

## 目 录

<b>1</b>	<b>分析 resize 流程前的必要知识</b>	<b>2</b>
1.1	nova 中的 RPC 机制 . . . . .	2
1.2	重要的数据类型 . . . . .	2
1.2.1	req . . . . .	2
1.2.2	context . . . . .	2
1.2.3	instance . . . . .	2
<b>2</b>	<b>nova-api 阶段</b>	<b>2</b>
<b>3</b>	<b>nova-conductor 部分</b>	<b>3</b>
<b>4</b>	<b>冷迁移</b>	<b>4</b>
4.1	冷迁移中的 nova-conductor 部分 . . . . .	4
4.2	冷迁移中的 nova-compute 部分 . . . . .	5
4.2.1	目的主机上的操作: prep_resize . . . . .	5
4.2.2	源主机的操作: resize_instance . . . . .	5
4.2.3	目的主机上的操作: finish_resize . . . . .	6

## 1 分析 resize 流程前的必要知识

### 1.1 nova 中的 RPC 机制

### 1.2 重要的数据类型

#### 1.2.1 req

#### 1.2.2 context

```
1 # 根据req创建环境上下文context
2 # context是nova/context.py中的RequestContext类
3 context = req.environ["nova.context"]
```

#### 1.2.3 instance

```
1 # 根据req和instance_id创建instance
2 # instance是nova/context/instance.py中的Instance类
3 instance = self._get_server(context, req, instance_id)
4
5
6 instance_type
7 flavor_id
8 deltas
9 quotas
10 vm_state
```

## 2 nova-api 阶段

入口函数为：

```
1 # 这个函数在nova/api/openstack/compute/servers.py
2 def _resize(self, req, instance_id, flavor_id, **kwargs):
3     ...
4     try:
5         # compute_api是nova/compute/api.py中的API类
6         self.compute_api.resize(context, instance, flavor_id, **kwargs)
7     ...
```

进一步看 API.resize() 函数：

```
1 # nova/compute/api.py API.resize()
2 def resize(self, context, instance, flavor_id=None,
3            **extra_instance_updates):
```

```

4         ...
5
6         # filter_properties与选择本地扩容或选择异地扩容有关
7         filter_properties = {'ignore_hosts': []}
8
9         if not CONF.allow_resize_to_same_host:
10             filter_properties['ignore_hosts'].append(instance['host'])
11
12         if (not flavor_id and not CONF.allow_migrate_to_same_host):
13             filter_properties['ignore_hosts'].append(instance['host'])
14
15         ...
16
17         # scheduler_hint挺重要的, 是nova-scheduler的参数
18         scheduler_hint = {'filter_properties': filter_properties}
19         self.compute_task_api.resize_instance(context, instance,
20                                             extra_instance_updates, scheduler_hint=scheduler_hint,
21                                             flavor=new_instance_type,
22                                             reservations=quotas.reservations or [])

```

```

1     # nova/conductor/api.py ComputeTaskAPI.resize_instance()
2     def resize_instance(self, context, instance, extra_instance_updates,
3                       scheduler_hint, flavor, reservations):
4         self.conductor_compute_rpcapi.migrate_server(
5             context, instance, scheduler_hint, False, False, flavor,
6             None, None, reservations)

```

```

1     # nova/conductor/rpcapi.py ComputeTaskAPI.migrate_server()
2     def migrate_server(self, context, instance, scheduler_hint, live, rebuild,
3                      flavor, block_migration, disk_over_commit,
4                      reservations=None):
5         ...
6         cctxt = self.client.prepare(version=version)
7         return cctxt.call(context, 'migrate_server',
8                          instance=instance, scheduler_hint=scheduler_hint,
9                          live=live, rebuild=rebuild, flavor=flavor_p,
10                         block_migration=block_migration,
11                         disk_over_commit=disk_over_commit,
12                         reservations=reservations)

```

### 3 nova-conductor 部分

```

1     # nova/conductor/manager.py ComputeTaskManager.migrate_server()
2     def migrate_server(self, context, instance, scheduler_hint, live, rebuild,
3                      flavor, block_migration, disk_over_commit, reservations=None):
4         ...
5         if live and not rebuild and not flavor:
6             self._live_migrate(context, instance, scheduler_hint,
7                               block_migration, disk_over_commit)
8         elif not live and not rebuild and flavor:

```

```

9         ...
10         with compute_utils.EventReporter(context, 'cold_migrate',
11                                           instance_uuid):
12             self._cold_migrate(context, instance, flavor,
13                               scheduler_hint['filter_properties'],
14                               reservations)
15         ...

```

## 4 冷迁移

### 4.1 冷迁移中的 nova-conductor 部分

```

1  # nova/conductor/manager.py ComputeTaskManager._cold_migrate()
2  def _cold_migrate(self, context, instance, flavor, filter_properties,
3                    reservations):
4      ...
5      try:
6          ...
7          # 选择目的主机
8          hosts = self.scheduler_client.select_destinations(
9                  context, request_spec, filter_properties)
10         host_state = hosts[0]
11         ...
12
13     try:
14         ...
15         (host, node) = (host_state['host'], host_state['nodename'])
16         self.compute_rpcapi.prep_resize(
17             context, image, instance,
18             flavor, host,
19             reservations, request_spec=request_spec,
20             filter_properties=filter_properties, node=node)
21     ...

```

```

1  # nova/compute/rpcapi.py ComputeAPI.prep_resize()
2  def prep_resize(self, ctxt, image, instance, instance_type, host,
3                  reservations=None, request_spec=None,
4                  filter_properties=None, node=None):
5      ...
6      cctxt = self.client.prepare(server=host, version=version)
7      cctxt.cast(ctxt, 'prep_resize',
8                 instance=instance,
9                 instance_type=instance_type_p,
10                 image=image_p, reservations=reservations,
11                 request_spec=request_spec,
12                 filter_properties=filter_properties,
13                 node=node)

```

## 4.2 冷迁移中的 nova-compute 部分

### 4.2.1 目的主机上的操作: `prep_resize`

```

1  # nova/compute/manager.py ComputeManager.prep_resize()
2  def prep_resize(self, context, image, instance, instance_type,
3                  reservations, request_spec, filter_properties, node):
4      ...
5      with self._error_out_instance_on_exception(context, instance,
6                                                  quotas=quotas):
7          ...
8          try:
9              self._prep_resize(context, image, instance,
10                              instance_type, quotas,
11                              request_spec, filter_properties,
12                              node)
13      ...

```

```

1  # nova/compute/manager.py ComputeManager._prep_resize()
2  def _prep_resize(self, context, image, instance, instance_type,
3                  quotas, request_spec, filter_properties, node):
4      ...
5      with rt.resize_claim(context, instance, instance_type,
6                          image_meta=image, limits=limits) as claim:
7          ...
8          self.compute_rpcapi.resize_instance(
9              context, instance, claim.migration, image,
10              instance_type, quotas.reservations)
11

```

```

1  # nova/compute/rpcapi.py ComputeAPI.resize_instance()
2  def resize_instance(self, ctxt, instance, migration, image, instance_type,
3                     reservations=None):
4      ...
5      cctxt = self.client.prepare(server=_compute_host(None, instance),
6                                  version=version)
7      cctxt.cast(ctxt, 'resize_instance',
8                 instance=instance, migration=migration,
9                 image=image, reservations=reservations,
10                 instance_type=instance_type_p)

```

### 4.2.2 源主机的操作: `resize_instance`

```

1  # nova/compute/manager.py ComputeManager.resize_instance()
2  def resize_instance(self, context, instance, image,
3                    reservations, migration, instance_type,
4                    clean_shutdown=True):
5      ...
6      with self._error_out_instance_on_exception(context, instance,
7                                                  quotas=quotas):

```

```

8      ...
9      # 获得虚拟机块设备的信息
10     block_device_info = self._get_instance_block_device_info(
11         context, instance, bdms=bdms)
12
13     # 关闭虚拟机并迁移虚拟机的增量文件
14     disk_info = self.driver.migrate_disk_and_power_off(
15         context, instance, migration.dest_host,
16         instance_type, network_info,
17         block_device_info,
18         timeout, retry_interval)
19     ...
20     self.compute_rpcapi.finish_resize(context, instance,
21         migration, image, disk_info,
22         migration.dest_compute, reservations=quotas.reservations)
23     ...

```

`migrate_disk_and_power_off()` 是源主机上将虚拟机迁移给目的主机的实现函数，主要利用了 `libvirt` API。这个函数的分析在《nova 调用 `libvirt`》中的“nova 扩容时对 `libvirt` 的调用”一节。

```

1  # nova/compute/rpcapi.py(690) ComputeAPI.finish_resize()
2  def finish_resize(self, ctxt, instance, migration, image, disk_info,
3      host, reservations=None):
4      ...
5      cctxt = self.client.prepare(server=host, version=version)
6      cctxt.cast(ctxt, 'finish_resize',
7          instance=instance, migration=migration,
8          image=image, disk_info=disk_info, reservations=reservations)

```

### 4.2.3 目的主机上的操作：finish\_resize

```

1  # nova/compute/manager.py ComputeManager.finish_resize()
2  def finish_resize(self, context, disk_info, image, instance,
3      reservations, migration):
4      quotas = quotas_obj.Quotas.from_reservations(context,
5          reservations,
6          instance=instance)
7      try:
8          self._finish_resize(context, instance, migration,
9              disk_info, image)
10     ...

```

```

1  # nova/compute/manager.py ComputeManager._finish_resize()
2  def _finish_resize(self, context, instance, migration, disk_info,
3      image):
4      ...
5      try:
6          self.driver.finish_migration(context, migration, instance,
7              disk_info,
8              network_info,

```

```
9         image, resize_instance ,  
10         block_device_info, power_on)  
11     ...
```