CHINA OpenInfra Days

CHINA OpenInfra Days | IT大咖说 知识共享平台

移动大云OpenStack针对Nova-Placement服务的研究与优化

演讲者：徐磊
中国移动苏州研发中心

内容：

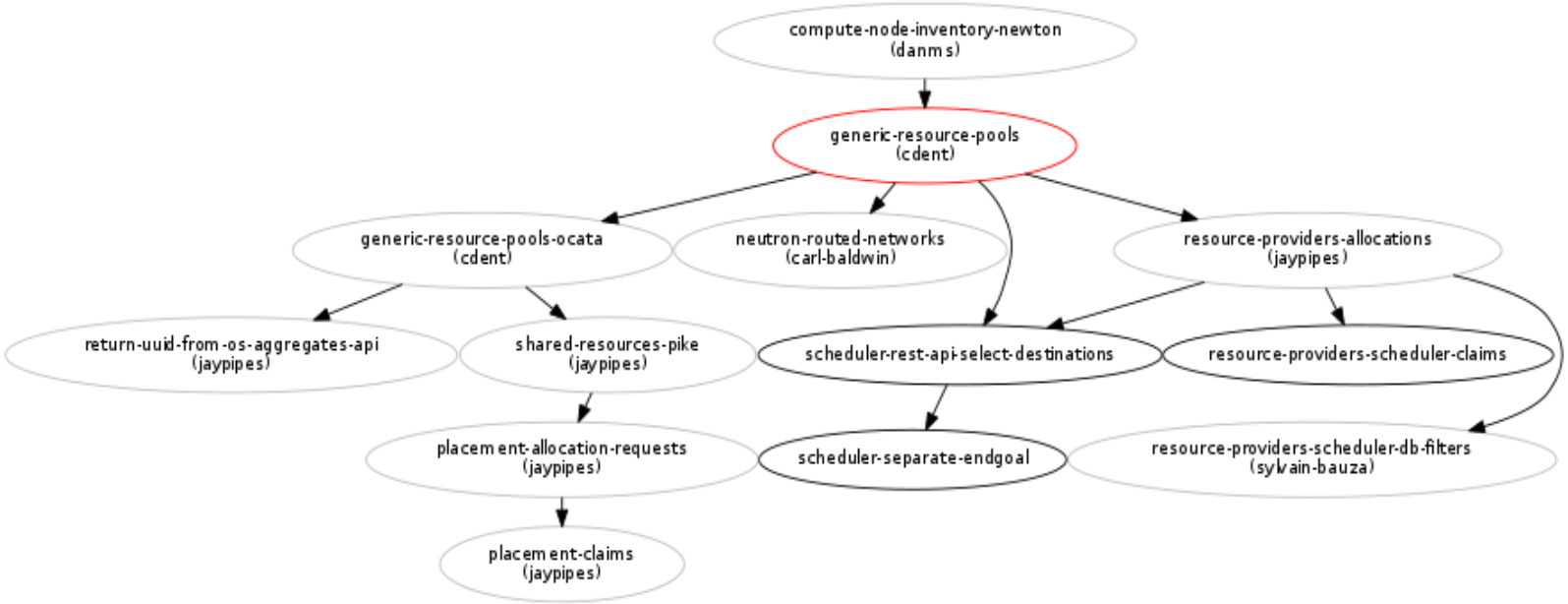
- 1. 背景
- 2. 针对Nova-Placement功能与使用研究
  - 2.1 概念说明
  - 2.2 资源池使用
  - 2.3 placement api服务
  - 2.4 安装调试流程
- 3. 针对Nova-Placement在实践中的优化
  - 3.1 资源上报、调度选择对比，虚机调度压力分析
  - 3.2 大规模节点场景下的调度优化

背景：

- OpenStack转向高版本开发
- NFV场景、裸金属场景的需要

- 功能单一，配置不灵活
- 共享资源无法统计
- 集群规模大，scheduler成瓶颈

社区目标：


compute-node-inventory-newton
(danms)
generic-resource-pools
(cdent)
generic-resource-pools-ocata
(cdent)
neutron-routed-networks
(carl-baldwin)
resource-providers-allocations
(jaypipes)
return-uuid-from-os-aggregates-api
(jaypipes)
shared-resources-pike
(jaypipes)
scheduler-rest-api-select-destinations
resource-providers-scheduler-claims
placement-allocation-requests
(jaypipes)
scheduler-separate-endgoal
resource-providers-scheduler-db-filters
(sylvain-bauza)
placement-claims
(jaypipes)

针对NOVA-PLACEMENT功能与使用研究：

概念：

- Resource provider --- 资源提供者
- Inventory --- 资源清单
- Resource class --- 资源种类
- Trait --- 资源特征
- Provider aggregate --- 资源聚合

Inventory

MEMORY_MB：　　# class
total=131072　　# 总共大小
reserved=512　　# 保留大小
min-unit=1　　# 最小分配单位
max-unit=131072　　# 最大分配单位
step-size=1　　# 步长
allocation-ratio=1.0　　# 超分比

以MEMORY_MB为例：
allocation-ratio对应：ram_allocation_ratio
reserved对应：reserved_host_memory_mb
min-unit、max-unit、step-size实现精细控制

Resource class

计算节点：VCPU、MEMORY_MB、DISK_GB、VGPU

总共：

name
VCPU
MEMORY_MB
DISK_GB
PCI_DEVICE
SRIOV_NET_VF
NUMA_SOCKET
NUMA_CORE
NUMA_THREAD
NUMA_MEMORY_MB
IPV4_ADDRESS
VGPU
VGPU_DISPLAY_HEAD

还有CUSTOM_的自定义resource class

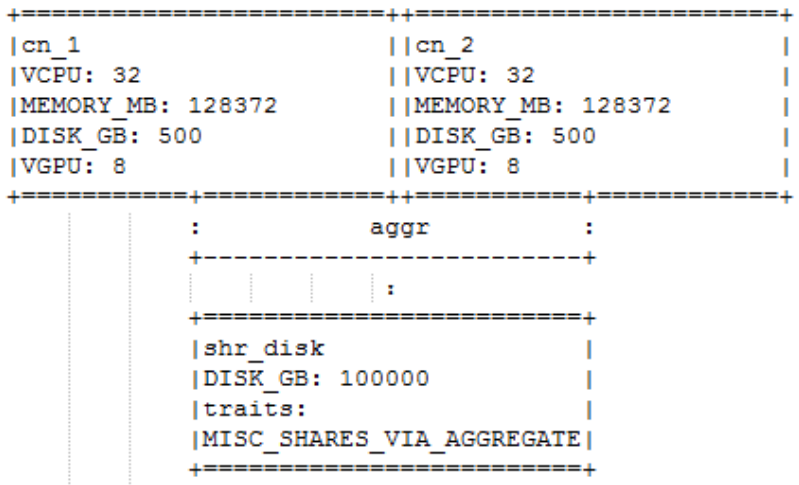
Trait

对应于metadata，提供调度所使用的关键字

os_traits:
hw -- cpu\gpu
storage -- ssd\hhd
misc -- MISC_SHARES_VIA_AGGREGATE

资源池：

shared storage ?



资源池：

创建测试

CN1_UUID=5648264b-c3a6-474f-bc01-eb21d1810339
CN2_UUID=c6bcdba0-a2aa-409e-9368-a15f108d6b1e
openstack aggregate create aggr
AGG_UUID=9bdfe3e3-02a0-4397-9a11-37cfb7838f87
openstack aggregate add host aggr cn_1
openstack aggregate add host aggr cn_2
openstack resource provider create shr_disk
RP_UUID=48b2e7a4-ca04-480b-bd32-a617e5de0adb
openstack resource provider aggregate set --aggregate $AGG_UUID $RP_UUID
openstack resource provider inventory class set $RP_UUID DISK_GB --total=100000 --reserved=1000 --min-unit=1 --max-unit=5000 --step-size=1 --allocation-ratio=1.0
openstack resource provider trait set $RP_UUID --trait MISC_SHARES_VIA_AGGREGATE
openstack resource provider aggregate set --aggregate $AGG_UUID $CN1_UUID
openstack resource provider aggregate set --aggregate $AGG_UUID $CN2_UUID

Flavor信息，创建instance

ID	Name	Memory_MB	Disk	Ephemeral	Swap	VCPUs
c6a07ef8-1925-49e0-b5aa-065d398138f9	test	4096	800	0		4

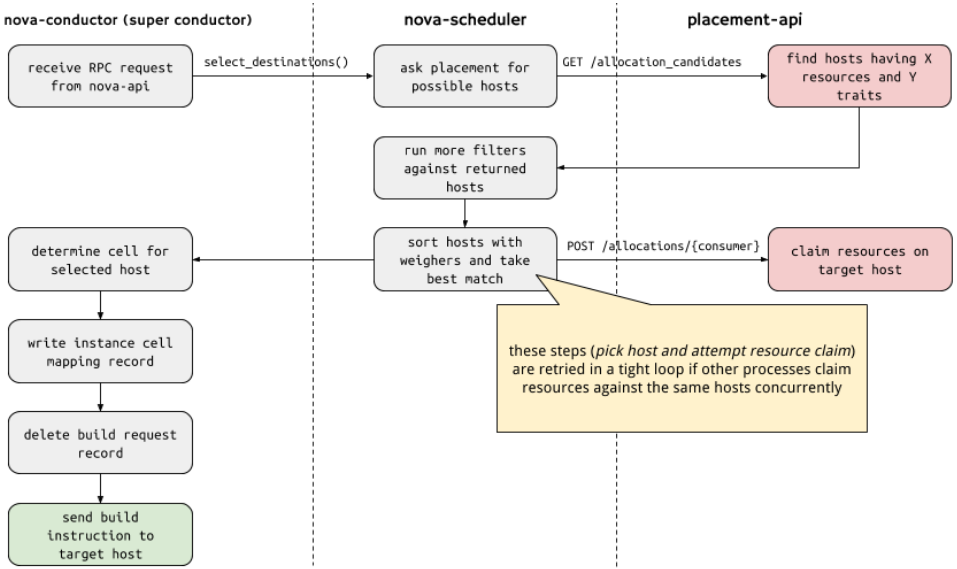
shr_disk provider 资源占用情况

```
openstack resource provider usage show 48b2e7a4-ca04-480b-bd32-a617e5de0adb
+----------------+-------+
| resource_class | usage |
+----------------+-------+
| DISK_GB        |   800 |
+----------------+-------+
```

Computenode provider 资源占用情况

```
openstack resource provider usage show 5648264b-c3a6-474f-bc01-eb21d1810339
+----------------+-------+
| resource_class | usage |
+----------------+-------+
| VCPU           |     4 |
| MEMORY_MB      |  4609 |
| DISK_GB        |     0 |
+----------------+-------+
```

PLACEMENT API：



GET /placement/allocation_candidates?limit=1000&resources=DISK_GB:800,MEMORY_MB:1024,VCPU:2

？limit含义

Flavor -- Disk：800，Memory_MB：1024，VCPUs：2

https://github.com/jaypipes/articles/blob/master/openstack/walkthrough-launch-instance-request.md

https://docs.openstack.org/nova/latest/user/placement.html

安装调试流程：

1. 同步数据库

2. 创建账户及catalog信息

3. 配置nova-compute配置文件

Final resource view: name=ec4-com phys_ram=15708MB used_ram=512MB phys_disk=98GB used_disk=0GB total_vcpus=8 used_vcpus=0 pci_stats=[]

Placement API service is not responding.: ConnectFailure: Unable to establish connection to http://xxxx:8778/resource_providers/e3a717fd-e906-40dd-a3e5-e7bdf96ea8c0/aggregates

安装Client：https://github.com/openstack/osc-placement/
https://github.com/Remy-xl/placement-rpm/

调试：建议使用remote_pdb
https://docs.openstack.org/devstack/latest/systemd.html#using-remote-pdb

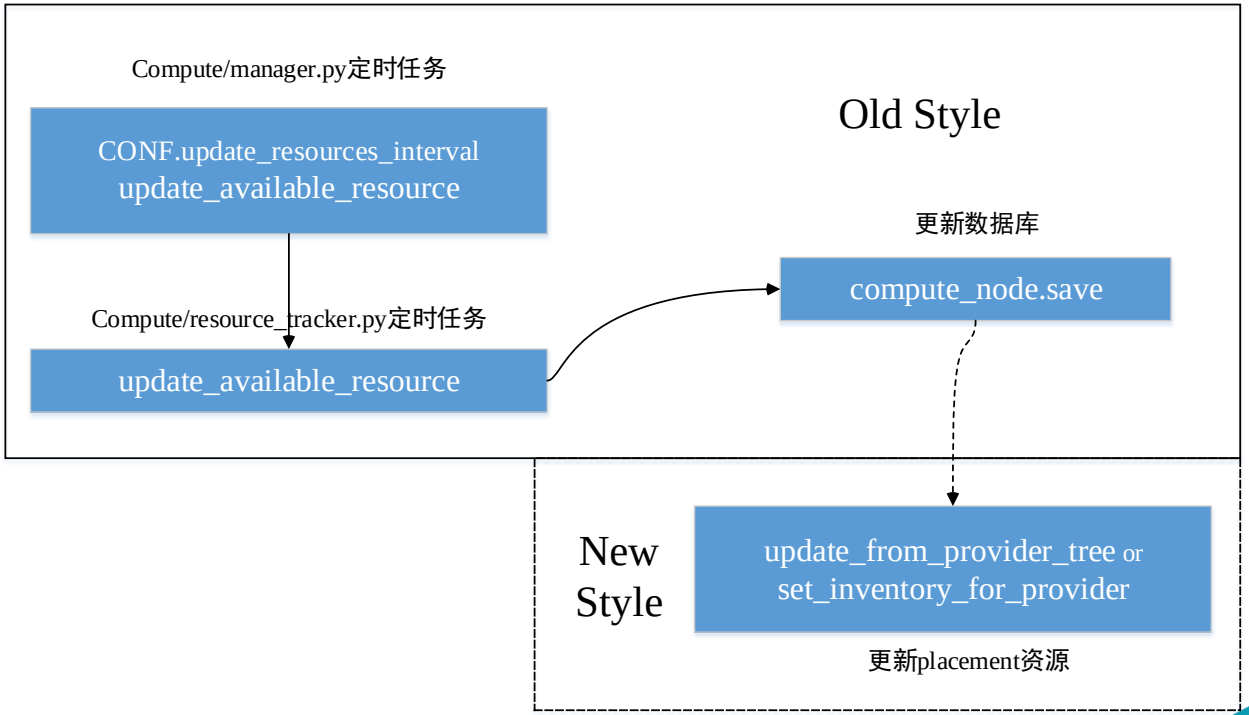
针对NOVA-PLACEMENT在实践中的优化

资源上报对比：

resource_tracker.py

↓

压力？



调度选择对比：

filter_scheduler

-> select_destinations　# manager.py　响应rpc请求

-> get_allocation_candidates 发送rest给placement，获取候选者

-> select_destinations　# filter_scheduler .py 调用driver方法

-> _schedule　　从候选者中调度

-> _get_all_host_states　从数据库中读取信息

-> _get_sorted_hosts　开始排序

-> get_filtered_hosts　过滤

-> get_weighed_hosts　权重

-> claim_resources　发送rest给placement，占用资源

caching_scheduler

-> select_destinations　# manager.py　响应rpc请求

-> select_destinations　# filter_scheduler .py 调用driver方法

-> _schedule　　从候选者中调度

-> _get_all_host_states　# caching_scheduler.py 从内存中读取信息

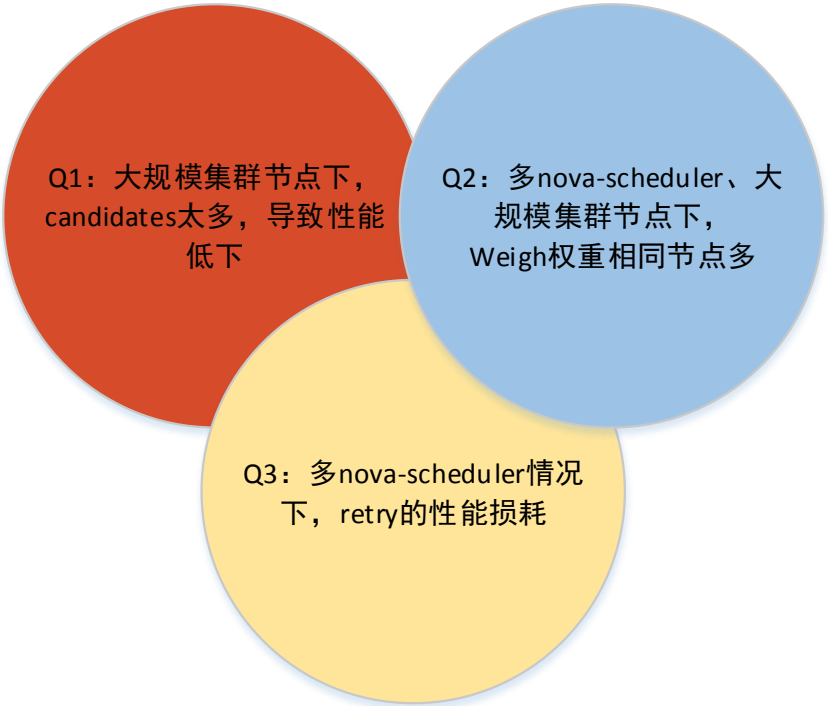
-> _ legacy_find_hosts　进入常规排序流程

-> _get_sorted_hosts　开始排序

-> get_filtered_hosts　过滤

-> get_weighed_hosts　权重

虚机调度压力分析：


Q1：大规模集群节点下，candidates太多，导致性能低下
Q2：多nova-scheduler、大规模集群节点下，Weigh权重相同节点多
Q3：多nova-scheduler情况下，retry的性能损耗

大规模节点场景下的调度优化 1：

max_placement_results： 默认值为1000，
建议根据环境规模修改

配合使用

```
if limit and limit <= len(alloc_request_objs):
    if CONF.placement.randomize_allocation_candidates:
        alloc_request_objs = random.sample(alloc_request_objs, limit)
    else:
        alloc_request_objs = alloc_request_objs[:limit]
```

列表随机截取

列表切片

randomize_allocation_candidates： 默认为false
如果max_placement_results设置小于节点数，
建议设为true

性能与可靠性 tradeoff？ problem?

https://review.openstack.org/#/c/513526/

大规模节点场景下的调度优化 2：

-> select_destinations

-> get_allocation_candidates

-> select_destinations

-> _schedule

-> _get_all_host_states

-> _get_sorted_hosts

-> get_filtered_hosts

-> get_weighed_hosts

-> claim_resources

```
if CONF.filter_scheduler.shuffle_best_same_weighed_hosts:
   best_hosts = [w for w in weighed_hosts
                 if w.weight == weighed_hosts[0].weight]
   random.shuffle(best_hosts)
   weighed_hosts = best_hosts + weighed_hosts[len(best_hosts):]
```

shuffle_best_same_weighed_hosts：默认false，建议开启

```
host_subset_size = CONF.filter_scheduler.host_subset_size
if host_subset_size < len(weighed_hosts):
   weighed_subset = weighed_hosts[0:host_subset_size]
else:
   weighed_subset = weighed_hosts
```

host_subset_size： 默认值为1，建议根据环境规模修改

```
chosen_host = random.choice(weighed_subset)
weighed_hosts.remove(chosen_host)
return [chosen_host] + weighed_hosts
```

大规模节点场景下的调度优化 3：

Claim机制主要解决的问题是：当一台主机被多个nova-scheduler同时选中并发送创建VM的请求时，对主机的资源进行验证。

```
# conductor
-> schedule_and_build_instances

   -> _schedule_instances

      #  scheduler
       ……调度流程……..

   -> build_and_run_instance

      #  compute
       -> rt.instance_claim
```

繁琐！

改进 claiming to placement

PUT /placement/allocations/1e3f34c8-a3e2-4a79-8b75-b86cc63140d1

1e3f34c8-a3e2-4a79-8b75-b86cc63140d1 为虚机id

```
+--------------------------------------+------+--------+------------+-------------+----------+
| ID                                   | Name | Status | Task State | Power State | Networks |
+--------------------------------------+------+--------+------------+-------------+----------+
| 1e3f34c8-a3e2-4a79-8b75-b86cc63140d1 | test | ACTIVE | -          | Running     |          |
+--------------------------------------+------+--------+------------+-------------+----------+
```

占用资源： openstack resource provider allocation show 1e3f34c8-a3e2-4a79-8b75-b86cc63140d1

```
+--------------------------------------+------------+------------------------------------------------+
| resource_provider                    | generation | resources                                      |
+--------------------------------------+------------+------------------------------------------------+
| 5648264b-c3a6-474f-bc01-eb21d1810339 |         28 | {u'VCPU': 1, u'MEMORY_MB': 512, u'DISK_GB': 1} |
+--------------------------------------+------------+------------------------------------------------+
```

CHINA
OpenInfra Days

Thank You