Resource Reservation with OpenStack Blazar

Pierre Riteau (StackHPC)
pierre@stackhpc.com

StackHPC

Resource sharing is hard

- “I need 100 VMs next week for project X, but our cloud is almost full. Can anyone terminate some non-critical instances?”
- “We have a few nodes with interesting hardware (GPUs, FPGAs), how can users take turn using it?”
- “How can I make sure resources will be available at the same time I am experimenting on instrument X?”

- Existing solutions such as quotas are limited
  - Admin-defined
  - Hard to dynamically adapt to changing user requirements
  - No concept of time: quota applies from now until next quota update

What is Blazar?

- **Blazar provides Reservation as a Service**
- Created in 2013 under the name Climate
- Revived at the Barcelona Summit in 2016
- Official project since September 2017 (Queens release cycle)
- Supports reservation of:
  - compute resources: whole hosts and individual instances
  - network resources: floating IPs, network segments in the next release
- Guarantees resource availability for a future event
- 21 commit authors and 30+ reviewers in the OpenStack Stein release cycle

Use cases

- Chameleon project (http://www.chameleoncloud.org)
  - Large-scale testbed for computer science research
  - Blazar guarantees resource availability for experiments

- OPNFV Promise project (https://wiki.opnfv.org/display/promise/Promise)
  - Resource reservation and management project for quality of network services
  - Blazar guarantees resource availability for VNF deployments

How does Blazar work?

Terminology

- *Reservation:* allocation of specific cloud resources, of the same resource type
- *Lease*: group of reservations granted to a project for a specific time period. Leases are characterized by start time, end time, set of individual reservations and associated events.

Supported reservation types

- Physical host reservation: allocates one or several physical hosts (hypervisors) to each reservation. Users can then launch instances on the reserved hosts until full capacity is reached. A host **cannot** be used by multiple reservations at the same time.
- Instance reservation: allocates enough capacity on reservable hosts to launch a number of instances. A host **can** be used by multiple reservations at the same time.
- Floating IP reservation: allocates one or several floating IPs for the duration of the reservation.

How does Blazar work?

Reservation of compute resources

- Admins add compute host(s) to be managed by Blazar:

`$ blazar host-create test_host` -> A new Resource Provider is created in Placement

```
$ blazar host-list
+----+---------------------+-------+-----------+----------+
| id | hypervisor_hostname | vcpus | memory_mb | local_gb |
+----+---------------------+-------+-----------+----------+
| 1  | test_host           |     4 |      8191 |       79 |
+----+---------------------+-------+-----------+----------+
$ openstack resource provider list
+--------------------------------------+------------------+------------+
| uuid                                 | name             | generation |
+--------------------------------------+------------------+------------+
| 4863b5a1-aeae-47ea-98a9-57af2d93be00 | test_host        |          3 |
| d7d2bdb1-ef62-4e8d-ac0e-0074640d0ea5 | blazar_test_host |          0 |
+--------------------------------------+------------------+------------+
```

How does Blazar work?

Instance reservation

- Reserve a number of instances with specific flavor sizing:

```
$ blazar lease-create --reservation resource_type=virtual:instance
,vcpus=1,memory_mb=1024,disk_gb=20
,amount=2,affinity=True test_lease
```

-> Blazar creates a custom flavor for users to launch instances

```
$ blazar lease-list
+------+-----------+----------------------------+----------------------------+
| id   | name      | start_date                 | end_date                   |
+------+-----------+----------------------------+----------------------------+
| xxxx | test_lease | 2019-05-07T10:31:00.000000 | 2019-05-08T10:31:00.000000 |
+------+-----------+----------------------------+----------------------------+
$ openstack flavor list --private
+-----+-----------------+------+------+-----------+------+-----------+
| ID  | Name            |  RAM | Disk | Ephemeral | VCPUs | Is Public |
+-----+-----------------+------+------+-----------+------+-----------+
| yyy | reservation:zzz  | 1024 |   20 |         0 |     1 | False     |
+-----+-----------------+------+------+-----------+------+-----------+
$ openstack server create […] --flavor reservation:zzz test_instance
```

How does Blazar work?

Host reservation

- Reserve a number of hosts, optionally with specific attributes:

```
$ blazar lease-create --physical-reservation min=2,max=4,
resource_properties='["=", "$node_type", "compute_skylake"]' test_lease
```

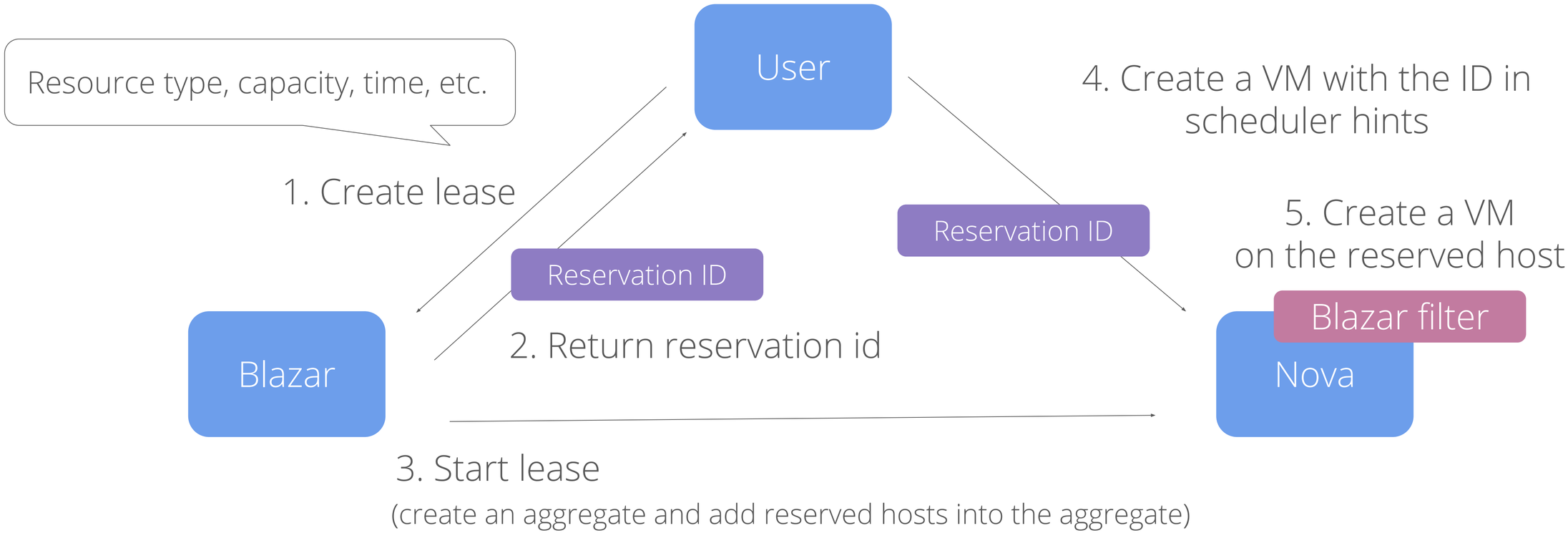
-> Blazar creates a host aggregate for the reserved hosts

```
$ blazar lease-list
+------+------------+----------------------------+----------------------------+
| id   | name       | start_date                 | end_date                   |
+------+------------+----------------------------+----------------------------+
| xxxx | test_lease | 2019-05-07T10:31:00.000000 | 2019-05-08T10:31:00.000000 |
+------+------------+----------------------------+----------------------------+
$ blazar lease-show test_lease
```

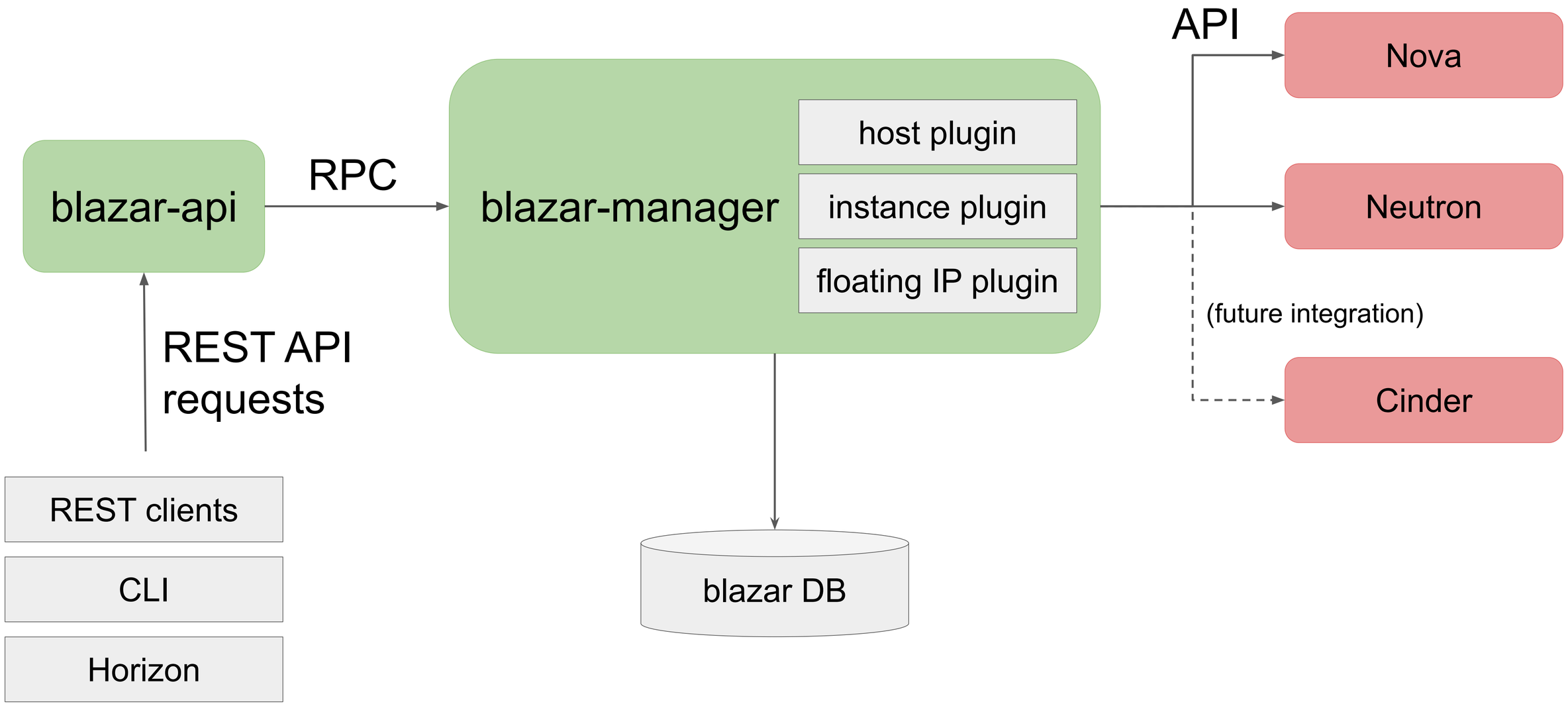
-> Blazar provides reservation IDs

```
$ openstack server create […] --hint reservation=zzz test_instance
```

Flow of reservation and lease (host reservation)



Architecture



Limitations

- No integration with quotas (yet)
  - Users can reserve more compute resources than allowed by their quota
- Incompatible with bare-metal compute nodes (yet)
  - Bare-metal nodes from Ironic cannot be managed by Blazar without patches
  - Chameleon uses a patched Nova which changes the behaviour of host aggregates
  - Work in progress for integration with standalone Ironic (no Nova)
- Reservable resource pool separate from other resources
  - Can lead to under-utilisation of the reservable resource pool
  - Planning to integrate with preemptible instances to increase utilisation
- Calendar not yet upstream
  - API implemented in Stein release cycle
  - Goal is to push calendar upstream in the current release cycle (Train)
- Centralised service
  - No high-availability

Demo

Create Lease

Please be courteous to other users of the testbed and make sure your lease represents a responsible use of Chameleon resources and complies with our best practices. Chameleon operators reserve the right to terminate leases judged to be abusive.

Lease Name *

For leases shorter than 24 hours, use a lease length of zero days.

Start Date Today

Lease Length (days) 1

Ends Tomorrow

Start Time Now

End Time Same time as now

Physical Hosts

☑ **Reserve Physical Host**

For specific node reservations, you can find the node UUID using Resource Discovery on the user portal.

Minimum Number of Hosts 1

Maximum Number of Hosts 1

Resource Properties

- node_type = compute_haswell X

Add Filter

Network

☐ **Reserve Network**

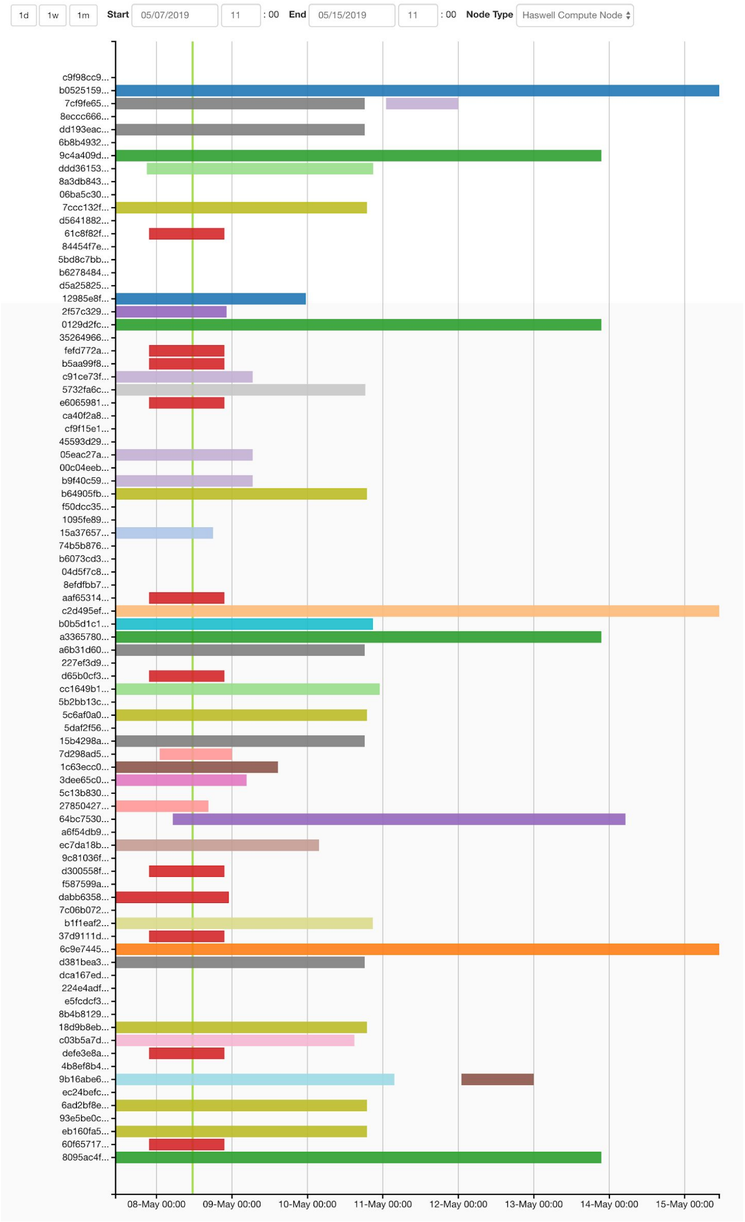
Network Name

Network Description

Number of Floating IP Addresses Needed 0

Cancel Create

Host Calendar



Thanks!

Q&A