Valet
Holistic Data Center Optimization for OpenStack

OpenStack Austin Summit, 27 April 2016

Bharath Balasubramanian, Joe D’Andrea, Matti Hiltunen, Kaustubh Joshi, and Gueyoung Jung
Cloud Services and Technology Research Lab
AT&T Advanced Technology and Architecture
Collaborators

AT&T Advanced Technology and Architecture
AT&T Technology Development - AIC
Additional partners in AT&T Domain 2.0
But first …

Let’s take a trip down memory lane and look at resource placement in OpenStack.
In the beginning …

There was **Nova**.

$ nova boot --image imageID --flavor flavorID --nic net-id=nicID
Constraints Were Simple

Amount of memory
Number of vCPUs
VM affinity and anti-affinity
... and then ...

There was Cinder.

$ cinder create --name volName --volume-type volType size
Constraints Were Still Simple

Size of volume
Backend selection
Backend affinity and anti-affinity
Scheduler Filters

One constraint check per filter. Hosts must pass all filters to be considered. All candidate hosts are weighted and sorted.
Nova Scheduler Filters

AggregateCoreFilter
AggregateDiskFilter
AggregateImagePropertiesIsolation
AggregateInstanceExtraSpecsFilter
AggregateIoOpsFilter
AggregateMultiTenancyIsolation
AggregateNumInstancesFilter
AggregateRamFilter
AggregateTypeAffinityFilter
AllHostsFilter
AvailabilityZoneFilter
ComputeCapabilitiesFilter
ComputeFilter
CoreFilter
NUMATopologyFilter
DifferentHostFilter
DiskFilter

GroupAffinityFilter
GroupAntiAffinityFilter
ImagePropertiesFilter
IsolatedHostsFilter
IoOpsFilter
JsonFilter
MetricsFilter
NumInstancesFilter
PciPassthroughFilter
RamFilter
RetryFilter
SameHostFilter
ServerGroupAffinityFilter
ServerGroupAntiAffinityFilter
SimpleCIDRAffinityFilter
TrustedFilter
TypeAffinityFilter

© 2016 AT&T Intellectual Property. All rights reserved. AT&T and the AT&T logo are trademarks of AT&T Intellectual Property.
What If …

A VM and volume need to be placed together? Cinder InstanceLocalityFilter … with caveats.

What if the VM hasn’t been created yet? What if it is and there isn’t enough disk space?
How About …

“Find a host with enough capacity to deploy a group of VMs that need to be together.”
How About …

“Place two or more VMs such that there’s 1 Gbps bandwidth available between each.”
How About …

“Deploy a bandwidth-intensive VNF such that it minimizes the use of oversubscribed spine switches.”
How About …

“Replicate a service chain of related VMs on different racks for fault tolerance.”
Example: Ceph distributed storage system

Bandwidth requirements

Affinity and diversity groups
“Austin, we have a problem.”

VMs and volumes are ultimately scheduled by Nova and Cinder one at a time.

What does this mean for cloud applications?
“We’re gonna need a bigger boat.”

Cloud apps can have lots of VMs and volumes. Often need complex combinations of filters.

Managing dependencies one resource at a time is tedious and error-prone.
Doesn’t Heat Solve This?

Deploy and update cloud applications holistically, using *declarative templates*.

```
$ heat stack-create my_app --template-file my_app.yaml
```
resources:

my_server:
  type: OS::Nova::Server
  properties:
    image: ubuntu-trusty-x86_64
    flavor: m1.small

my_volume:
  type: OS::Cinder::Volume
  properties:
    size: 5

my_volume_attachment:
  type: OS::Cinder::VolumeAttachment
  properties:
    instance_uuid: {get_resource: my_server}
    volume_id: {get_resource: my_volume}
Heat Is Constrained

Relies on Nova and Cinder schedulers.
Resources are still scheduled one at a time.
Introducing Valet
A Heat-level scheduler.

Provides a holistic placement service for cloud resources within a data center.
Valet’s Mission

Help meet cloud application requirements while optimizing the resource usage of a cloud’s infrastructure.
Without Holistic Application Placement
With Holistic Application Placement
How To Use Valet

Valet extends Heat with new resource types. Use these types to express app constraints.
ATT::Valet::Pipe

VM to VM bandwidth
VM to Volume IOPS
Bandwidth / IOPS Reservation

Valet picks hosts or backends with enough available bandwidth or IOPS between them.

Runtime enforcement is delegated to QoS.
Use Valet Piple to declare VM-to-VM (or VM-to-volume) bandwidth reservations for performance.

```yaml
mon1-client-bandwidth:
  type: ATT::Valet::Pipe
  properties:
    bandwidth: 5.0
    resources:
    - {get_resource: mon1-ceph}
    - {get_resource: client-ceph}
```

“Reserve 5 Mbps bandwidth between this Ceph monitor and client VM.”
ATT::Valet::ResourceGroup

Affinity groups
Diversity groups
Exclusivity groups
Affinity Group

Sometimes an application wants VMs and volumes to be close to each other.

Colocate resources on the same host, rack, or cluster.
Use Valet Resource Groups to declare placement affinity between related resources for **performance**.

```yaml
mon1-affinity:
  type: ATT::Valet::ResourceGroup
  properties:
    relationship: affinity
    level: host
    resources:
    - {get_resource: mon1-ceph}
    - {get_resource: vol1-mon1-ceph}
```

“Place this Ceph monitor VM and its volume on the **same host**.”
Diversity Group

Sometimes an app wants VMs and volumes to be distributed across multiple domains.

Can place resources on different hosts, racks, or clusters. Groups can also be nested.
Use Valet Resource Groups to declare placement diversity between resources for fault tolerance.

```
monitor-diversity:
  type: ATT::Valet::ResourceGroup
  properties:
    relationship: diversity
    level: rack
    resources:
    - {get_resource: mon1-affinity}
    - {get_resource: mon2-affinity}
    - {get_resource: mon3-affinity}
```

“Place these three Ceph monitor affinity groups on different racks.”
Exclusivity Group

Sometimes an application requires exclusive placement (e.g., core infrastructure VNF).
Exclusivity Group

Only VMs from tenants that are members of a group can be placed on a given host.

Hosts are marked for use by group members in a just-in-time fashion.
Use Valet Resource Groups to declare placement exclusivity across resources for application isolation.

```
my_exclusive_group:
  type: ATT::Valet::ResourceGroup
  properties:
    name: Core_Infrastructure_VNFs
    relationship: exclusivity
    level: host
    resources:
      - {get_resource: first_osd_VM}
      - {get_resource: second_osd_VM}
```

“Place these OSD VMs on the same host. Only tenants that are members of this group can use it.”
How Valet Works

It takes **five simple steps**.
Well, one of them is pretty hard.

(As in actually NP-hard.)
Valet Components
Front-end CLI and REST API
Application Placement Optimizer (Ostro)
Heat Stack Lifecycle Plugin
Nova and Cinder Scheduler Filter Plugins
Step 1: Infrastructure Discovery

The optimizer (Ostro) collects baseline information including physical data center topology.
Step 2: Use Heat

Create an **app template** with Valet resources. Give it to Heat. “Go build this.”
Step 2: Use Heat

Extended Heat Template → Heat → OpenStack Resource Plugins

App Declaration → OpenStack → Valet Stack Lifecycle Plugin → Valet

Application deployment

Valet's Stack Lifecycle Plugin intercepts the request.
Step 2: Use Heat

The Valet API Service is asked to create an application plan.
Heat Stack Lifecycle

Create → Update → Resume → Suspend → Delete
Step 3: Holistic Placement

The request is sanity **checked**. Then we call in the **optimizer**.
Optimization: Ostro

Hybrid of an A* “best first” graph search and a greedy algorithm to find a placement that maximizes host and network utilization.

Comprehensive evaluation in ICDCS ’15 paper:
Step 3: Holistic Placement

Application placement is held in escrow. Each decision is tied to an Orchestration ID.
Heat Orchestration ID

A unique ID assigned by Heat to each VM and volume before resource creation.

Heat can send these IDs to Nova / Cinder using **Scheduler Hints** behind-the-scenes.
Step 4: Heat Builds the Stack

Valet Scheduler Filters ask API service for placement. No Orchestration ID: Placed on-the-fly.
Step 5: Closing the Loop

Optimizer monitors the **Oslo message bus**. Cloud infrastructure view is kept **up-to-date**.
High Availability in AIC: Music

Data replication / coordination service. Used for data sharing and persistence.
High Availability in AIC: Music

HAProxy Service

Active

Valet

Persistence Service

HAValet

OpenStack

Valet / HAValet

Persistence Service

Passive

Valet

Persistence Service

HAValet

Persistence Service
What's Next

Initial deployment to AIC is in progress. Will release as Open Source once deployed.

Aspiring to join the OpenStack big tent as a new project, working with the community.
Contact Us

Cloud Services and Technology Research Lab
AT&T Advanced Technology and Architecture
cloud-lab@research.att.com