

一、关于从Volume和Snapshot创建卷时不经过c-sch原因分析

从目前Ocata版本的情形来看，默认情形下Cinder从Volume和Snapshot创建卷时跳过了c-sch，c-api根据源的host信息直接指定了生成target的host信息，并直接调用vol的API执行创建！

该段逻辑在名为create_volume_api的TaskFlow中，具体实现为VolumeCastTask的 cast_create_volume方法

```
cinder.volume.flows.api.create_volume._cast_create_volume
```

该方法中只要request_spec['source_volid']，或者 request_spec['source_replicaid']非空，该方法会直接跳过scheduler执行volume的create_volume_anager。因此对于从volume创建的卷，必然跳过scheduler。而对于从scheduler创建的volume，提供了一个配置项（CONF.snapshot_same_host），改配置项默认为True，此时从快照创建的volume也必须跳过scheduler。

```
def _cast_create_volume(self, context, request_spec, filter_properties):
    source_volume_ref = None
    source_volid = (request_spec['source_volid'] or
                    request_spec['source_replicaid'])
    volume = request_spec['volume']
    snapshot_id = request_spec['snapshot_id']
    image_id = request_spec['image_id']
    cgroup_id = request_spec['consistencygroup_id']
    cgsnapshot_id = request_spec['cgsnapshot_id']
    group_id = request_spec['group_id']
    if cgroup_id:
        cgroup = objects.ConsistencyGroup.get_by_id(context, cgroup_id)
        request_spec['CG_backend'] = vol_utils.extract_host(cgroup.host)
    elif group_id:
        group = objects.Group.get_by_id(context, group_id)
        request_spec['group_backend'] = vol_utils.extract_host(group.host)
    """
    根据配置项控制snapshot方式创建的逻辑
    """
    elif snapshot_id and CONF.snapshot_same_host:
        snapshot = objects.Snapshot.get_by_id(context, snapshot_id)
        source_volume_ref = snapshot.volume
    elif source_volid:
        source_volume_ref = objects.Volume.get_by_id(context, source_volid)
    if not source_volume_ref:
        self.scheduler_rpcapi.create_volume(
            context,
            volume,
            snapshot_id=snapshot_id,
            image_id=image_id,
            request_spec=request_spec,
            filter_properties=filter_properties)
    else:
        """
        执行create_volume_manager，直接指定新volume的HOST和源相当。
        """
        volume.host = source_volume_ref.host
        volume.cluster_name = source_volume_ref.cluster_name
        volume.scheduled_at = timeutils.utcnow()
        volume.save()
        if not cgsnapshot_id:
            self.volume_rpcapi.create_volume(
                context,
```

```
        volume,
        request_spec,
        filter_properties,
        allow_reschedule=False)
```

二、关于create_volume_manager的调用链

方式	VolumeManager.driver接口	NetApp的driver相应实现	NetAPP Client的接口
raw	create_volume	create_volume	self.zapi_client.create_lun(volu
snap	create_volume_from_snapshot	create_volume_from_snapshot -->_clone_source_to_destination -->_clone_lun	self.zapi_client.clone_lun(volu qos_ src_ bloc sour is_s
volume	create_cloned_volume	create_cloned_volume -->_clone_source_to_destination -->_clone_lun -->_copy_lun	
source_replica	create_replica_test_volume	没有实现	
image	image加密时: clone_image Raw格式, 且cinder对接glance: create_cloned_volume 下载, 或者缓存: create_volume	没有实现 create_volume_from_snapshot create_cloned_volume	