

本站文章大部分为作者原创, 非商业用途转载无需作者授权, 但务必在文章标题下面注明作者 刘世民 (Sammy Liu) 以及可点击的本博客地址超级链接 <http://www.cnblogs.com/sammyliu/>, 谢谢合作



www.cnblogs.com

世民谈云计算

(声明: 本站文章皆基于公开来源信息, 仅代表作者个人观点, 与作者所在公司无关)

昵称: SammyLiu
园龄: 2年6个月
荣誉: 推荐博客
粉丝: 470
关注: 30
[+加关注](#)

<	2017年5月						>
	日	一	二	三	四	五	六
30	1	2	3	4	5	6	
7	8	<u>9</u>	10	11	12	13	
14	15	16	17	18	19	20	
21	22	23	24	25	26	27	
28	29	30	31	1	2	3	
4	5	6	7	8	9	10	

常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)

我的标签

[GRE\(1\)](#)
[Neutron\(1\)](#)
[Open vSwitch\(1\)](#)
[OpenStack\(1\)](#)

随笔分类(254)

[Ceilometer\(3\)](#)
[Ceph\(13\)](#)
[Cinder\(6\)](#)
[Docker\(8\)](#)
[Glance](#)
[Heat\(2\)](#)
[K8S](#)
[Keystone\(1\)](#)
[KVM\(10\)](#)
[MessageQueue\(4\)](#)
[MySQL\(1\)](#)
[Neutron\(17\)](#)
[Nova\(10\)](#)
[OpenStack\(33\)](#)
[Sahara](#)
[Storage\(1\)](#)
[Swift\(3\)](#)
[Trove](#)
[Ubuntu\(3\)](#)
[VMware\(3\)](#)

[博客园](#) [首页](#) [新随笔](#) [订阅](#) [XML](#) [管理](#)

随笔-121 评论-504 文章-45

KVM 介绍 (7) : 使用 libvirt 做 QEMU/KVM 快照和 Nova 实例的快照 (Nova Instances Snapshot Libvirt)

学习 KVM 的系列文章:

- [\(1\) 介绍和安装](#)
- [\(2\) CPU 和 内存虚拟化](#)
- [\(3\) I/O QEMU 全虚拟化和准虚拟化 \(Para-virtualization\)](#)
- [\(4\) I/O PCI/PCIe设备直接分配和 SR-IOV](#)
- [\(5\) libvirt 介绍](#)
- [\(6\) Nova 通过 libvirt 管理 QEMU/KVM 虚拟机](#)
- [\(7\) 快照 \(snapshot\)](#)
- [\(8\) 迁移 \(migration\)](#)

本文将梳理 QEMU/KVM 快照相关的知识, 以及在 OpenStack Nova 中使用 libvirt 来对 QEMU/KVM 虚拟机做快照的过程。

1. QEMU/KVM 快照

1.1 概念

QEMU/KVM 快照的定义: 快照就是将虚拟机在某一个时间点上的磁盘、内存和设备状态保存一下, 以备将来之用。它包括以下几类:

- 磁盘快照: 磁盘的内容 (可能是虚拟机的全部磁盘或者部分磁盘) 在某个时间点上被保存, 然后可以被恢复。
 - 磁盘数据的保存状态:
 - 在一个运行着的系统上, 一个磁盘快照很可能只是崩溃一致的 (crash-consistent) 而不是完整一致 (clean) 的, 也是说它所保存的磁盘状态可能相当于机器突然掉电时硬盘数据的状态, 机器重启后需要通过 fsck 或者别的工具来恢复到完整一致的状态 (类似于 Windows 机器在断电后会执行文件检查)。(注: 命令 `qemu-img check -f qcow2 --output=qcow2 -r all filename=img.qcow2` 可以对 qcow2 和 vid 格式的镜像做一致性检查。)
 - 对一个非运行中的虚拟机来说, 如果上次虚拟机关闭的时候磁盘是完整一致的, 那么其被快照的磁盘快照也将是完整一致的。
 - 磁盘快照有两种:
 - 内部快照 - 使用单个的 qcow2 的文件来保存快照和快照之后的改动。这种快照是 libvirt 的默认行为, 现在的支持很完善 (创建、回滚和删除), 但是只能针对 qcow2 格式的磁盘镜像文件, 而且其过程较慢等。
 - 外部快照 - 快照是一个只读文件, 快照之后的修改是另一个 qcow2 文件中。外置快照可以针对各种格式的磁盘镜像文件。外置快照的结果是形成一个 qcow2 文件链: `original <- snap1 <- snap2 <- snap3`。 [这里有文章](#) 详细讨论外置快照。
- 内存状态 (或者虚拟机状态): 只是保持内存和虚拟机使用的其它资源的状态。如果虚拟机状态快照在做和恢复之间磁盘没有被修改, 那么虚拟机将保持一个持续的状态; 如果被修改了, 那么很可能导致数据 corruption。
- 系统还原点 (system checkpoint): 虚拟机的所有磁盘的快照和内存状态快照的集合, 可用于

安装和配置(1)
版本(4)
备份(1)
大数据(5)
翻译(4)
高可用 (HA) (6)
基础知识(19)
监控(1)
容器(4)
容器编排
使用案例(4)
网络(8)
问题定位(3)
行业(14)
性能(4)
虚拟化(7)
原理(22)
云Cloud(29)
随笔档案(121)
2017年5月 (1)
2017年3月 (1)
2017年1月 (1)
2016年10月 (7)
2016年9月 (5)
2016年8月 (4)
2016年7月 (1)
2016年6月 (5)
2016年5月 (1)
2016年4月 (1)
2016年3月 (9)
2016年2月 (4)
2016年1月 (2)
2015年12月 (7)
2015年11月 (7)
2015年10月 (4)
2015年9月 (4)
2015年8月 (5)
2015年7月 (9)
2015年6月 (10)
2015年5月 (3)
2015年4月 (11)
2015年3月 (2)
2015年2月 (6)
2015年1月 (5)
2014年12月 (6)
文章分类(21)
Ceph(1)
GlusterFS
Web 服务器(2)
操作系统(1)
存储

恢复完整的系统状态（类似于系统休眠）。

关于 崩溃一致（crash-consistent）的附加说明：

- 应该尽量避免在虚拟机I/O繁忙的时候做快照。这种时候做快照不是可取的办法。
- vmware 的做法是装一个 tools，它是个 PV driver，可以在做快照的时候挂起系统
- 似乎 KVM 也有类似的实现 QEMU Guest Agent，但是还不是很成熟，可参考 http://wiki.libvirt.org/page/Qemu_guest_agent

快照还可以分为 live snapshot（热快照）和 Clod snapshot：

- Live snapshot：系统运行状态下做的快照
- Cold snapshot：系统停止状态下的快照

libvit 做 snapshot 的各个 API：

snapshot	做快照的 libvirt API	从快照恢复的 libvirt API	virsh 命令
磁盘快照	virDomainSnapshotCreateXML（flags = VIR_DOMAIN_SNAPSHOT_CREATE_DISK_ONLY）	virDomainRevertToSnapshot	virsh snapshot-create/snapshot-revert
内存（状态）快照	virDomainSave virDomainSaveFlags virDomainManagedSave	virDomainRestore virDomainRestoreFlags virDomainCreate virDomainCreateWithFlags	virsh save/restore
系统检查点	virDomainSnapshotCreateXML	virDomainRevertToSnapshot	virsh snapshot-create/snapshot-revert

分别来看看这些 API 是如何工作的：

1. virDomainSnapshotCreateXML (virDomainPtr domain, const char * xmlDesc, unsigned int flags)


作用：根据 xmlDesc 指定的 snapshot xml 和 flags 来创建虚拟机的快照。

flags 包含	虚拟机处于运行状态时快照的做法	虚拟机处于关闭状态时快照的做法
0	创建系统检查点，包括磁盘状态和内存状态比如内存内容	保持关机时的磁盘状态
VIR_DOMAIN_SNAPSHOT_CREATE_LIVE	做快照期间，虚拟机将不会被 paused。这会增加内存 dump file 的大小，但是可以减少系统停机时间。部分 Hypervisor 只在做外部的系统检查点时才设置该 flag，这意味着普通快照还是需要暂停虚拟机。	
VIR_DOMAIN_SNAPSHOT_CREATE_DISK_ONLY	只做指定磁盘的快照。对应运行着的虚拟机，磁盘快照可能是不完整的（类似于突然电源被拔了的情形）。	只做指定磁盘的快照。

其内部实现根据虚拟机的运行状态有两种情形：

- 对运行着的虚拟机，API 使用 QEMU Monitor 去做快照，磁盘镜像文件必须是 qcow2 格式，虚拟机的 CPU 被停止，快照结束后会重新启动。
- 对停止着的虚拟机，API 调用 qemu-img 方法来操作所有磁盘镜像文件。

这里有其实现代码，可见其基本的实现步骤：



```
static virDomainSnapshotPtr qemuDomainSnapshotCreateXML
{
```

大数据(2)

分布式系统

服务器(1)

网络(11)

虚拟化(3)

云

文章档案(42)

2016年10月 (2)

2016年9月 (1)

2016年6月 (1)

2016年5月 (3)

2015年12月 (4)

2015年10月 (5)

2015年9月 (2)

2015年6月 (1)

2015年4月 (23)

积分与排名

积分 - 286831

排名 - 535

最新评论

1. Re:Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network]

eth1 - 公共网络 (untagged) , 管理网络 (tag=102) , 存储网络 (tag=103) 不好意思, 大家共用同一个eth1端口的时候, 请问这里交换机端口是配置为tagged还是untagged.....

--xianke9

2. Re:理解Docker (5) : Docker 网络

1.12版本上网络的表现如何?

--幽灵狼

3. Re:理解Docker (5) : Docker 网络

我想请问一下运行docker quickstart terminal时一直卡在"waiting for an IP"应该如何解决呢? 希望楼主能解答一下。

--silentbell

4. Re:理解Docker (6) : 若干企业生产环境中的容器网络方案

写得好好! 加油。

--itbj00

5. Re:理解Docker (5) : Docker 网络

非常好, 写得很详细。加油!

--itbj00

阅读排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](22087)

2. 理解 OpenStack 高可用 (HA) (1) : OpenStack 高可用和灾备方案 [OpenStack HA and DR](13707)

```
....
call qemuDomainSnapshotCreateDiskActive
{
    call qemuProcessStopCPUs # 停止 vCPUs
    for each disk call qemuDomainSnapshotCreateSingleDiskActive
    {
        call qemuMonitorDiskSnapshot # 调用 QEMU Monitor 去为每个磁盘做snapshot
    }
    call qemuProcessStartCPUs # 启动 vCPUs
}
....
}
```

2. virDomainSave 相关的几个 API

这几个API 功能都比较类似:

virDomainSave	该方法会 suspend 一个运行着的虚机, 然后保存内存内容到一个文件中。成功调用以后, domain 将不会处于 running 状态。使用 virDomainRestore 来恢复虚机。
virDomainSaveFlags	类似于 virDomainSave API, 可使用几个 flags。一些 Hypervisor 在调用该方法前需要调用 virDomainBlockJobAbort() 方法来停止 block copy 操作。
virDomainManagedSave	也类似于 virDomainSave API。主要区别是 libvirt 将其内存保存到一个受 libvirt 管理的文件中, 因此libvirt 可以一直跟踪 snapshot 的状态; 当调用 virDomainCreate/virDomainCreateWithFlags 方法重启该 domain的时候, libvirt 会使用该受管文件, 而不是一个空白的文件, 这样就可以 restore 该snapshot。

Features/SnapshotsMultipleDevices 这篇文章讨论同时对多个磁盘做快照的问题。

1.2 使用 virsh 实验

1.2.1 virsh save 命令

对运行中的 domain d-2 运行 "virsh save" 命令。命令执行完成后, d-2 变成 "shut off" 状态。

```
virsh # save d-2 d-2.snap1 --verbose
Save: [100 %]
Domain d-2 saved to d-2.snap1

virsh # virsh --all
error: unknown command: 'virsh'
virsh # list --all
  Id   Name                               State
-----
  10   d-1                               running
  -    d-2                               shut off
  -    rh64-1                           shut off
  -    rh64-2                           shut off
```

看看 domain 的磁盘镜像文件和 snapshot 文件:

```
[root@rh65 ~]# qemu-img info domains/rh64-9.qcow2
image: domains/rh64-9.qcow2
file format: qcow2
virtual size: 20G (21474836480 bytes)
disk size: 334M
cluster_size: 65536
backing file: rh61-2.img (actual path: domains/rh61-2.img)
[root@rh65 ~]# ls
anaconda-ks.cfg  Documents  install.log.syslog  Public  testlibvirt
d-1.snap         domains    isoimages          rh64-1.snap  Videos
d-2.snap1        Downloads  Music              rh64-1.snap1
Desktop          install.log Pictures
[root@rh65 ~]# qemu-img info d-2.snap1
image: d-2.snap1
file format: raw
virtual size: 424M (444960256 bytes)
disk size: 424M
```

http://www.cnblogs.com/sammyliu/p/4468757.html

3/14

3. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virtual Network](13434)
4. 探索 OpenStack 之 (9) : 深入块存储服务 Cinder (功能篇) (12921)
5. 理解 OpenStack + Ceph (1) : Ceph + OpenStack 集群部署和配置 (12444)

评论排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](63)
2. Neutron 理解 (14) : Neutron ML2 + Linux bridge + VxLAN 组网 (54)
3. Neutron 理解 (8): Neutron 是如何实现虚拟机防火墙的 [How Neutron Implements Security Group](34)
4. Neutron 理解 (3): Open vSwitch + GRE/VxLAN 组网 [Netruon Open vSwitch + GRE/VxLAN Virtual Network](25)
5. Neutron 理解 (5) : Neutron 是如何向 Nova 虚拟机分配固定 IP 地址的 (How Neutron Allocates Fixed IPs to Nova Instance) (21)

推荐排行榜

1. Neutron 理解 (1): Neutron 所实现的虚拟化网络 [How Netruon Virtualizes Network](9)
2. 我所了解的 京东、携程、eBay、小米 的 OpenStack 云(6)
3. 理解 OpenStack 高可用 (HA) (1) : OpenStack 高可用和灾备方案 [OpenStack HA and DR](6)
4. Neutron 理解 (2): 使用 Open vSwitch + VLAN 组网 [Netruon Open vSwitch + VLAN Virtual Network](6)
5. 理解 OpenStack 高可用 (HA) (2) : Neutron L3 Agent HA 之虚拟路由冗余协议 (VRRP) (5)

内存数据被保存到 raw 格式的文件中。

要恢复的时候, 可以运行 "vish restore d-2.snap1" 命令从保存的文件上恢复。

1.2.2 virsh snapshot-create/snapshot-create-as

先看看它的用法:



```
virsh # help snapshot-create-as
```

NAME

snapshot-create-as - Create a snapshot from a set of args

SYNOPSIS

```
snapshot-create-as <domain> [<name>] [<description>] [--print-xml]
[--no-metadata] [--halt] [--disk-only] [--reuse-external] [--quiesce]
[--atomic] [--live] [--memspec <string>] [--diskspec <string>]...
```

DESCRIPTION

Create a snapshot (disk and RAM) from arguments

OPTIONS

```
--domain] <string> domain name, id or uuid
--name] <string> name of snapshot
--description] <string> description of snapshot
--print-xml print XML document rather than create
--no-metadata take snapshot but create no metadata #创建的快照不带任何元数据
--halt halt domain after snapshot is created #快照创建后虚拟机机会关闭
--disk-only capture disk state but not vm state #只对磁盘做快照, 忽略其它参数
--reuse-external reuse any existing external files
--quiesce quiesce guest's file systems #libvirt 会通过 QEMU GA 尝试去freeze和unfreeze客户机已经mounted的文件系统; 如果客户机没有安装QEMU GA, 则操作会失败。
--atomic require atomic operation #快照要么完全成功要么完全失败, 不允许部分成果。不是所有的VMM都支持。
--live take a live snapshot #当客户机处于运行状态下做快照
--memspec <string> memory attributes: [file=]name[,snapshot=type]
[--diskspec <string> disk attributes: disk[,snapshot=type]
[,driver=type][,file=name]
```



其中一些参数, 比如 --atomic, 在一些老的 QEMU library 上不支持, 需要更新它到新的版本。根据 [这篇文章](#), atomic 应该是 QEMU 1.0 中加入的。

(1) 默认的话, 该命令创建虚拟机的所有磁盘和内存做内部快照, 创建快照时虚拟机处于 paused 状态, 快照完成后变为 running 状态。持续时间较长。

```
<memory snapshot='internal' />
<disks>
  <disk name='vda' snapshot='internal' />
  <disk name='vdb' snapshot='internal' />
  <disk name='vdc' snapshot='internal' />
</disks>
```



```
virsh # list --all
Id      Name      State
-----
10      d-1      running
13      d-2      paused
-       rh64-1   shut off
-       rh64-2   shut off

virsh # date
error: unknown command: 'date'
virsh #

virsh #
virsh # snapshot-create d-2
Domain snapshot 1434274252 created
virsh # snapshot-list d-2
Name      Creation Time      State
-----
1434273515 2015-06-14 17:18:35 +0800 running
1434273753 2015-06-14 17:22:33 +0800 running
1434274252 2015-06-14 17:30:52 +0800 running
```

每个磁盘的镜像文件都包含了 snapshot 的信息：

```
root@compute1:/var/lib/nova/instances/eddc46a8-e026-4b2c-af51-
dfaa436fcc7b# qemu-img info disk
image: disk
file format: qcow2
virtual size: 1.0G (1073741824 bytes)
disk size: 43M
cluster_size: 65536
backing file:
/var/lib/nova/instances/_base/fbad3d96a1727069346073e51d5bbb1824e76e34
Snapshot list:
ID      TAG      VM SIZE      DATE      VM
CLOCK
1      1433950148      41M 2015-06-10 23:29:08
05:16:55.007
Format specific information:
  compat: 1.1
  lazy refcounts: false
```

你可以运行 snapshot-revert 命令回滚到指定的snapshot。

```
virsh # snapshot-revert instance-0000002e 1433950148
```

根据 [这篇文章](#)，libvirt 将内存状态保存到某一个磁盘镜像文件内（"state is saved inside one of the disks (as in qemu's 'savevm'system checkpoint implementation). If needed in the future,we can also add an attribute pointing out _which_ disk saved the internal state; maybe disk='vda'.）

（2）可以使用 "--memspec" 和 "--diskspec" 参数来给内存和磁盘外部快照。这时候，在获取内存状态之前需要 Pause 虚机，就会产生服务的 downtime。

```
virsh # snapshot-create-as 0000002e livesnap2 --memspec
/home/s1/livesnap2mem,snapshot=external --diskspec
vda,snapshot=external
Domain snapshot livesnap2 created
virsh # snapshot-dumpxml 0000002e livesnap2
<memory snapshot='external' file='/home/s1/livesnap2mem' />
<disks>
  <disk name='vda' snapshot='external' type='file'>
    <driver type='qcow2' />
    <source file='/home/s1/testvm/testvm1.livesnap2' />
  </disk>
</disks>
```



(3) 可以使用 "--disk-only" 参数, 这时会做所有磁盘的外部快照, 但是不包含内存的快照。不指定快照文件名字的话, 会放在原来的磁盘文件所在的目录中。多次快照后, 会形成一个外部快照链, 新的快照使用前一个快照的镜像文件作为 backing file。



```
virsh # snapshot-list instance-0000002e --tree
1433950148 #内部快照
1433950810 #内部快照
1433950946 #内部快照
snap1 #第一个外部快照
|
+- snap2 #第二个外部快照
|
+- 1433954941 #第三个外部快照
|
+- 1433954977 #第四个外部快照
```



而第一个外部快照的镜像文件是以虚机的原始镜像文件作为 backing file 的 :



```
root@compute1:/var/lib/nova/instances/eddc46a8-e026-4b2c-af51-
dfaa436fcc7b# qemu-img info disk.snap1
image: disk.snap1
file format: qcow2
virtual size: 30M (31457280 bytes)
disk size: 196K
cluster_size: 65536
backing file: /var/lib/nova/instances/eddc46a8-e026-4b2c-af51-
dfaa436fcc7b/disk.swap #虚机的 swap disk 原始镜像文件
backing file format: qcow2
Format specific information:
  compat: 1.1
  lazy refcounts: false
```



目前还不支持回滚到某一个external disk snapshot, [这篇文章](#) 谈到了一个workaround。

```
[root@rh65 osdomains]# virsh snapshot-revert d-2 1434467974
error: unsupported configuration: revert to external disk snapshot not
supported yet
```

(4) 还可以使用 "--live" 参数创建系统还原点, 包括磁盘、内存和设备状态等。使用这个参数时, 虚拟机不会被 Paused (那怎么实现的?)。其后果是增加了内存 dump 文件的大小, 但是减少了系统的 downtime。该参数只能用于做外部的系统还原点 (external checkpoint)。



```
virsh # snapshot-create-as 0000002e livesnap3 --memspec
/home/s1/livesnap3mem,snapshot=external --diskspec
vda,snapshot=external --live
Domain snapshot livesnap3 created
virsh # snapshot-dumpxml 0000002e livesnap3
<memory snapshot='external' file='/home/s1/livesnap3mem' />
<disks>
  <disk name='vda' snapshot='external' type='file'>
    <driver type='qcow2' />
    <source file='/home/s1/testvm/testvm1.livesnap3' />
  </disk>
</disks>
```



注意到加 “--live” 生成的快照和不加这个参数生成的快照不会被链在一起：



```
virsh # snapshot-list 0000002e --tree
livesnap1 #没加 --live
|
+- livesnap2 #没加 --live

livesnap3 #加了 --live
|
+- livesnap4 #加了 --live
```



不过，奇怪的是，使用 QEMU 2.3 的情况下，即使加了 --live 参数，虚拟机还是会被短暂的 Paused 住：



```
[root@rh65 ~]# virsh snapshot-create-as d-2 --memspec /home/work/d-2/mem3,snapshot=external --diskspec hda,snapshot=external --live
Domain snapshot 1434478667 created

[root@rh65 ~]# virsh list --all
Id      Name      State
-----
 40     osvm1     running
 42     osvm2     running
 43     d-2       running

[root@rh65 ~]# virsh list --all
Id      Name      State
-----
 40     osvm1     running
 42     osvm2     running
 43     d-2       paused # 不是说好我用 --live 你就不
pause  虚拟机的么？这是肿了么。。

[root@rh65 ~]# virsh list --all
Id      Name      State
-----
 40     osvm1     running
 42     osvm2     running
 43     d-2       running
```



综上所述，对于 snapshot-create-as 命令来说，

参数	结果
<不使用额外的参数>	所有磁盘和内存的内部的内部快照
--memspec snapshot=external --diskspec vda,snapshot=external	磁盘和内存的外部快照，虚拟机需要被暂停
--live --memspec snapshot=external -- diskspec vda,snapshot=external	创建系统检查点（包括磁盘和内存的快照），而且虚拟机不会被暂停（？测试结果显示还是会暂停，只是暂停时间比不使用 --live 要短一些）
--disk-only	创建所有或者部分磁盘的外部快照

可以使用 snapshot-revert 命令来回滚到指定的系统还原点，不过得使用 “-force” 参数：

```
[root@rh65 ~]# virsh snapshot-revert d-2 1434478313
error: revert requires force: Target device address type none does not
match source pci

[root@rh65 ~]# virsh snapshot-revert d-2 1434478313 --force

[root@rh65 ~]#
```

1.3 外部快照的删除

目前 libvirt 还不支持直接删除一个外部快照，可以参考 [这篇文章](#) 介绍的 workaround。

2. OpenStack 中的快照

OpenStack Snapshot 可分为下面的几种情形：

2.1 对 Nova Instance 进行快照

(1) 对从镜像文件启动的虚拟机做快照

- 只将运行当中的虚拟机的 Root disk（第一个vd 或者 hd disk）做成 image，然后上传到 glance 里面
- Live Snapshot：对满足特定条件（QEMU 1.3+ 和 Libvirt 1.0.0+，以及 source_format not in ('lvm', 'rbd') and not CONF.ephemeral_storage_encryption.enabled and not CONF.workarounds.disable_libvirt_livesnapshot，以及能正常调用 libvirt.blockJobAbort，其前提条件可参考[这篇文章](#)）的虚拟机，会进行 Live snapshot。Live Snapshot 允许用户在虚拟机处于运行状态时不停机做快照。
- Cold Snapshot：对不能做 live snapshot 的虚拟机做 Cold snapshot。这种快照必须首先 Pause 虚拟机。

(2) 对从卷启动的虚拟机做快照

- 对虚拟机的每个挂载的 volume 调用 cinder API 做 snapshot。
- Snapshot 出的 metadata 会保存到 glance 里面，但是不会有 snapshot 的 image 上传到 Glance 里面。
- 这个 snapshot 也会出现在 cinder 的数据库里面，对 cinder API 可见。

2.2 对卷做快照

- 调用 cinder driver api，对 backend 中的 volume 进行 snapshot。
- 这个 snapshot 会出现在 cinder 的数据库里面，对 cinder API 可见。

3. 从镜像文件启动的 Nova 虚拟机做快照

严格地说，Nova 虚拟机的快照，并不是对虚拟机做完整的快照，而是对虚拟机的启动盘（root disk，即 vda 或者 hda）做快照生成 qcow2 格式的文件，并将其传到 Glance 中，其作用也往往是方便使用快照生成的镜像来部署新的虚拟机。Nova 快照分为 Live Snapshot（不停机快照）和 Clold Snapshot（停机快照）。

3.1 Nova Live Snapshot

满足 2.1.1 中所述条件时，运行命令 "nova image-create <instance name or uuid> <name of new image>" 后，Nova 会执行 Live Snapshot。其过程如下：

- 找到虚拟机的 root disk (vda 或者 hda)。
- 在 CONF.libvirt.snapshots_directory 指定的文件夹（默认为 /var/lib/nova/instances/snapshots）中创建一个临时文件夹，在其中创建一个 qcow2 格式的 delta 文件，其文件名为 uuid 字符串，该文件的 backing file 和 root disk 文件的 backing file 相同（下面步骤 a）。
- 调用 virDomainGetXMLDesc 来保存 domain 的 xml 配置。
- 调用 virDomainBlockJobAbort 来停止对 root disk 的活动块操作（Cancel the active block job on the given disk）。

5. 调用 `virDomainUndefine` 来将 domain 变为 `transient` 类型的, 这是因为 `BlockRebase` API 不能针对 `Persistent` domain 调用。
6. 调用 `virDomainBlockRebase` 来将 `root` disk image 文件中不同的数据拷贝到 `delta` disk file 中。(下面步骤 b)
7. 步骤 6 是一个持续的过程, 因为可能有应用正在向该磁盘写数据。Nova 每隔 0.5 秒调用 `virDomainBlockJobInfo` API 来检查拷贝是否结束。
8. 拷贝结束后, 调用 `virDomainBlockJobAbort` 来终止数据拷贝。
9. 调用 `virDomainDefineXML` 将 domain 由 `transient` 回到 `persistent`。
10. 调用 `qemu-img convert` 命令将 `delta` image 文件和 `backing file` 变为一个 `qcow2` 文件 (下面步骤 c)
11. 将 image 的元数据和 `qcow2` 文件传到 Glance 中。



(a) 执行 `qemu-img create -f qcow2` (`qemu-img create` 创建一个基于镜像1的镜像2, 镜像2的文件将基于镜像1, 镜像2中的文件将基于镜像1中的。在镜像2中所作的任何读写操作都不会影响到镜像1。 镜像1可以被其他镜像当做 `backing file`。但是要确保镜像1不要被修改)。比如:

```
qemu-img create -f qcow2 -o
backing_file=/var/lib/nova/instances/_base/ed39541b2c77cd7b069558570fa
1dff4fda4f678,size=21474836480
/var/lib/nova/instances/snapshots/tmpzfjdJS/7f8d11be9ff647f6b7a0a643fa
d1f030.delta
```

(b) 相当于执行 `virsh blockjob <domain> <path> [--abort] [--async] [--pivot] [--info] [<bandwidth>]`

(c) 执行 `'qemu-img convert -f qcow2 -o dest_fmt'` 来将带 `backing file` 的 `qcow2` image 转化成不带 `backing file` 的 `flat` image。其中 `dest_fmt` 由 `snapshot_image_format` 决定, 有效值是 `raw`, `qcow2`, `vmdk`, `vdi`, 默认值是 `source image` 的 `format`。比如: `qemu-img convert -f qcow2 -o qcow2 /var/lib/nova/instances/snapshots/tmpzfjdJS/7f8d11be9ff647f6b7a0a643fa d1f030.delta /var/lib/nova/instances/snapshots/tmpzfjdJS/7f8d11be9ff647f6b7a0a643fa d1f030`



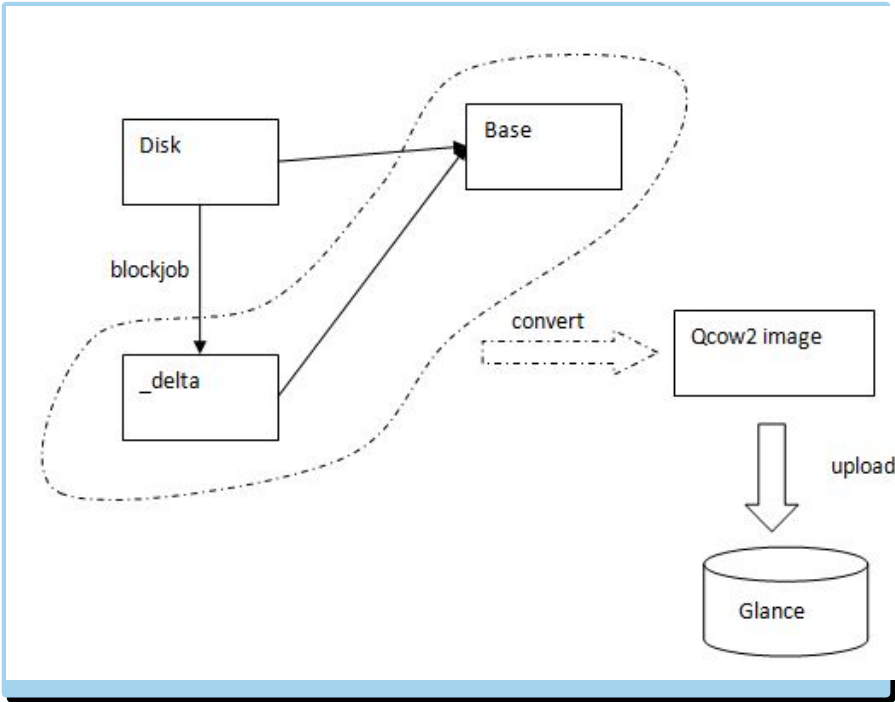
来看看其中的一个关键 API `int virDomainBlockRebase (virDomainPtr dom, const char * disk, const char * base, unsigned long bandwidth, unsigned int flags)`

该 API 从 `backing` 文件中拷贝数据, 或者拷贝整个 `backing` 文件到 `@base` 文件。Nova 中的调用方式为: `domain.blockRebase(disk_path, disk_delta, 0, libvirt.VIR_DOMAIN_BLOCK_REBASE_COPY | libvirt.VIR_DOMAIN_BLOCK_REBASE_REUSE_EXT | libvirt.VIR_DOMAIN_BLOCK_REBASE_SHALLOW)`

默认的话, 该 API 会拷贝整个 `@disk` 文件到 `@base` 文件, 但是使用

`VIR_DOMAIN_BLOCK_REBASE_SHALLOW` 的话就只拷贝差异数据 (`top data`) 因为 `@disk` 和 `@base` 使用相同的 `backing` 文件。 `VIR_DOMAIN_BLOCK_REBASE_REUSE_EXT` 表示需要使用已经存在的 `@base` 文件因为 Nova 会预先创建好这个文件。

简单的示意图:



[这里](#) 有个过程的 PoC 代码描述该过程。

[这里](#) 有该过程的完整 libvirt 日志分析。

[这里](#) 有文章讲 Libvirt Features/SnapshotsMultipleDevices。

3.2 Nova Cold Snapshot

当虚拟机不在运行中时或者不满足 live snapshot 的条件 的情况下，Nova 会执行 Cold snapshot。其主要过程如下：

- (1) 当虚拟机处于 running 或者 paused 状态时：
 - 1. detach PCI devices
 - 2. detach SR-IOV devices
 - 3. 调用 virDomainManagedSave API 来将虚拟机 suspend 并且将内存状态保存到磁盘文件中。
- (2) 调用 qemu-img convert 命令将 root disk 的镜像文件转化一个相同格式的镜像文件。
- (3) 调用 virDomainCreateWithFlags API 将虚拟机变为初始状态
- (4) 将在步骤1 中卸载的 PCI 和 SR-IOV 设备重新挂载回来
- (5) 将元数据和 qcow2 文件传到 Glance 中

4. 从 volume 启动的 Nova 实例的快照

(0) 从卷启动虚拟机，并且再挂载一个卷，然后运行 nova image-create 命令。

```
| image | Attempt to boot from volume -
no image supplied |
| key_name | -
| metadata | {}
| name | vm10
| os-extended-volumes:volumes_attached | [{"id": "26446902-5a56-4c79-b839-a8e13a66dc7a"}, {"id": "de127d46-ed92-471d-b18b-e89953c305fd"}]
```

- (1) 从 DB 获取该虚拟机的块设备 (Block Devices Mapping) 列表。
- (2) 对该列表中的每一个卷，依次调用 Cinder API 做快照。对 LVM Driver 的 volume 来说，执行的命令类似于 "lvcreate --size 100M --snapshot --name snap /dev/vg00/lvol1"。

```
s1@controller:~$ cinder snapshot-list
+-----+-----+-----+-----+
|          ID          |          Volume ID          |
| Status | Name | Size |
+-----+-----+-----+-----+
| a7c591fb-3413-4548-abd8-86753da3158b | de127d46-ed92-471d-b18b-e89953c305fd | available | snapshot for vm10-snap | 1 |
| d1277ea9-e972-4dd4-89c0-0b9d74956247 | 26446902-5a56-4c79-b839-a8e13a66dc7a | available | snapshot for vm10-snap | 1 |
+-----+-----+-----+-----+
```

(3) 将快照的 metadata 放到 Glance 中。(注：该 image 只是一些属性的集合，比如 block device mapping, kernel 和 ramdisk IDs 等，它并没有 image 数据, 因此其 size 为 0。)

```
s1@controller:~$ glance image-show e86cc562-349c-48cb-a81c-896584accde3
+-----+-----+
| Property | Value |
+-----+-----+
| Property 'bdm_v2' | True |
| Property 'block_device_mapping' | [{"guest_format": null, "boot_index": 0, "no_device": null, "snapshot_id": |
| # 分别是该虚拟机挂载的两个volume 的 | "d1277ea9-e972-4dd4-89c0-0b9d74956247", "delete_on_termination": null, |
| snapshot 的信息 | "disk_bus": "virtio", "image_id": null, "source_type": "snapshot", |
| | "device_type": "disk", "volume_id": null, "destination_type": "volume", |
| | "volume_size": null}, {"guest_format": null, "boot_index": null, "no_device": |
| | null, "snapshot_id": "a7c591fb-3413-4548-abd8-86753da3158b", |
| | "delete_on_termination": null, "disk_bus": null, "image_id": null, |
| | "source_type": "snapshot", "device_type": null, "volume_id": null, |
| | "destination_type": "volume", "volume_size": null}] |
| Property 'checksum' | 64d7c1cd2b6f60c92c14662941cb7913 |
| Property 'container_format' | bare |
| Property 'disk_format' | qcow2 |
| Property 'image_id' | bb9318db-5554-4857-a309-268c6653b9ff |
| Property 'image_name' | image |
| Property 'min_disk' | 0 |
| Property 'min_ram' | 0 |
| Property 'root_device_name' | /dev/vda |
| Property 'size' | 13167616 |
```

created_at	2015-06-10T05:52:24
deleted	False
id	e86cc562-349c-48cb-a81c-896584accde3
is_public	False
min_disk	0
min_ram	0
name	vm10-snap
owner	74c8ada23a3449f888d9e19b76d13aab
protected	False
size	0 # 这里 size 是 0, 表明该 image 只是元数据,
status	active
updated_at	2015-06-10T05:52:24
+-----+ -----+	

5. 当前 Nova snapshot 的局限

- Nova snapshot 其实只是提供一种创造系统盘镜像的方法。不支持回滚至快照点，只能采用该快照镜像创建一个新的虚拟机。
- 在虚拟机是从 image boot 的时候，只对系统盘进行快照，不支持内存快照，不支持系统还原点（blueprint：<https://blueprints.launchpad.net/nova/+spec/live-snapshot-vms>）
- Live Snapshot 需要用户进行一致性操作：<http://www.sebastien-han.fr/blog/2012/12/10/openstack-perform-consistent-snapshots/>
- 只支持虚拟机内置（全量）快照，不支持外置（增量）快照。这与当前快照的实现方式有关，因为是通过 image 进行保存的。
- 从 image boot 的虚拟机的快照以 Image 方式保存到 Glance 中，而非以 Cinder 卷方式保存。
- 过程较长（需要先通过存储快照，然后抽取并上传至 Glance），网络开销大。

那为什么 Nova 不实现虚拟机的快照而只是系统盘的快照呢？据说，社区关于这个功能有过讨论，讨论的结果是不加入这个功能，原因主要有几点：

- 这应该是一种虚拟化技术的功能，不是云计算平台的功能。
- openstack 由于底层要支持多种虚拟化的技术，某些虚拟化技术实现这种功能比较困难。
- 创建的 VM state snapshot 会面临 cpu feature 不兼容的问题。
- 目前 libvirt 对 QEMU/KVM 虚拟机的外部快照的支持还不完善，即使更新到最新的 libvirt 版本，造成兼容性比较差。

[这里](#) 也有很多的讨论。

分类: [Nova](#), [KVM](#), [基础知识](#), [虚拟化](#)

好文要顶 关注我 收藏该文 微博 微信



SammyLiu
关注 - 30
粉丝 - 470


荣誉: 推荐博客
[+加关注](#)

4 推荐

0 反对


« 上一篇: [Nova: 虚拟机的块设备总结 \[Nova Instance Block Device\]](#)
» 下一篇: [KVM 介绍 \(8\) : 使用 libvirt 迁移 QEMU/KVM 虚拟机和 Nova 虚拟机 \[Nova Libvirt QEMU/KVM Live Migration\]](#)

评论:

#1楼 2015-10-08 10:24 | [hojanelson](#) 


寫得很詳細，我也被--live弄到暈了

支持(0) 反对(0)

#2楼 2015-11-27 16:42 | [zhuo.wang](#) 

写的很详细。请问如果想要用OPENSTACK实现虚拟机重启后自动还原的功能有什么好办法吗，


支持(0) 反对(0)

#3楼[楼主] 2015-11-27 19:54 | [SammyLiu](#) 

@ zhuo.wang

你能详细描述下“重启后自动还原”是什么意思吗？

支持(0) 反对(0)

#4楼 2015-11-30 10:08 | [zhuo.wang](#) 


@ SammyLiu

是这样的，是想做个设置，让虚拟机在重启后恢复到最初的状态。就是那种还原精灵那种功能。

在VBOX中可以设置磁盘不可变，来实现这个功能。用VIRT-MANAGER看虚拟机时候，发现可以设置磁盘为不可变，但是设置后虚拟机就无法启动了。

实在不行了我就通过VM里安装还原精灵来实现这个功能吧。


支持(0) 反对(0)

#5楼[楼主] 2015-11-30 15:59 | [SammyLiu](#) 

@ zhuo.wang

看起来你是想实现类似于休眠后恢复这样的功能，据我所知libvirt 或者 kvm 本身应该是不支持的，你需要借助第三方工具来实现。

支持(0) 反对(0)


#6楼 2015-12-15 14:54 | [jython.li](#) 

@ SammyLiu

楼主,qemu本身是可以实现的,在启动虚拟机时加上"-snapshot"参数,现有镜像变为一个backing file,虚拟机中所有的读写操作会被存放在一个Temporary镜像,重启后Temporary镜像会被清空,这个文章的最后一段有介绍

<http://wiki.qemu.org/Documentation/CreateSnapshot>


支持(0) 反对(0)

#7楼[楼主] 2015-12-15 15:26 | [SammyLiu](#) 

@ jython.li

谢谢。希望对 @zhuo.wang 有帮助。


支持(0) 反对(0)

#8楼 2016-03-27 00:06 | [manigete](#) 

virsh做在线快照，增加了--live参数仍然会Pause是因为，在线快照的代码实际上调用了live migration的代码块进行内存的拷贝，live migration在迁移的最后会有短暂的暂停，因此live snapshot也存在暂停。

qemu的官网上和社区中看到的。

支持(0) 反对(0)


#9楼[楼主] 2016-03-27 10:20 | [SammyLiu](#) 

@ [manigete](#)

谢谢

[支持\(0\)](#) [反对\(0\)](#)

[刷新评论](#) [刷新页面](#) [返回顶部](#)

 注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【报表】Excel 报表开发18 招式，人人都能做报表

【活动】阿里云海外云服务全面降价助力企业全球布局

【实用】40+篇云服务器操作及运维基础知识！



最新IT新闻:

- 知乎上线视频功能，以后看教程更方便了
- 一年只赚2万元：乐视游戏或被出售
- Unity获得4亿美元投资，现估值为26亿美元
- 直播对陌陌的意义，就像王者荣耀之于腾讯游戏
- 死磕支付宝？苏宁金融发布“星辰计划”：扫码支付返888元

» [更多新闻...](#)



最新知识库文章:

- 程序员的工作、学习与绩效
- 软件开发为什么很难
- 唱吧DevOps的落地，微服务CI/CD的范本技术解读
- 程序员，如何从平庸走向理想？
- 我为什么鼓励工程师写blog

» [更多知识库文章...](#)

Powered by: [博客园](#) 模板提供: [沪江博客](#) Copyright ©2017 SammyLiu