancing large-scale and diverse
  - Large-scale: ~large homogenous partition (~15,000 cores), 5 PB of storage distributed over 2 sites (now +1!) connected with 100G network…
  - …and diverse: ARMs, Atoms, FPGAs, GPUs, Corsa switches, etc.
- Cloud on cloud: leveraging mainstream cloud technologies
  - Powered by OpenStack with bare metal reconfiguration (Ironic) + "special sauce"
  - Chameleon team contribution recognized as official OpenStack component
- We live to serve: open, production testbed for Computer Science Research
  - Started in 10/2014, testbed available since 07/2015, renewed in 10/2017
  - Currently 3,000+ users, 500+ projects, 100+ institutions

ALLOCATABLE RESOURCES

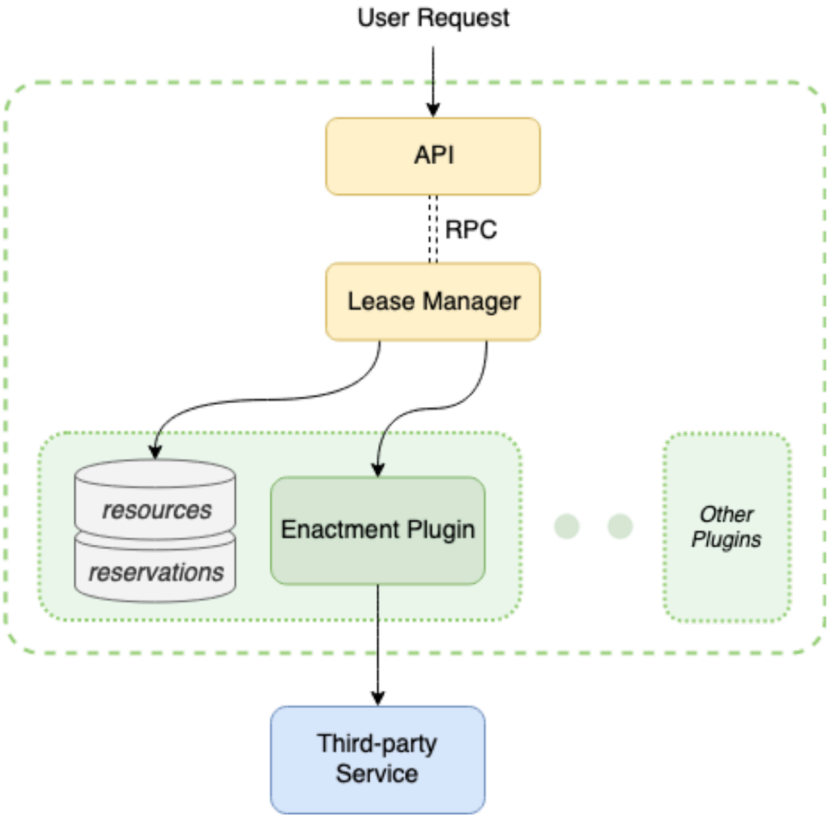
- **Definition:** object within a system that can be automatically allocated and managed for exclusive, metered usage, delimited by well- defined time events
- Exclusive usage/isolation:
  - System versus performance isolation
  - Examples: GENI slice, containers, virtual machines, instances on commercial clouds, physical modes
  - Implementation costs: e.g., hypervisor overhead, or default state recovery
- Time-bounded: e.g., advance reservation (including on-demand)
- Metered: policy enforcement and incentive management (via monetary or non-monetary means)
- Automated lease management: extends over both time and resource types; ensures scalability

NON-FUNGIBLE ALLOCATABLE RESOURCE AVAILABILITY REPRESENTATION



ARCHITECTURE

- Components
  - Service interfaces
  - Lease Manager (functionality generic to all leases)
  - Enactment Plugins (resource-specific functionality)
- Third-party Services
  - Adaptation of non-allocatable resources
  - Examples: Nova, Neutron



SERVICE INTERFACES

- Inventory management (operators)
  - Objective: manage resource database
  - Create, update, delete manage resource database
  - Informational (show and list)
- Lease management (resource clients)
  - Objectives: create and manage records in lease database
  - Leases versus reservations
  - Create and delete: range from a very partial description of a resource, (e.g., “two nodes on the same rack”, “node with at least 2GB of memory”) to very specific (“node X”) and can include multiple resources (e.g., nodes, IPs, VLANs)
  - Update (active or inactive) lease: change resources, numbers, or temporal constraints
  - Informational (show and list)

LEASE MANAGER

- Interface to lease database
  - Information persisted includes original constraints
- Early or late assignment
  - Simplicity versus efficiency
  - Resolved at activation time at the latest
  - Selection targets ranging from user choice to operation optimization
- Event-based assignment management
  - E.g., reassignment in case of resource failure

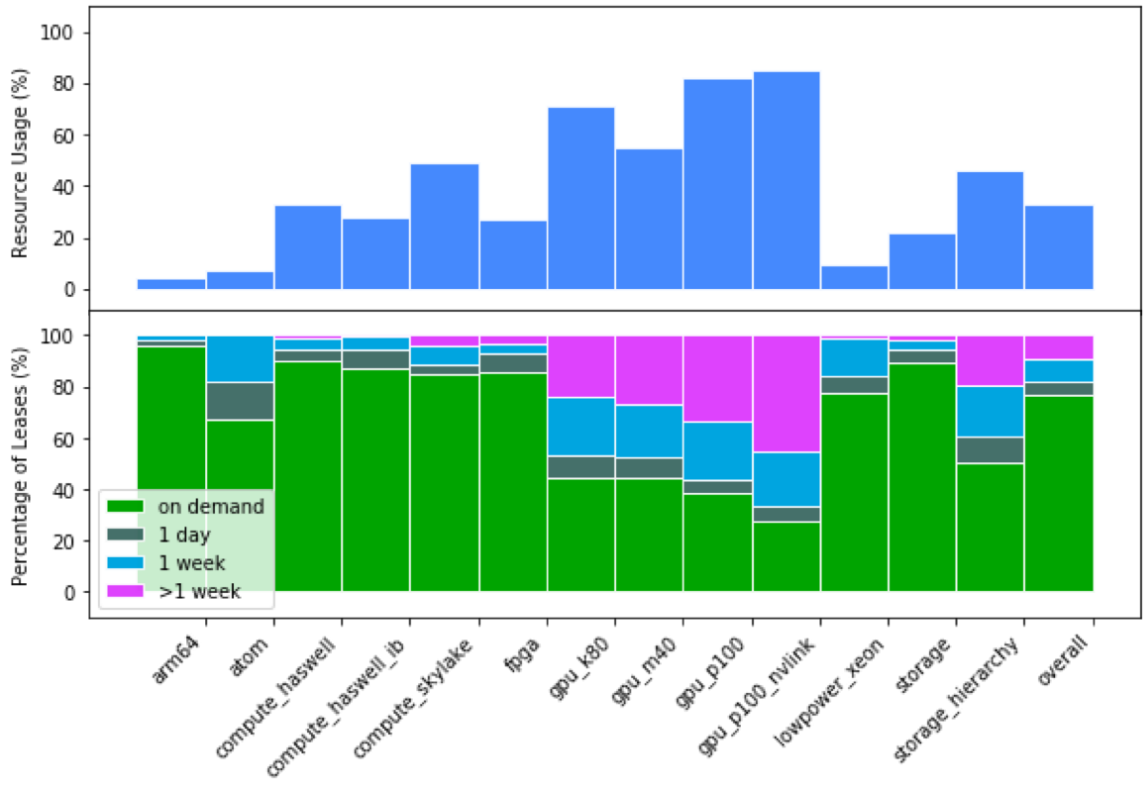
RESOURCE PLUGINS

- Allow lease manager to manage diverse resources
  - Inventory management: separate reservable resources from freely available resources
  - Lease management: allocating and deallocating resources to a lease
- Lease management functions
  - on_start: resource allocation
  - on_end: resource deallocation
  - before_end: trigger an action at a configurable time before the end of a reservation (e.g., snapshot instances)
  - update_reservation: e.g., add nodes to a lease

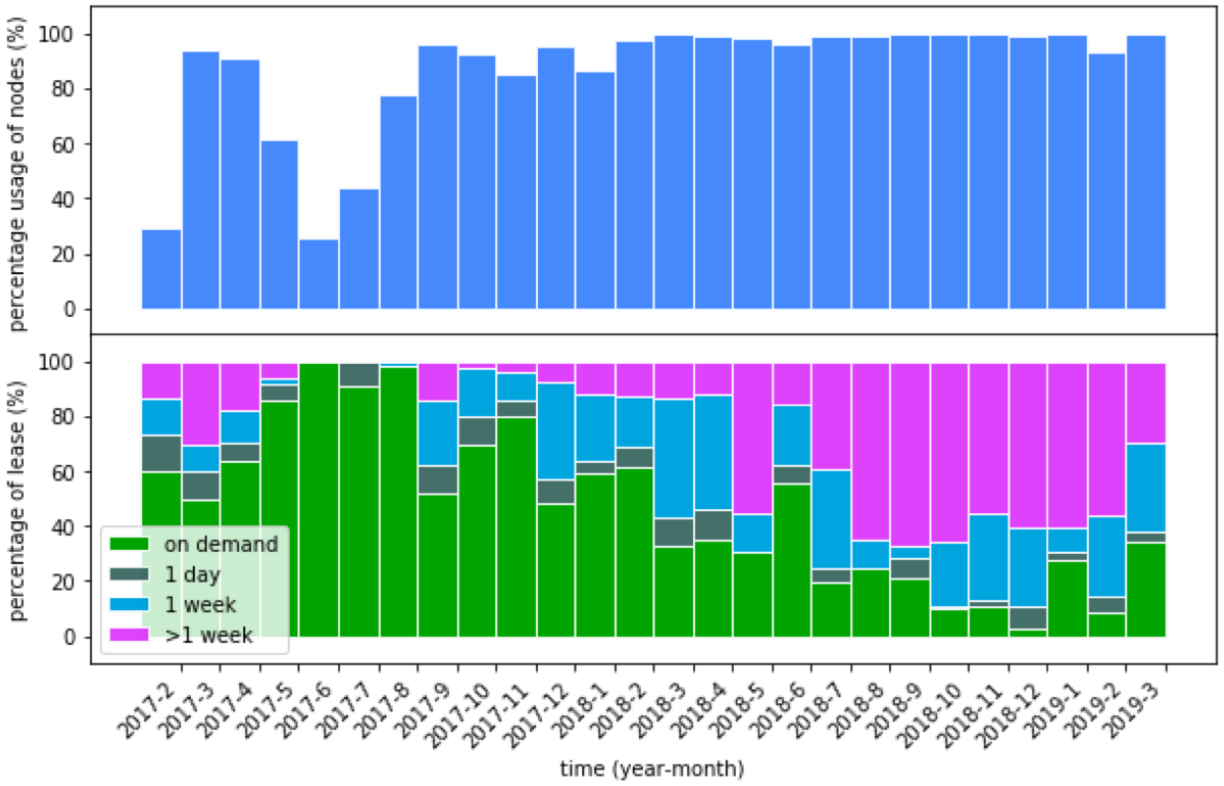
IMPLEMENTATION

- Allocatable resources in Chameleon
  - Heterogeneous bare metal machines
  - Isolated network segments (VLANs)
  - Public IP addresses
- OpenStack Blazar implementation with plugins for each
  - HTTP/JSON (Keystone for authentication), SQL Alchemy, early binding with optimizations
- Bare-metal nodes: system and performance isolation, implemented as OpenStack Nova/Ironic plugin
- VLANs: system isolation, implemented as OpenStack Neutron plugin
- IP addresses: implemented as OpenStack Neutron plugin

ALLOCATABLE RESOURCE USAGE ON CHAMELEON

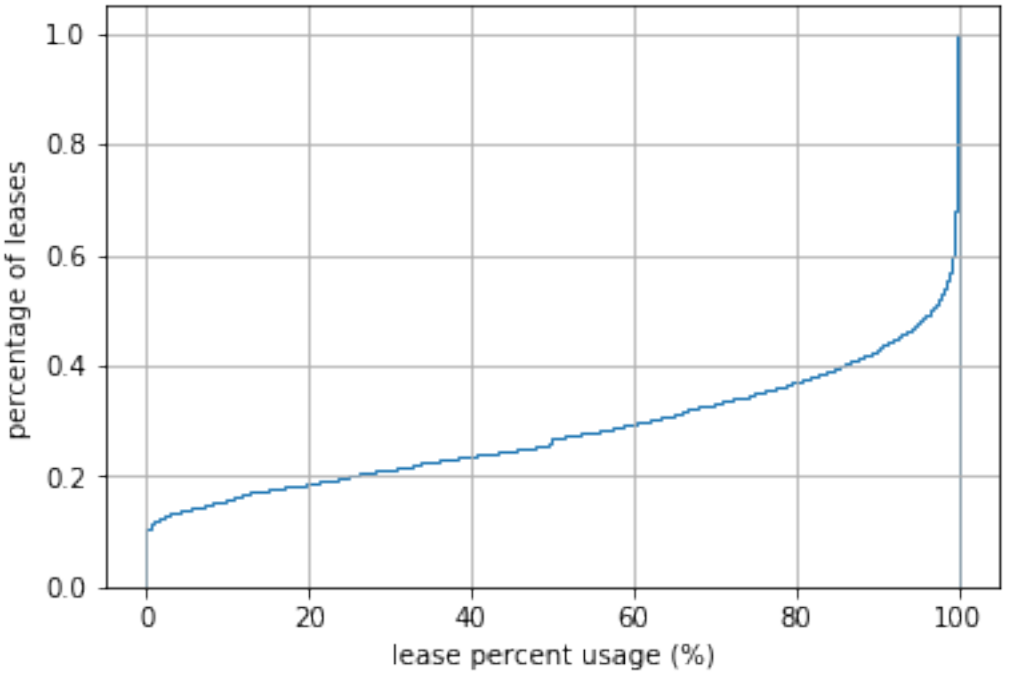


ALLOCATABLE RESOURCE USAGE OVER TIME



HOW RESPONSIBLY ARE RESERVATIONS USED

- Fully used: ~40%
- 80% used: ~60%
- 20% used: ~ 20%
- Not used: ~15%
- Management
  - Idle at beginning or end: ~5% each
  - ~20% early release



CONCLUSIONS

- Articulated resource management service
- Implemented in OpenStack Blazar
  - Collaboration with OpenStack Blazar community
  - Can be used independently of OpenStack – anticipate future work with IoT devices
- Effective tool in the management of scarce resources
- Managing user incentives
- Come and give it a shot! www.chameleoncloud.org