

UniCloud Usphere 云计算管理平台

应急故障恢复指导

资料版本：5W100-20230717

Copyright © 2023 新华三技术有限公司 版权所有，保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

除新华三技术有限公司的商标外，本手册中出现的其它公司的商标、产品标识及商品名称，由各自权利人拥有。

本文档中的信息可能变动，恕不另行通知。

目 录

1 概述	3
2 应急恢复流程和原则	3
2.1 应急恢复流程	3
2.2 应急故障处理原则	4
2.2.1 应急处理原则	4
2.2.2 业务快速恢复原则	4
2.3 应急日常准备	4
3 应急故障恢复方法	5
3.1 应急恢复流程	5
3.2 判断故障类型	5
3.3 常用应急恢复方法	6
3.4 计算相关异常	6
3.4.1 主机无法正常启动	6
3.4.2 系统安装卡住或者安装后无 xconsole 界面	6
3.4.3 虚拟机启动失败	7
3.4.4 虚拟机无法正常启动，报“Could not read snapshots: File too large”错误	8
3.4.5 虚拟机重启报错，找不到启动盘	9
3.4.6 虚拟机迁移长时间不能完成	10
3.4.7 虚拟机部署部分成功，错误码信息为 7501	11
3.4.8 虚拟机使用云彩虹迁移失败，报“文件拷贝错误”错误	11
3.4.9 CPU 为海光 7280 的主机，Centos5/6.8 虚拟机启动时报 soft lockup 错误	12
3.5 存储相关异常	13
3.5.1 主机磁盘 IO 延迟较大造成虚拟机卡顿	13
3.5.2 共享存储池启动失败	13
3.5.3 共享存储阻塞导致集群不可用	14
3.5.4 主机存储 Fence 重启	16
3.5.5 共享文件系统的强制修复	17
3.5.6 启动共享存储池时失败，提示“存储池没有可用的 slot”	17
3.5.7 界面添加共享存储时报配置多路径失败	18
3.6 网络相关异常	19
3.6.1 主机或者虚拟机网络出现丢包或者业务不通	19
3.6.2 虚拟机 IP 地址冲突	23
3.6.3 主备聚合丢包问题	24

3.6.4	配置 ACL 导致虚拟机无法访问网关地址	26
3.6.5	虚拟防火墙导致 SSH 连接无法建立	27
3.6.6	主机中毒对外部发起攻击	29
3.6.7	银河麒麟操作系统的虚拟机配置 DNS 后，重启主机，配置失效	30
3.6.8	Ubuntu 20 虚拟机配置 IPv4 地址，界面报错：修改虚拟机 xxx 配置失败	30
3.7	故障告警	30
3.7.1	主机网卡故障	30
3.7.2	主机 MCE 故障告警	31
3.8	其他异常	31
3.8.1	界面无法正常登录	31
3.8.2	前台界面无法正常打开，日志中报连接数据库失败	32
3.8.3	前台界面查看虚拟机、存储池等信息卡住	33
3.8.4	虚拟机内部操作卡顿问题	33
3.8.5	USB 插到 VKS 主机上后，主机无法识别到该设备	34
3.8.6	修改系统时间导致集群开启 HA 失败，报“HA 进程响应超时”	35
4	故障信息收集	35
4.1	在系统后台收集日志信息	35
4.2	在管理平台中收集日志文件	35
4.3	查看故障告警信息	36

1 概述

本文为 UniCloud Usphere 服务器虚拟化常见故障应急解决方案，目的是帮助在一些突发情况下尽快恢复业务。本手册主要适用于：现场技术支持与维护人员、网络管理员。

由于设备不同型号、不同配置、不同版本等原因，本手册中的内容可能与用户使用的设备显示信息不一致。实际使用中请以设备显示的内容为准。我们会在后续使用过程中，持续改进文档，力求做到更好地为客户、为生产、为现场服务提供最佳的指导。如果您在使用过程中发现任何问题，请联系技术支持并反馈！

2 应急恢复流程和原则

2.1 应急恢复流程

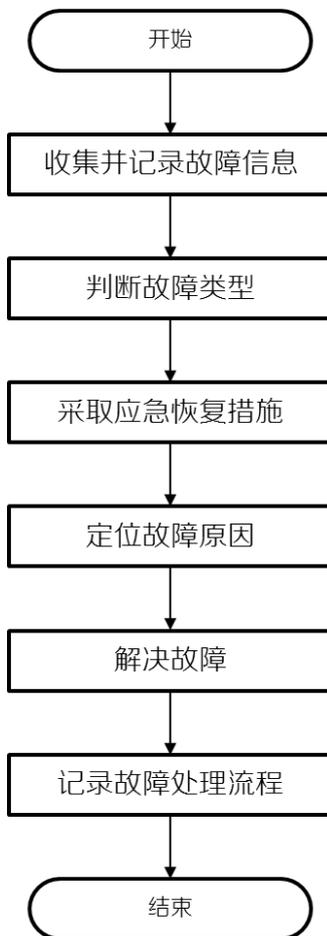
应急处理以快速恢复设备的正常运行和维持业务的正常提供为核心指导思想，其总体处理流程如下图所示。



说明

在出现需要应急恢复的故障时，应先尝试采取应急恢复措施，尝试恢复业务。无论是否恢复成功，均应在后续进行故障定位并彻底解决。

图1 故障处理流程图



2.2 应急故障处理原则

2.2.1 应急处理原则

重大故障很容易导致大面积的用户虚拟机故障、设备瘫痪等严重后果，具有很大的危害性。为提高重大故障的处理效率、并尽最大的限度降低此类故障的损失，在维护本设备之前，应充分考虑并遵循以下应急处理的基本原则。

- 应急处理以快速恢复设备的正常运行与业务的提供为核心，因此，提高重大故障处理效率的关键是：客户应参考应急处理手册及时制定各种重大故障的处理预案，并定期组织相关管理人员与维护人员进行学习、演练。
- 以客户业务尽快恢复，对客户影响最低为原则。在此前提下，进行问题定位恢复和数据收集。
- 维护人员在上岗前必须接受必要的应急处理培训，学习判断重大故障的基本方法、掌握处理重大故障的基本技能。
- 在重大故障的处理过程中，维护人员应及时联系新华三公司客户服务中心或新华三公司驻当地办事处，以便能够快速获取新华三公司的技术支持。
- 当维护人员完成重大故障的处理以后，应该及时采集与本次故障有关的设备故障告警信息，并将相关的故障处理报告、设备告警文件、日志文件等发送给新华三公司进行分析与定位，以便新华三公司能够更好地为客户提供售后服务。

2.2.2 业务快速恢复原则

业务恢复应综合考虑相应操作恢复业务成功的可能性和相应操作时间代价。参考的操作排序如下：

- (1) 耗时比较短，成功可能性比较大的操作。
- (2) 耗时比较短，成功可能性比较小的操作。
- (3) 耗时比较长，成功可能性比较大的操作。

2.3 应急日常准备

表1 应急日常准备

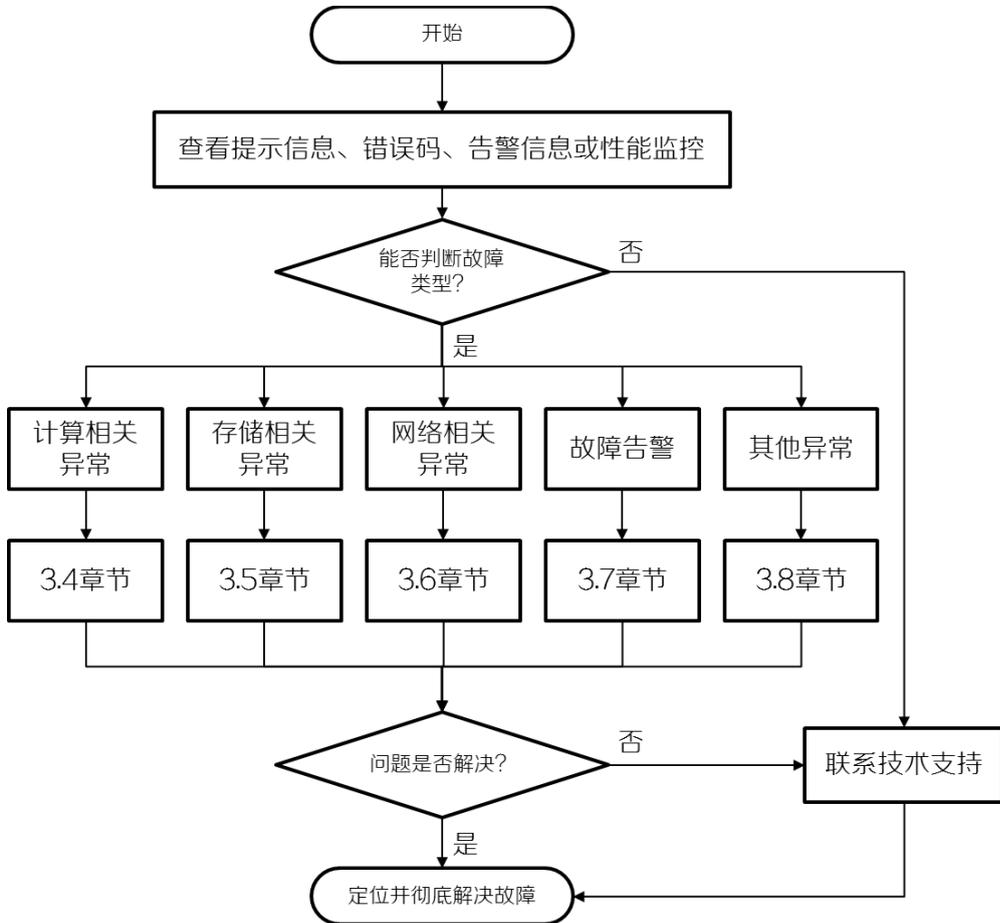
类别	应急日常准备
设备级备份	<ul style="list-style-type: none">• 主备用设备要求：定期进行数据一致性检查，以及运行状态检查，确保应急时能够接管业务。• 负荷分担设备要求：定期进行负荷评估，评估业务单平面运行性能评估，确保单点故障业务可以全部由另一个设备接管。
容灾（可选）	容灾局及相关切换准备。
日常告警清理	关键设备需要常备备件。
基本信息	日常告警需要及时处理，确保没有未确认的活动告警，避免出现问题时，信息混乱，影响事故处理中的判断决策。
人员要求	维护人员需要准备以下基本信息： <ul style="list-style-type: none">• 组网信息• 设备基础信息• 软件列表• 网络设备 IP 地址信息• 业务信息• 备件信息• 远程维护信息

类别	应急日常准备
	<ul style="list-style-type: none"> 相关接口人

3 应急故障恢复方法

3.1 应急恢复流程

图2 应急恢复流程图



3.2 判断故障类型

观察故障状态并判断故障类型，故障类型包括：

- 计算相关异常：主要包括主机异常，虚拟机异常等。例如主机无法启动，虚拟机无法启动，迁移无法完成等情况。
- 存储相关异常：主要是共享存储池异常，例如磁盘 IO 异常，主机存储异常，存储池配置失败、不可用等情况。
- 网络相关异常：主要包括主机或虚拟机网络不可用，配置不正确，丢包率高等情况。
- 故障告警：明确的故障告警信息。
- 其他异常：主要包括管理平台不可用、外设不可用等。例如管理平台无法登陆，无法升级，系统时间异常等情况。

故障类型可通过如下信息进行判断：

- 查看任务执行失败的提示信息。

- 查看任务台任务执行情况。
- 查看管理平台中的告警信息。
- 观察虚拟机或主机性能监控信息。

3.3 常用应急恢复方法

在主机或虚拟机出现应急故障时，可以尝试采用如下方法尝试恢复业务。

- 重启虚拟机：在管理平台或 SSH 至虚拟机后台尝试重新启动虚拟机。
- 重启业务主机：请联系 UniCloud 技术支持工程师处理。
- 重启网络设备：重启故障相关的物理交换机，路由器等网络设备，观察链路情况。
- 在管理主机后台重启如 tomcat8 等服务，各服务对应的模块详情可参考维护手册或联系技术支持。

3.4 计算相关异常

3.4.1 主机无法正常启动

1. 故障现象

系统硬件检测完成后，报错 “No bootable device”。

```
iPXE 1.0.0+ -- Open Source Network Boot Firmware -- http://ipxe.org
Features: iSCSI HTTP DNS TFTP AoE bzImage ELF MBOOT PXE Menu PXEXT

net0: 0c:da:41:1d:ca:ad using virtio-net on PCI00:03.0 (open)
  [Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 0c:da:41:1d:ca:ad)..... Error 0x040ee119 (http://ipxe.org/040ee119)
No more network devices

No bootable device.
```

2. 故障影响

主机无法正常启动。

3. 故障原因

- 服务器没有安装操作系统或服务器没有引导磁盘设备。
- 服务器引导模式发生过变更。
- RAID 卡损坏或者 RAID 卡驱动不匹配。

4. 恢复方法

- 检查是否存在磁盘设备，或者磁盘是否损坏，联系硬件修复，然后再安装系统。
- 若服务器引导模式发生过变更，需要重新安装系统。
- 若 RAID 卡损坏，联系硬件厂家修复，若 RAID 卡驱动不匹配，需联系技术支持进行驱动适配。

3.4.2 系统安装卡住或者安装后无 xconsole 界面

1. 故障现象

- 系统安装过程中卡死或黑屏。
- 安装完成后无 xconsole 界面。

2. 故障影响

系统无法正确安装、使用。

3. 故障原因

- 由于网络状态的原因，安装过程中出现中断。
- 用于安装的 U 盘格式不对。

4. 恢复方法

- 使用飞腾 CPU 服务器，无 xsconsole 界面。
- 使用 Linux DD 方式制作 U 盘安装盘，命令格式为：`dd if=/root/CAS-*.iso of=/dev/sdc bs=1M`。

3.4.3 虚拟机启动失败

1. 故障现象

虚拟机无法正常启动。

2. 故障影响

虚拟机无法启动，用户业务无法正常使用。

3. 故障原因

- CPU 或内存资源不足，检查物理服务器的 CPU 和内存资源占用情况。管理平台是否开启了“主机内存预留”功能，该功能会增加一个限制，如果物理主机内存+虚拟机内存>80%，后续虚拟机就不让启动了。



- Libvirt 所做的资源检查不通过，导致无法启动。这种启动失败会记录在 Libvirt 的日志中，Libvirt 的内存门限在如下配置文件中配置，如下所示。

```
cat /etc/cvk/cpu_mem_threshold.conf
CpuUsePercent 85 MemSysReserved 4
```

- 卸载亚信杀毒后，存在虚拟机配置文件残留配置，导致虚拟机无法启动。

4. 恢复方法

- 如果是由于资源不够导致的问题，请手动释放 CPU 或者内存资源；如果是由于资源超配导致的问题，可以暂时关闭不使用的虚拟机或将其他虚拟机迁移至另外的主机以释放资源。
- 如果是配置文件残留，查看虚拟机日志有如下打印：

```
2017-05-18 11:01:52.617+0000: 29917: error : qemuProcessWaitForMonitor:1852 : internal
error process exited while connecting to monitor: 2017-05-18 11:01:52.414+0000: domain
pid 32504, created by libvirtd pid 2974
char device redirected to /dev/pts/8 (label charserial0)
kvm: -chardev socket,id=charshmem0,path=/tmp/kvmsec: Failed to connect to socket: No
such file or directory
kvm: -chardev socket,id=charshmem0,path=/tmp/kvmsec: chardev: opening backend "socket"
failed
```

- 查看虚拟机配置，包含如下内容：

```
<shmem name='nu_fsec-4af99204-6623-45e9-afbf-d852746bf187'>
  <size unit='M'>8</size>
  <server path='/tmp/kvmsec' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x09' function='0x0' />
</shmem>
```

这种情况的解决方案为：删除 xml 标签，然后重新启动虚拟机。

3.4.4 虚拟机无法正常启动，报“Could not read snapshots: File too large”错误

1. 故障现象

查看虚拟机日志，存在如下内容：

```
error: Failed to start domain centos
error: internal error process exited while connecting to monitor: 2016-08-02 19:52:42.707+0000:
domain pid 31971, created by libvirtd pid 3434
char device redirected to /dev/pts/5 (label charserial0)
kvm: -drive
file=/vms/share/Centos6.5-tf.5-tf,if=none,id=drive-virtio-disk0,format=qcow2,cache=directsync: could not open disk image /vms/share/Centos6.5-tf.5-tf: Could not read snapshots: File too large
```

2. 故障影响

用户业务受此影响无法正常使用。

3. 故障原因

虚拟机镜像文件快照信息损坏。

4. 恢复方法

- (1) 用 `qcow2.py` 工具将 `qcow2` 头文件中的快照信息修改为 0。使用的命令包括：`qcow2.py dump-header`、`qcow2.py set-header nb_snapshots 0` 以及 `qcow2.py set-header snapshot_offset 0x0`，示例如下。

```
root@mi-service:~# qcow2.py /vms/share/Centos6.5-tf.5-tf dump-header
magic                0x514649fb
version              3
backing_file_offset  0x0
backing_file_size    0x0
cluster_bits        21
size                 26843545600
crypt_method         0
l1_size              1
l1_table_offset      0x600000
refcount_table_offset 0x200000
refcount_table_clusters 1
nb_snapshots         4
snapshot_offset      0x130600000
incompatible_features 0x0
compatible_features  0x0
autoclear_features   0x0
refcount_order       4
header_length        104

root@mi-service:~# qcow2.py /vms/share/Centos6.5-tf.5-tf set-header nb_snapshots 0
root@mi-service:~# qcow2.py /vms/share/Centos6.5-tf.5-tf set-header snapshot_offset 0x0
root@mi-service:~# qcow2.py /vms/share/Centos6.5-tf.5-tf dump-header
magic                0x514649fb
version              3
backing_file_offset  0x0
backing_file_size    0x0
cluster_bits        21
size                 26843545600
crypt_method         0
l1_size              1
l1_table_offset      0x600000
refcount_table_offset 0x200000
refcount_table_clusters 1
nb_snapshots         0
```

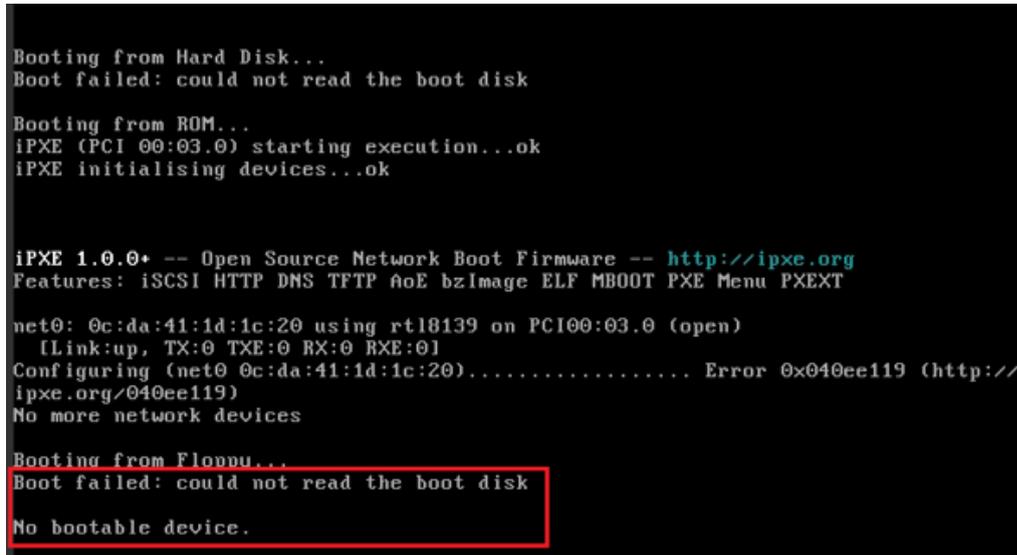
```
snapshot_offset      0x0
incompatible_features 0x0
compatible_features  0x0
autoclear_features   0x0
refcount_order       4
header_length        104
```

(2) 操作完成后重新启动虚拟机。

3.4.5 虚拟机重启报错，找不到启动盘

1. 故障现象

重启虚机时报错，找不到启动盘。



```
Booting from Hard Disk...
Boot failed: could not read the boot disk

Booting from ROM...
iPXE (PCI 00:03.0) starting execution...ok
iPXE initialising devices...ok

iPXE 1.0.0+ -- Open Source Network Boot Firmware -- http://ipxe.org
Features: iSCSI HTTP DNS TFTP AoE bzImage ELF MBOOT PXE Menu PXEXT

net0: 0c:da:41:1d:1c:20 using rtl8139 on PCI00:03.0 (open)
[Link:up, TX:0 TXE:0 RX:0 RXE:0]
Configuring (net0 0c:da:41:1d:1c:20)..... Error 0x040ee119 (http://ipxe.org/040ee119)
No more network devices

Booting from Floppy...
Boot failed: could not read the boot disk
No bootable device.
```

2. 故障影响

用户业务受此影响无法正常使用。

3. 故障原因

查看虚拟机日志有如下字段：

```
qcow2: Preventing invalid write on metadata (overlaps with snapshot table); image marked as corrupt.
```

使用 `qemu-img check XXX`（XXX 代表磁盘镜像）命令检查该虚机的磁盘镜像有报错，如下图：

```
root@LSSZF- :~# qemu-img check /vms/SG_IPSAN/mzgk.img
Leaked cluster 263448 refcount=1 reference=0
Leaked cluster 263465 refcount=1 reference=0
Leaked cluster 271515 refcount=1 reference=0
ERROR cluster 274639 refcount=0 reference=1
Leaked cluster 274955 refcount=2 reference=1
Leaked cluster 275507 refcount=1 reference=0
Leaked cluster 283000 refcount=1 reference=0
Leaked cluster 283216 refcount=1 reference=0
Leaked cluster 308097 refcount=1 reference=0
Leaked cluster 317849 refcount=1 reference=0
Leaked cluster 319317 refcount=1 reference=0
Leaked cluster 320284 refcount=1 reference=0
Leaked cluster 324133 refcount=1 reference=0

1 errors were found on the image.
Data may be corrupted, or further writes to the image may corrupt it.

12 leaked clusters were found on the image.
This means waste of disk space, but no harm to data.
264974/4569248 = 5.80% allocated, 4.59% fragmented, 0.00% compressed clusters
Image end offset: 21894463488
root@LSSZF- :~# qemu-img info /vms/SG_IPSAN/mzgk.img
image: /vms/SG_IPSAN/mzgk.img
file format: qcow2
virtual size: 279G (299450236928 bytes)
disk size: 20G
cluster size: 65536
```

4. 恢复方法

执行 `qemu-img check -r all xxx` (XXX 代表磁盘镜像) 命令恢复镜像文件, 恢复成功后, 虚拟机即可启动。



注意

修复之前, 需要对于文件备份, 以便后续可能等多次修复等。

3.4.6 虚拟机迁移长时间不能完成

1. 故障现象

- 虚拟机迁移卡住, 长时间不能完成, 进度显示为 99%。
- 虚拟机详情页中的虚拟机迁移速度为 0, 保持 15 分钟以上。

2. 故障影响

用户业务可能受影响。

3. 故障原因

- 磁盘文件较大, 管理网带宽为千兆网, 所以磁盘迁移耗时较长。
- 业务繁忙, 内存产生脏数据较快, 内存迁移无法收敛。
- 存储故障, 虚拟机被冻结。
- 迁移网络中断。

4. 恢复方法

- 查看目的磁盘文件大小, 并等待磁盘迁移完成。
- 暂停虚拟机内部业务, 或等待业务压力减小后再继续。
- 检查虚拟机所使用的存储池是否正常, 如有需要可终止目的端 KVM 进程来结束迁移任务。
- 如果是迁移网络中断, 使用 `virsh list` 命令观察源端以及目的端虚拟机的状态, 此时源端应为 `pause`, 目的端为 `running`, 使用 `virsh destroy` 命令关闭源端以及目的端虚拟机, 使

用 `systemctl restart libvirtd` 命令重启 `libvirtd` 服务，使用 `virsh undefined` 命令注销源端虚拟机，使用 `virsh start` 命令启动目的端虚拟机，并恢复相关业务。

3.4.7 虚拟机部署部分成功，错误码信息为 7501

1. 故障现象

在管理平台中部署虚拟机，结果为部分部署成功，错误码为 7501。

2. 故障影响

虚拟机没有正确部署，可能影响正常使用。

3. 故障原因

错误码 7501 一般都是由于 `uspheretools.py` 执行失败引起的，可以在日志 `/var/log/uspheretools.log` 中查看具体故障原因。

目前比较常见的原因有：

- 虚拟机磁盘损坏，`libguestfs` 会检查磁盘分区等信息，磁盘损坏则无法挂载。
- 虚拟机没有正确安装 `Uspheretools`，虚拟机内部 `qemu-ga` 服务异常。

4. 恢复方法

- 在主机后台执行 `qemu-img check -r all xxx` (`xxx` 代表磁盘镜像) 命令恢复镜像文件。
- 重新安装 `Uspheretools`。

3.4.8 虚拟机使用云彩虹迁移失败，报“文件拷贝错误”错误

1. 故障现象

虚拟机从 E0710P11 及之前的版本中通过云彩虹迁移至 E0718 及之后的版本中，出现部分虚拟机迁移任务失败，报错“文件拷贝错误”。

2. 故障影响

虚拟机无法使用云彩虹在线迁移。

3. 故障原因

查看 `cas.log` 时可以看到 SCP 任务执行失败，错误信息显示为“Bad SSH2 cipher spec 'arcfour128,aes128-ctr,aes192-ctr,aes256-ctr'”。

```
2022-08-10 17:14:00 [ERROR] [Domain Request Processor Manager 10] [com.virtual.plat.server.vm.ssh.ScpUtil::copyStorage]
=====SCP : scp -o Ciphers=arcfour128,aes128-ctr,aes192-ctr,aes256-ctr root@8.8.8.8:/vms/MacroSAN-
37T/test_inc_1" /vms/ONESTor-CAS-32T-1/test_inc_1"
2022-08-10 17:14:00 [ERROR] [Domain Request Processor Manager 10]
[com.virtual.plat.server.vm.ssh.SecureShellManagerImpl::execCmd] Execute command Error Msg:command-line line 0: Bad SSH2
cipher spec 'arcfour128,aes128-ctr,aes192-ctr,aes256-ctr'.
```

自 E0718 版本起，为了解决 `openssh` 漏洞升级了 `openssh` 的版本。因此，加密算法参数发生了变化。但老版本的云彩虹迁移接口下发的参数仍会采用“`scp -o Ciphers=xxx`”的方式。这种情况在 E0718 之后的版本上会执行失败。

在线迁移时，若虚拟机仅存在单级磁盘文件，则通过 `libvirt` 进行迁移，不会触发 SCP 任务。而对于离线迁移虚拟机，或磁盘文件存在多级镜像文件、快照的情况下，会触发 SCP 任务，从而导致云彩虹任务失败。

4. 恢复方法

合并虚拟机镜像文件后，采用在线迁移即可。

3.4.9 CPU 为海光 7280 的主机，Centos5/6.8 虚拟机启动时报 soft lockup 错误

1. 故障现象

在海光 C86 7280 32-Core CPU 环境下,使用 Centos6.5/6.8 虚拟机进行启动时,会出现“soft lockup”错误。

2. 故障影响

虚拟机无法正常启动。

3. 故障原因

在抓取虚拟机的 kernel dump 时,发现在启动过程中可能会出现“soft lockup”,并且这种情况发生在不同的应用程序中。值得注意的是,这些应用程序在运行到“exception RIP: get_random_bytes+41”时都无法返回。

```
PID: 1208 TASK: ffff8802374ff540 CPU: 0 COMMAND: "modprobe"
[exception RIP: get_random_bytes+41]
RIP: ffffffff81331419 RSP: ffff880236b6fcf8 RFLAGS: 00000203
RAX: ffffffff81331419 RBX: ffff880237f79cf0 RCX: 000000000000000a
RDX: 00000000000000ff RSI: 0000000000000008 RDI: ffff880237f79cf0
RBP: ffff880236b6fd38 R8: 0000000000000000 R9: 0000000000000180
R10: 0000000000000000 R11: 0000000000000000 R12: 0000000000000008
R13: ffff880237f79ce4 R14: ffff880237f79df0 R15: 0000000000000001
CS: 0010 SS: 0018
#0 [ffff880236b6fd40] __ipv6_regen_rndid at ffffffff81559ed [ipv6]
#1 [ffff880236b6fd60] ipv6_regen_rndid at ffffffff8157240 [ipv6]
#2 [ffff880236b6fd80] ipv6_add_dev at ffffffff815753e [ipv6]
#3 [ffff880236b6fdb0] addrconf_notify at ffffffff815a8e9 [ipv6]
#4 [ffff880236b6fe90] register_netdevice_notifier at ffffffff8145d9bf
#5 [ffff880236b6fee0] addrconf_init at ffffffff819e3be [ipv6]
#6 [ffff880236b6ff00] init_module at ffffffff819e198 [ipv6]
#7 [ffff880236b6ff20] do_one_initcall at ffffffff8100204c
#8 [ffff880236b6ff50] sys_init_module at ffffffff810bc291
#9 [ffff880236b6ff80] system_call_fastpath at ffffffff8100b072
RIP: 00007efef7b33eda RSP: 00007ffeffffb808 RFLAGS: 00010287
RAX: 00000000000000af RBX: ffffffff8100b072 RCX: 0000000000000030
RDX: 000000000261e6e0 RSI: 000000000008ad20 RDI: 00007efef7e12010
RBP: 00007ffeffffb00 R8: 00007efef7e9cd30 R9: 00007efef7ff7700
R10: 00007ffefffb770 R11: 0000000000000206 R12: 000000000261e6e0
R13: 0000000002625250 R14: 000000000261d550 R15: 000000000261e6e0
ORIG_RAX: 00000000000000af CS: 0033 SS: 002b
```

经过分析,发现 get_random_bytes()函数最终是通过 CPU 指令 rdrand 来获取随机数的。在虚拟机运行在主机匹配模式、直通模式下时,可能会存在 rdrand flag 问题。这主要是因为 AMD CPU 的 rdrand 指令不够稳定。此外,虚拟机所使用的 CentOS6.8 内核版本较老(为 2.6.32),而该内核版本的 Linux 社区已经停止使用 rdrand 特性。

4. 恢复方法

为了解决上述问题,可以在虚拟机的 XML 文件中修改 CPU 部分,将 rdrand 特性禁用即可。

```
<features>
  <acpi/>
  <apic/>
  <paе/>
</features>
<cpu mode='host-model' check='partial'>
  <topology sockets='40' dies='1' cores='4' threads='1'/>
  <feature policy='disable' name='rdrand'/>
  <numa>
    <cell id='0' cpus='0-159' memory='8388608' unit='KiB' memAccess='shared'/>
  </numa>
</cpu>
<clock offset='utc'/>
<on_poweroff>destroy</on_poweroff>
<on_reboot>restart</on_reboot>
<on_crash>coredump-restart</on_crash>
</vm>
```

3.5 存储相关异常

3.5.1 主机磁盘 IO 延迟较大造成虚拟机卡顿

1. 故障现象

在管理平台中，可以看到主机性能监控处磁盘 IO 延迟较大，虚拟机出现了卡顿现象。此外，在 ping 虚拟机时网络出现了抖动，并且甚至出现了丢包的现象。

2. 故障影响

虚拟机无法正常使用，影响用户业务。

3. 故障原因

存储池 IO 吞吐量较大，并且 IOPS 较高。

4. 恢复方法

针对虚拟机业务 IO 压力较大的情况，可以考虑将虚拟机迁移到其他存储池中。如果必要的话，也可以对虚拟机的 IO 进行适当限制。如下图所示。



3.5.2 共享存储池启动失败

1. 故障现象

启动共享存储池时失败，报相应的错误信息。

2. 故障影响

用户无法启动共享存储池，业务无法正常使用。

3. 故障原因

可能的原因可能是现场网络或存储的配置的原因。因此，我们可以从网络和存储两个方面对问题进行排查。

4. 恢复方法

在无法激活存储池的情况下，以下方面需要进行排查和考虑以找到解决方案：

- 检查存储是否发生了配置更改，导致主机无法访问对应的 LUN。
- 检查服务器到存储的链路是否可达，可通过后台命令检查 FC 和 IP SAN。
- 分析 `/var/log/ocfs2_shell_201511.log`，`/var/log/libvirt/libvirtd.log` 和 `/var/log/syslog` 日志文件，查看错误信息并尝试找到原因。
- 检查节点间共享文件的配置文件 `/etc/default/o2cb` 和 `/etc/ocfs2/cluster.conf` 是否一致。
- 检查多路径是否正常，可执行 `multipath -ll` 命令分析 LUN 是否正确建立了多路径，目录下是否有 WWID(NAA) 名称为 `/dev/disk/by-id/dm-name-36000eb34fe49de76000000000004064` 的路径。
- 如果条件允许，执行 `fdisk -l` 命令检查磁盘是否可用。
- 如果条件允许，可尝试重启对应的服务器以解决此问题。

如果通过上述步骤，仍然无法激活存储池，请及时收集日志文件，联系技术支持进行分析。

3.5.3 共享存储阻塞导致集群不可用

1. 故障现象

在管理平中，点击主机详情页中的存储池页签，页面无法正常显示，管理平台阻塞。

2. 故障影响

用户业务受较大面积影响。

3. 故障原因

故障可能性较多，需要逐个排查。

4. 恢复方法

找到共享文件系统阻塞的根节点，重启对应的主机解决。

(1) 执行 `df -h` 命令，检查存储池挂载的各个 mount point，观察能否正常访问。如果命令阻塞，长时间没有返回，则说明共享文件系统阻塞了。

(2) 执行 `mount | grep ocfs2` 命令，检查正在使用的共享文件系统。

```
root@B12-R390-FC-06:~# mount |grep ocfs2
ocfs2_dlmfs on /dlm type ocfs2_dlmfs (rw)
/dev/sdj on /vms/IPsan-St-002 type ocfs2 (rw,_netdev,heartbeat=local)
/dev/sdm on /vms/VM_BackUp type ocfs2 (rw,_netdev,heartbeat=local)
/dev/dm-0 on /vms/FC-St type ocfs2 (rw,_netdev,heartbeat=local)
/dev/sdk on /vms/IPsan-St-001 type ocfs2 (rw,_netdev,heartbeat=local)
/dev/sdl on /vms/ISO_POOL type ocfs2 (rw,_netdev,heartbeat=local)
```

(3) 执行 `debugfs.ocfs2 -R "fs_locks -B"` 命令，检查各个磁盘是否有阻塞 lock。

a. 如果输出如下内容，说明磁盘 `sdb`，`sde` 没有阻塞，而 `sdc` 则存在阻塞的 locks。sdc 有 3 个 dlm lock 处于阻塞状态。

```
root@ZJ-WZDX-313-B11-R390-IP-07:~# mount |grep ocfs2
ocfs2_dlmfs on /dlm type ocfs2_dlmfs (rw)
/dev/sdb on /vms/IPsan-St-002 type ocfs2 (rw,_netdev,heartbeat=local)
/dev/sde on /vms/VM_BackUp type ocfs2 (rw,_netdev,heartbeat=local)
/dev/sdc on /vms/IPsan-St-001 type ocfs2 (rw,_netdev,heartbeat=local)
```

```

/dev/sdd on /vms/ISO_POOL type ocfs2 (rw,_netdev,heartbeat=local)
root@ZJ-WZDX-313-B11-R390-IP-07:~# debugfs.ocfs2 -R "fs_locks -B" /dev/sdb
root@ZJ-WZDX-313-B11-R390-IP-07:~# debugfs.ocfs2 -R "fs_locks -B" /dev/sde
root@ZJ-WZDX-313-B11-R390-IP-07:~# debugfs.ocfs2 -R "fs_locks -B" /dev/sdc
Lockres: P00000000000000000000000000000000 Mode: No Lock
Flags: Initialized Attached Busy Needs Refresh
RO Holders: 0 EX Holders: 0
Pending Action: Convert Pending Unlock Action: None
Requested Mode: Exclusive Blocking Mode: No Lock
PR > Gets: 0 Fails: 0 Waits (usec) Total: 0 Max: 0
EX > Gets: 13805 Fails: 0 Waits (usec) Total: 8934764 Max: 5
Disk Refreshes: 0
Lockres: M000000000000000000000000020737820a41 Mode: No Lock
Flags: Initialized Attached Busy
RO Holders: 0 EX Holders: 0
Pending Action: Convert Pending Unlock Action: None
Requested Mode: Protected Read Blocking Mode: No Lock
PR > Gets: 2192274 Fails: 0 Waits (usec) Total: 15332879 Max: 1784
EX > Gets: 2 Fails: 0 Waits (usec) Total: 5714 Max: 3
Disk Refreshes: 1
Lockres: M000000000000000000000000020137820a41 Mode: No Lock
Flags: Initialized Attached Busy
RO Holders: 0 EX Holders: 0
Pending Action: Convert Pending Unlock Action: None
Requested Mode: Protected Read Blocking Mode: No Lock
PR > Gets: 851468 Fails: 0 Waits (usec) Total: 409746 Max: 8
EX > Gets: 3 Fails: 0 Waits (usec) Total: 6676 Max: 3
Disk Refreshes: 0

```

- b. 查询阻塞的 dlm lock 的使用信息，在 owner 节点上查询对应的使用情况。查询 dlm lock 的信息，发现其 owner 为 5，则需要到/etc/ocfs2/cluster.conf 文件中找到编号为 5 的 node，ssh 到该 node 中，再次执行 dlm 的查询命令，查看 dlm 的各个节点的使用情况。

```

root@ZJ-WZDX-313-B11-R390-IP-07:~# debugfs.ocfs2 -R "dlm_locks
P00000000000000000000000000000000" /dev/sdc
Lockres: P00000000000000000000000000000000 Owner: 5 State: 0x0
Last Used: 0 ASTs Reserved: 0 Inflight: 0 Migration Pending: No
Refs: 3 Locks: 1 On Lists: None
Reference Map:
Lock-Queue Node Level Conv Cookie Refs AST BAST Pending-Action
Converting 10 NL EX 10:1260

```

- c. 节点 5 上对于该 lock 的使用情况，可以看到节点 7 一直持有，但不释放导致了其他 node 的阻塞：

```

root@ZJ-WZDX-313-B12-R390-FC-05:~# debugfs.ocfs2 -R "dlm_locks
P00000000000000000000000000000000" /dev/sdk
Lockres: P00000000000000000000000000000000 Owner: 5 State: 0x0
Last Used: 0 ASTs Reserved: 0 Inflight: 0 Migration Pending: No
Refs: 12 Locks: 10 On Lists: None
Reference Map: 1 2 3 4 6 7 8 9 10
Lock-Queue Node Level Conv Cookie Refs AST BAST Pending-Action
Granted 7 EX -1 7:1446 2 No No None
Converting 4 NL EX 4:1443 2 No No None
Converting 2 NL EX 2:1438 2 No No None

```

```

Converting 8 NL EX 8:1442 2 No No None
Converting 6 NL EX 6:1441 2 No No None
Converting 3 NL EX 3:1438 2 No No None
Converting 1 NL EX 1:1442 2 No No None
Converting 10 NL EX 10:1260 2 No No None
Converting 5 NL EX 5:1442 2 No No None
Converting 9 NL EX 9:1259 2 No No None

```

d. 在阻塞的 **node** 上，例如上文中的 **node 7**，也可以通过如下命令查看对应的阻塞进程。

```

ps -e -o pid,stat,comm,wchan=WIDE-WCHAN-COLUMN | grep D
PID STAT COMMAND WIDE-WCHAN-COLUMN
22299 S< o2hb-C12DDF3DE7 msleep_interruptible
27904 D+ dd ocfs2_cluster_lock.isra.30

```

e. 存储阻塞则需要重启对应的服务器来解决问题，重启的步骤如下：

- 在阻塞的 **node** 上，执行 **ps -ef |grep kvm** 命令查看所有该主机上运行的虚拟机。
- 后台查询所有的虚拟机进程，并记录各个虚拟机的 **VNC** 端口等信息。
- 采用 **VNC** 或者虚拟机的远程桌面，尽量从虚拟机内部关闭虚拟机。部分虚拟机无法关闭，则需要采用 **kill** 命令强制关闭虚拟机进程，命令未 **kill -9 pid**。
- 采用 **reboot** 命令重启故障的主机。有时 **reboot** 命令会失败，此时需要通过 **iLo** 口登录到主机的管理界面，选择安全重启服务器，或者 **reset** 进行重启。
- 服务器启动后，检查系统是否恢复正常。

3.5.4 主机存储 Fence 重启

1. 故障现象

物理机在使用共享文件系统时会定期向设备写入时间戳。此外，当设备连接不通时，物理机会根据 **fence** 特性执行重启操作。这种操作是集群保护性策略的一种形式，用于规避集群资源访问完整性的风险，并将对于资源的损坏降到最低。**fence** 技术是一种保障集群高可用的解决方案，旨在确保资源访问的连续性和完整性。通过采用 **fence** 技术，可以降低数据丢失和不可用的风险，提高集群的可靠性和稳定性。

2. 故障影响

用户业务受较大面积影响。

3. 故障原因

- 服务器到存储设备的链路不稳定或反复震荡。
- 存储异常。

4. 恢复方法

一般情况下，在 **/var/log/syslog** 中会存在重启前链路断开的日志信息，如网口或 **FC** 的 **down**、**iSCSI** 或 **FC** 的链接断开、磁盘 **IO** 的异常信息。

另外在 **/var/log/ocfs2_fence_restart.log** 中也能看到重启的记录，具体由哪个 **LUN** 产生的，以及重启的原因：

```

Restarted at 2017-03-14 14:28:52 (1489501732.600511). UNKOWN_DEV, UNKOWN_UUID,
o2quo_make_decision: TCP disconnected with other nodes.
Restarted at 2017-04-08 22:04:57 (1491689097.184531). dm-1,
BE229C2491E64FEA98550DF0469B6C10, o2quo_disk_timeout: Disk heartbeat timeout.

```

重启问题需要排查中间链路，存储的访问是否正常。常见的排查策略，可从几个方面进行排查链路问题：

- 服务器，刀框的 **bios/firmware** 版本是否需要升级。
- **VC** 刀框的配置是否合适。
- 服务器上 **IML** 硬件日志是否有硬件故障。

- 服务器/var/log/mcelog 中是否报硬件异常。
- 连接 FC 交换机的 FC 光纤是否正常，可以通过交换机上查看对应接口光功率，crc 错误等进行查看。
- 如果使用 FC SAN，需要检查 FC 交换机上的配置是否一对一的端口 zone，隔离不同 HBA 之间的影响。如果 FC 交换机做了堆叠，由于主机上对于 LUN 启用的多路径配置，建议取消 FC 交换机堆叠。对于某些型号的 FC 交换机，则需要考虑交换机的版本升级，请与交换机相关的技术支持进行确认。
- 如果使用 IP SAN 存储，需要查看主机对应物理网口的丢包情况，中间网线和交换机上是否有异常日志等信息。
- 在存储中检查是否有异常日志等信息，或者误操作等情况。

3.5.5 共享文件系统的强制修复

1. 故障现象

当服务器突然断电或出现其他异常情况时，共享文件系统存储在磁盘上的数据可能存在不一致性。这种问题可能会导致后续的共享文件系统使用异常。特别是在进行磁盘 dd 和 fio 等操作时，可能会导致磁盘损坏，需要进行修复，否则可能会在运行过程中发生数据丢失和服务器挂起等问题，从而导致业务中断。

2. 故障影响

用户业务中断，有数据丢失的风险。

3. 故障原因

服务器突然断电等异常。

4. 恢复方法

- (1) 需要在所有主机上暂停对应的共享文件系统。
- (2) 以其中一台主机为例，采用命令进行预先检查，例如对于 dm-0 磁盘，执行 fsck.ocfs2 -fn /dev/dm-0 命令，并保存命令的输出。
- (3) 执行修复命令（例如 fsck.ocfs2 -fpy /dev/dm-0）进行修复。
- (4) 待修复结束后，在所有的主机上激活存储池。
- (5) 在所有的主机上启动对应的虚拟机业务。



注意

- 出现需要修复处理，推荐优先联系技术支持人员远程处理。
 - 需要记录操作的中间过程的信息输出，连同收集到的日志，一并发送给技术支持人员用于分析。
 - 如果损坏严重，修复后还有可能无法使用，需要具体问题具体分析，请及时联系研发。
-

3.5.6 启动共享存储池时失败，提示“存储池没有可用的 slot”

1. 故障现象

在管理平台中启动共享存储池时提示“存储池没有可用的 slot”。

2. 故障影响

无法启动存储池，影响客户业务上线。

3. 故障原因

查看后台日志，确认挂载时是否有如下报错：

```
Error: Disk /dev/disk/by-id/dm-name-2225d000155a30dda had been mounted by other hosts, not need to format it.
```

如有，则是由于该存储卷已经挂给了其它集群的主机使用，系统后台检测到后该情况后，没有继续进行挂载。

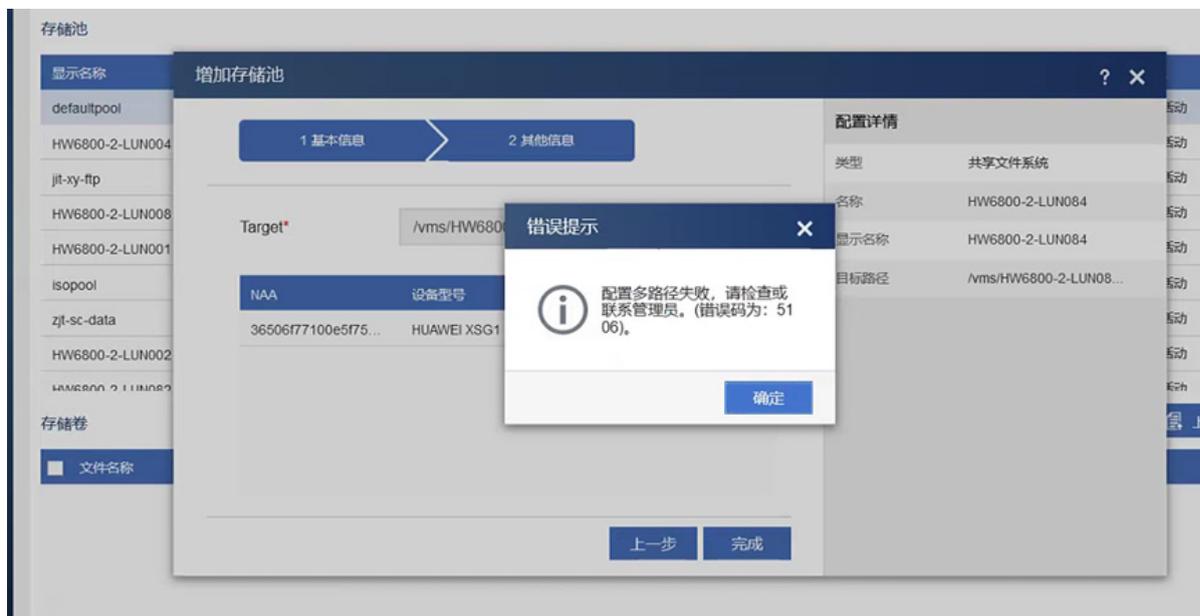
4. 恢复方法

将存储从另一平台中解除挂载，解除后，本平台即可正常挂载。

3.5.7 界面添加共享存储时报配置多路径失败

1. 故障现象

界面增加存储池时，选择所需添加的存储池时报错：“配置多路径失败，请检查或联系管理员。（错误代码：5106）”：



2. 故障影响

无法增加存储池，影响客户业务上线。

3. 故障原因

查看 **syslog** 日志，确认 **Usphere** 识别的共享存储卷的 **WWID** 号与存储端设置的是否一致。如果不一致，说明存储侧在将所划的存储卷映射到 **Usphere** 之后，存储端又对该存储卷进行了修改，使得 **WWID** 号发生了改变。如图所示：**Usphere** 识别到的 **wwid** 为~698，存储端的 **wwid** 为~699。


```

root@cvknode11:~# ovs-appctl bond/show
---- vswitch0_bond ----
bond_mode: active-backup
bond_may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lACP_status: off
lACP_fallback_ab: true
active slave mac: 40:a8:f0:28:14:d8(eth0)

slave eth0: enabled
           active slave
           may_enable: true

slave eth1: enabled
           may_enable: true

```

聚合负责分担模式
active-backup: 主备
balance-slb: 基本
balance-tcp: 高级

lACP状态:
off: 静态
configured: 动态协商失败
negotiated: 动态协商成功

主链路

聚合链路状态

如上图所示，执行 `ovs-appctl bond/show` 命令可以查看虚拟交换机的聚合配置：

- 如果虚拟交换机使用的是静态主备聚合，则物理交换机不需要配置聚合组。
- 如果虚拟交换机使用的是静态负载分担聚合，要求物理交换机也配置静态聚合组，如交换机和服务器的聚合配置不一致，则会出现网络转发故障。
- 如果虚拟交换机使用动态聚合，则需要注意查看 `lACP_status` 状态。如果其状态是 `configured`，则表示动态聚合协商失败，此时需要排查物理交换机聚合组的配置情况，包括聚合组类型、聚合组端口成员的正确性、以及端口是否处于 `select` 状态等情况。
- 检查虚拟交换机的物理网卡状态是否正确。从界面查询到虚拟交换机使用的物理网卡后，可以执行 `ifconfig ethx` 命令查看网卡的状态，例如：

```

[root@cvknode110 casloal# ifconfig eth6
eth6: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
       inet6 fe80::62da:83ff:fe3d:a5e prefixlen 64 scopeid 0x20<link>
       inet6 fe80::62da:83ff:fe3d:a5e prefixlen 64 scopeid 0x20<link>
       ether 60:da:83:3d:0a:5e txqueuelen 1000 (Ethernet)
       RX packets 54153826 bytes 106867962564 (99.5 GiB)
       RX errors 0 dropped 7 overruns 83979 frame 0
       TX packets 8404047 bytes 1528917305 (1.4 GiB)
       TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
       device memory 0xc7460000-c747ffff

```

网卡状态说明如下：

- `<UP,BROADCAST,RUNNING,MULTICAST>`：表示网卡正常 UP，可以正常收发包；
- `<UP,BROADCAST,MULTICAST>`：表示网卡软件层面 UP，但是链路层 DOWN。此时可以使用 `ip link show dev ethx` 查看，如果网卡状态中有 `NO-CARRIER` 则表示网卡物理链路 down，此时应该检查网卡插线是否有问题，如果是光卡，则可以尝试更换光纤或者光模块。
- `<BROADCAST,MULTICAST>`：表示网卡软件层面 DOWN，可以尝试执行 `ifup ethx` 或 `ip link set dev ethx up` 命令，在软件层面将网卡 UP。
- 检查网卡 MTU 是否正确，如 MTU 设置有误，可以在虚拟交换机界面进行修改。
- 网卡收发包是否存在丢包错包：如果存在丢包错包，则执行 `ifconfig ethx; sleep 10; ifconfig ethx` 命令查看丢包错包计数是否有增加。如果没有增加，说明计数是历史统计数据，可以忽略，如果有增加，则需要进行下一步的排查。

```

[root@cvknode110 test]# ip link show dev eth2
10: eth2: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc mq master ovs-system state DOWN mode DEFAULT
   link/ether d4:61:fe:17:c9:aa brd ff:ff:ff:ff:ff:ff

```

- **dropped**: 表示数据包已经进入 Ring Buffer，但是由于内存不够等系统原因，导致在拷贝到内存的过程中被丢弃。可以通过 `ethtool -G ethx rx xxx` 的方式增大网卡 Ring Buffer 的方式规避。注：修改网卡 Ring Buffer 会导致网卡短时中断，建议无业务时修改。
- **overruns**: Ring Buffer 满，导致报文还没有到网卡的 Ring Buffer 时就被物理网卡丢弃，一般是 CPU 无法及时处理网卡中断引起。遇到此种情况，可以修改网卡中断的亲中性，将网卡中断放到空闲的 CPU 上处理。
- **frame**: 接收到的 CRC 校验错误、且长度不是整字节数的报文的数量，这种情况建议更换网线/光纤和光模块。
- **errors**: 所有收包错误数量的总和，报错：Ring Buffer 溢出，too-long-frames 错误，crc 错误，overruns 等。
- 检查虚拟网卡网络配置是否正确。在 VKS 主机后台执行 `/opt/bin/ovs_dbg_listports` 命令，获取 VKS 上当前网络拓扑信息，此网络拓扑信息在后续的问题排查中有很重要的作用。

虚拟交换机名称	vnet名称	虚拟网卡mac	虚拟机内部ip	用户态端口号	内核态端口号	vlan	MTU	vnet所属虚拟机名称	虚拟机vnc端口号
vswitch0	(vnb)	40a8f02814d8	172.16.204.111	65534/2			1500		
int	vswitch0	0cda411d0e15	172.16.130.96	14/5			1500	win7 5900	
	vnet0	0cda411d4379	172.16.51.23	12/7		100x	2800	E0513 5902	
	vnet1	0cda411d102e	172.16.51.70	11/6			1500	E0530 5901	
bond	vswitch0_bond	eth1 40a8f02814d9		31/3	uxe		1500	03:00.1 21	
	eth0 40a8f02814d8			32/4	*uxe		1500	03:00.0 21	

聚合端口

- 检查 VLAN 配置是否正确：在 Usphere 虚拟化网络拓扑图中，根据虚拟机的 MAC 地址（可在管理平台界面中查询），找到对应的 vnetx 设备后，可以查看虚拟网卡的 VLAN 配置，请核对是否和现网的业务要求相符。
- 检查 IP/MAC 绑定是否正确：在管理平台的修改虚拟机页面，找到对应的虚拟网卡，可以查看虚拟网卡是否配置了 IP/MAC 绑定：



IP/MAC 绑定表示虚拟网卡绑定此 IP 后，此网卡只能使用绑定的 IP 发送数据报文，若使用其他的 IP 发送报文，则报文会被 OVS 流表拦截。

所以虚拟机内部有多个 IP 的场景，建议将 IP/MAC 绑定配置取消。同时也需要排查下绑定的 IP 与虚拟机内部的 IP 是否一致。

- 检查 ACL/虚拟防火墙是否正确：ACL/虚拟防火墙配置不正确时，可能导致虚拟机发送的报文被 OVS 流表拦截。尤其是 Cloud OS 场景，需要排查与虚拟机不通的 IP 地址是否被安全组放行（OS 默认白名单机制）。
- 检查是否开启虚拟机防病毒：防病毒异常场景，可能会导致虚拟机报文被防病毒拦截导致网络不通，或者虚拟机流量被防病毒限制导致带宽限速问题。遇到此情况，可以尝试关闭防病毒，查看问题是否恢复。可执行 `service ds_am stop; service ds_agent stop; service ds_filter stop` 命令，关闭亚信防病毒。
- 抓包确认具体丢包位置：如果以上的所有方法均无法确认问题，可以通过抓包的方法，确认具体的丢包位置，锁定问题范围。

“check_network”脚本是在 Usphere 虚拟化场景的通用抓包脚本，使用方法以及参数说明如下。

check_network.sh

`[--vswitch=xxx]`//指定需要抓包的 vswitch，脚本会默认从 vswitch，vswitch 绑定的物理口抓包。

[--iface="vnet0,vnet1,vnet2,..."]//指定需要抓包的 vnet 设备。
 [--iface_mac="0c:da:41:1d:f6:4b,0c:da:41:1d:1d:7d,..."]//可以通过 mac 指定需要抓包的虚拟机网卡。
 [--ping=x.x.x.x]//同步在执行脚本的设备上发起 ping 任务, ping 参数只支持指定一个 ip。
 [xxxxxxxxxxxx]//指定抓包的过滤条件,防止 pcap 文件占用太大的空间,支持的条件请参考下方的 example,上述参数均为可选参数,可以根据实际场景选择需要的参数。

dump example:

```
ARP: arp and host x.x.x.x
ICMP: icmp and host x.x.x.x
ICMP6: icmp6 and host x:x::x:x
DNS: port domain
DHCP: udp and port 67 and port 68
SSH: port 22 and host x.x.x.x
LLDP: ether proto 0x88cc
LACP: ether proto 0x8809
```

脚本使用方法举例:

- o 问题 1: 偶现 vswitch0 故障告警。抓包方法如下所示。
 - 在出现问题的 VKS 中执行/home/test/check_network.sh --vswitch=vswitch0 --ping=172.23.51.112 "icmp and host 172.23.51.112"命令。
 --vswitch= vswitch0, 指定使用虚拟交换机 vswitch0 来抓包。
 --ping=172.23.51.112, 表示脚本自动发起 ping 任务, ping 的目的端为 172.23.51.112。实际使用时, 请指定成 VMS 的 IP 地址。
 "icmp and host 172.23.51.112", 表示抓包的过滤条件, 请将“172.23.51.112”替换为实际 VMS 的 IP 地址。
 - 在 VMS 中执行/home/test/check_network.sh --vswitch=vswitch0 "icmp and host 172.23.51.114"命令。
 无需--ping 参数, 因为在 VKS 中已经发起 ping 任务, 所以不需要在 VMS 上重新发起 ping 任务。
 --vswitch= vswitch0, 同上。
 "icmp and host 172.23.51.112", 表示抓包的过滤条件, 请将 IP 地址替换为实际 VKS 主机的 IP 地址。
- o 问题 2: 虚拟机凌晨虚拟机 I/O 大量飙升问题。该问题无明显指向, 需排查 I/O 飙升时, 虚拟机主要存在什么业务, 所以需要抓取虚拟机的报文。
 - 在虚拟机所在的 VKS 主机中执行/home/check_network/check_network.sh --iface_mac="0c:da:41:1d:f6:4b,0c:da:41:1d:1d:7d"命令。
 --iface_mac 表示需要抓包的异常虚拟机的网卡 MAC 地址, 可以指定多个 MAC 地址, 中间用“,” 隔开。虚拟机网卡的 MAC 地址可以在管理平台的虚拟机概要界面中查询。



- 脚本产生文件说明。

在/root/tcpdump/目录下, 会生成相关的日志以及抓包文件。

-rw-r----- 1 root root 37 Jun 7 10:37 20230607103752-check_network.pid, 记录 check_network 脚本运行生成的 tcpdump 以及 ping 任务的进程 pid 号。

```

-rw-r----- 1 root root 3313 Jun  7 10:37 20230607103752-ovs_dbg_listports, 记录
check_network 脚本运行时的网络拓扑信息。
-rw-r----- 1 root root 1412 Jun  7 10:38 20230607103752-ping.log, ping 任务日志。
-rw-r----- 1 root root  287 Jun  7 10:37 check_network.log, 记录 check_network 脚本
运行的日志。
drwxr-x--- 2 root root 4096 Jun  7 10:37 eth2, vswitch0 绑定的物理网卡的抓包文件
drwxr-x--- 2 root root 4096 Jun  7 10:37 vnet0, 指定的虚拟网卡的抓包文件
drwxr-x--- 2 root root 4096 Jun  7 10:37 vnet1, drwxr-x--- 2 root root 4096 Jun  7
10:37 vnet3, drwxr-x--- 2 root root 4096 Jun  7 10:37 vswitch0, vswitch0 的抓包文
件。

```

o 结束脚本任务。

查看 check_network 运行记录的 pid 文件，需要注意文件的时间戳信息。

```
cat /root/tcpdump/20230518100318-check_network.pid
```

```
2403 2405 2407 2408
```

执行 kill 命令，关闭上述 pid 文件中记录的 pid，如 kill 2403、kill 2405、kill 2407、kill 2408。



注意

check_network.sh 可多次运行，多次运行会产生多个 pid 文件，抓包结束后都需要手动 kill pid 文件中记录的 pid。

抓包结束后，请一定即使结束抓包任务，避免持续抓包占用系统空间。

抓包结束后，请将 /root/tcpdump 目录打包发送给研发分析。

3.6.2 虚拟机 IP 地址冲突

1. 故障现象

主机（以 172.16.51.23 为例）处于连接异常状态，查看主机的服务都是正常的；查看日志有连接该主机失败日志。SSH 登录主机输入正确的密码，SSH 无法连接并报密码不正确的错误。

2. 故障影响

无法正常链接主机。

3. 故障原因

在本地 PC 的网卡上开启抓包，同时在主机上的 vswitch0 上开启抓包，之后在 PC 上发起 SSH 请求，分析抓到的报文，示例如下。

- 本地 PC 上的请求报文如下，从报文中可以看到 172.16.51.23 对应的 MAC 地址是 0c:da:41:1d:10:2e。

No.	Time	Source	Destination	Protocol	Length	Info
9	0.011372	172.16.51.23	172.16.204.111	SSHv2	1442	Server: Key Exchange Init
11	0.018108	172.16.204.111	172.16.51.23	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
12	0.022376	172.16.51.23	172.16.204.111	SSHv2	346	Server: Diffie-Hellman Key Exchange Reply, New Keys
13	0.027075	172.16.204.111	172.16.51.23	SSHv2	82	Client: New Keys

Frame 11: 114 bytes on wire (912 bits), 114 bytes captured (912 bits)
 Ethernet II, Src: HewlettP_28:14:d8 (40:a8:f0:28:14:d8), Dst: Hangzhou_1d:10:2e (0c:da:41:1d:10:2e)
 Internet Protocol Version 4, Src: 172.16.204.111, Dst: 172.16.51.23
 Transmission Control Protocol, Src Port: 51682 (51682), Dst Port: 22 (22), Seq: 1717, Ack: 1421, Len: 48
 SSH Protocol

```
0000  0c da 41 1d 10 2e 40 a8 f0 28 14 d8 08 00 45 00  ..A...@. ....E.
```

- 主机中未抓到请求报文，查看 VKS 中 vswitch0 的 MAC 地址是 0c:da:41:1d:43:79，IP 地址也是 172.16.51.23。

```
root@cvknode5123:~# ip addr show dev vswitch0
5: vswitch0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN group default
    link/ether 0c:da:41:1d:43:79 brd ff:ff:ff:ff:ff:ff
    inet 172.16.51.23/16 brd 172.16.255.255 scope global vswitch0
        valid_lft forever preferred_lft forever
    inet6 fe80::eda:41ff:fe1d:4379/64 scope link
        valid_lft forever preferred_lft forever
```

- 上述分析发现局域网内部有两个设备都配置了 IP 地址 172.16.51.23，导致 IP 地址冲突。

4. 恢复方法

将存在 IP 冲突的地址重新配置为其他 IP 地址。

3.6.3 主备聚合丢包问题

1. 问题现象

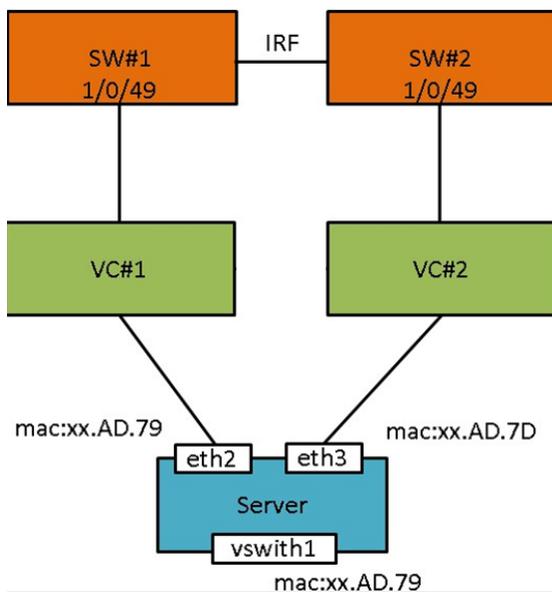
某局点现场发现从服务器（以 192.168.12.19 为例）ping 物理交换机（以 192.168.12.254 为例）可以正常 ping 通，但是从物理交换机（192.168.12.254）ping 服务器（192.168.12.19）出现时通时不通的现象。

2. 故障影响

用户网络出现丢包，影响虚拟化环境的稳定性。

3. 故障原因

组网示例如下：



在本例中，现场使用了 UIS 刀片服务器，并且刀箱内部使用了 VC 设备作为网络互联模块，刀片服务器的 eth2 和 eth3 分别连接 VC#1 和 VC#2，VC#1 和 VC#2 分别连接交换机 SW#1 和 SW#2，SW#1 和 SW#2 做 IRF 堆叠。

刀片服务器安装 Usphere 系统，虚拟交换机 vswitch1 绑定 eth2 和 eth3，并且配置静态主备负载分担聚合。eth2 和 eth3 都开启了 LLDP 功能。

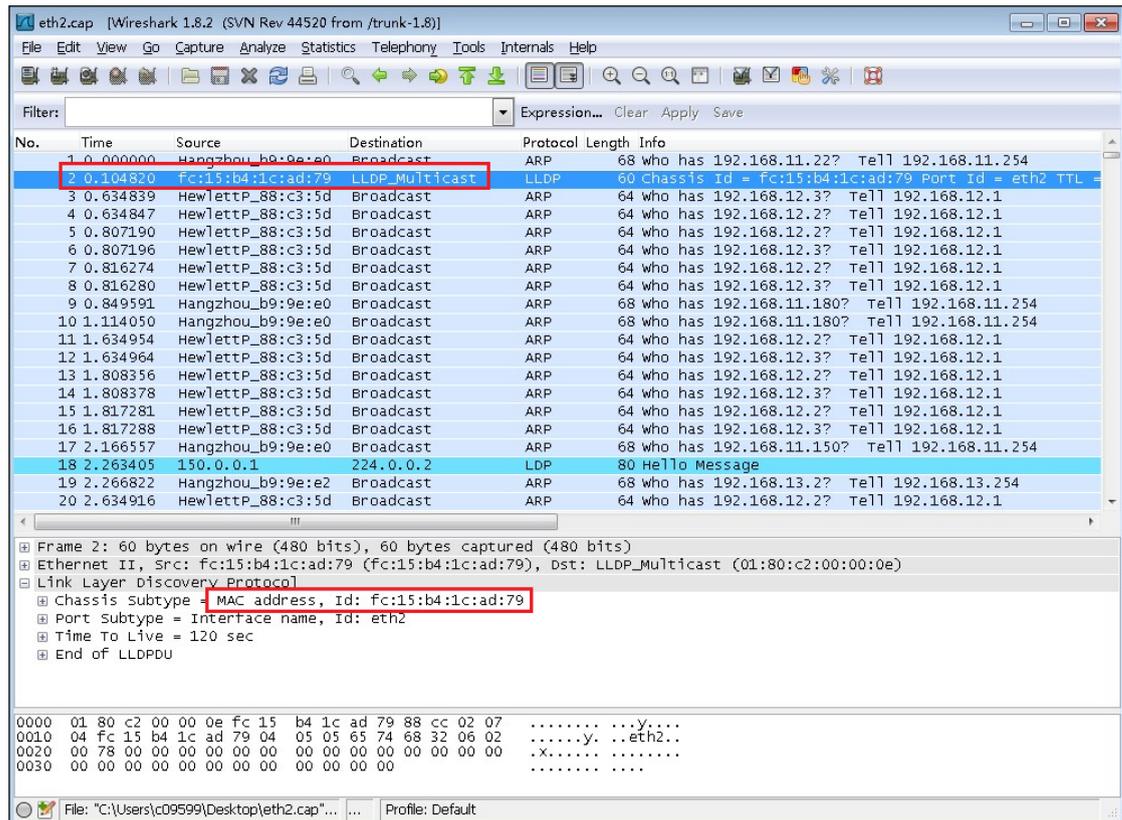
虚拟交换机 vswitch1 的 IP 地址为 192.168.12.19，网关为 192.168.12.254，该网关配置在物理交换机上。

- (1) 发现从服务器（192.168.12.19）ping 交换机（192.168.12.254）可以正常 ping 通，但是从交换机（192.168.12.254）ping 服务器（192.168.12.19）出现时通时不通的现象。
- (2) 进一步测试，发现当 eth2 网卡为主网卡时，VKS 主机（192.168.12.19）与网关（192.168.12.254）互相均可以通信；主网卡为 eth3 时，网关 ping 主机时会出现丢包。因此重点分析 eth3 作为主链路时的报文转发情况。

- VKS 主机（192.168.12.19）到网关（192.168.12.254）的报文的流程为：“vswitch1-eth3-VC#2-SW#2”。
 - vswitch1 的 MAC 地址“FC:15:B4:1C:AD:79”会记录在 VC#2 的下行口和 SW#2 的下行口。
 - 网关（192.168.12.254）到 VKS 主机（192.168.12.19）的报文会根据 SW#2 和 VC#2 的 MAC 地址表项转发，流程为：“SW#2-VC#2-eth3-vswitch1”。
- (3) 使用 tcpdump 命令对网卡 eth2 进行抓包，并保存报文。

```
root@HZ-CAS02-CVK19:~# tcpdump -i eth2 -s 0 -w /tmp/eth2.cap
tcpdump: listening on eth2, link-type EN10MB (Ethernet), capture size 262144 bytes
^C53438 packets captured
53438 packets received by filter
0 packets dropped by kernel
root@HZ-CAS02-CVK19:~#
```

- (4) 将报文保存到本地 PC，使用 Wireshark 工具分析报文，可以看到虽然网卡 eth2 状态为备链路，但仍在不停的发送 LLDP_multicast 报文（网卡 eth2 上开启了 LLDP），报文信息如下所示：



LLDP_multicast 报文的 MAC 地址为：FC:15:B4:1C:AD:79。

- (5) eth3 为主链路时，虚拟交换机 vswitch1（mac: FC:15:B4:1C:AD:79）的 MAC 应该学习到 SW#2 和 VC#2，但是网卡 eth2(MAC 也是 FC:15:B4:1C:AD:79)会定时发送 LLDP_multicast，从而影响到 VC#1 上的 MAC 地址表项信息，导致 MAC FC:15:B4:1C:AD:79 漂移到了 VC#1 和 SW#1，网关 ping 主机的报文会按照 SW#1-VC#1-eth2 的路径发送，此时 eth2 的状态为备用状态，不处理报文，OVS 直接丢弃报文，导致 ping 不通。

4. 恢复方法

关闭物理网卡的 LLDP 功能即可恢复。

3.6.4 配置 ACL 导致虚拟机无法访问网关地址

1. 故障现象

虚拟机可以正常访问其他地址，但是无法访问网关（以 172.16.51.6 为例）。

2. 故障影响

用户虚拟机业务受影响。

3. 故障原因

报文转发路径为：虚拟机网卡-**vnet**-主机网卡-物理交换机。在 **vnet** 和主机网卡 **eth0** 上抓包发现如下现象。

```
root@cvknode112:~# tcpdump -i vnet0 -ne icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on vnet0, link-type EN10MB (Ethernet), capture size 262144 bytes
10:57:31.449569 0c:da:41:1d:56:a7 > a0:36:9f:97:76:b8, ethertype IPv4 (0x8000), length 98: 172.16.51.20 > 172.16.51.6: ICMP echo request, id 4244, seq 83, length 64
10:57:32.473592 0c:da:41:1d:56:a7 > a0:36:9f:97:76:b8, ethertype IPv4 (0x8000), length 98: 172.16.51.20 > 172.16.51.6: ICMP echo request, id 4244, seq 84, length 64
^C
2 packets captured
4 packets received by filter
0 packets dropped by kernel
root@cvknode112:~#
root@cvknode112:~# tcpdump -i eth0 -ne icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C
0 packets captured
0 packets received by filter
0 packets dropped by kernel
2 packets dropped by interface
root@cvknode112:~#
```

vnet 上已经收到了 **icmp** 请求报文，但是 **eth** 上没有抓到。

通过如下方法，可以查看 **vnet** 上的 **ACL** 配置：

- (1) 执行 `ovs_dbg_listports` 命令，可以查看 **vnet** 所属虚拟机。
- (2) 执行 `virsh dumpxml vm_name` 命令，可以查看虚拟机的 XML 文件，可以找到 **interface** 节点，可以查看到如下信息：

```
<interface type='bridge'>
  <mac address='0c:da:41:1d:56:a7' />
  <source bridge='vswitch0' />
  <vlan>
    <tag id='1' />
  </vlan>
  <virtualport type='openvswitch'>
    <parameters interfaceid='bfcd8333-2282-4d08-a0bb-cc14a37ff1e9' />
  </virtualport>
  <target dev='vnet0' />
  <model type='virtio' />
  <driver name='vhost' />
  <hotpluggable state='on' />
  <filterref filter='acl-test' /> //acl 的名称
  <priority type='low' />
  <mtu size='1500' />
  <alias name='net0' />
  <address type='pci' domain='0x0000' bus='0x00' slot='0x03' function='0x0' />
</interface>
```

- (3) 执行 `virsh nwfilter-dumpxml acl_name` 命令，可以查看 **acl** 的具体内容。

```
root@cvknode112:~# virsh nwfilter-dumpxml acl-test
<filter name='acl-test' chain='root'> ACL 名称 acl-test
  <uuid>41f62fca-275b-4a72-8ad8-515d9e9589c1</uuid>
  <rule action='accept' direction='in' priority='4' statematch='false'> //入方向默认放行
  <all/>
</rule>
```

```

<rule action='accept' direction='out' priority='5' statematch='false'>//出方向默认放行
行
<all/>
</rule>
<rule action='accept' direction='in' priority='6' statematch='false'>//入方向默认放行
行
<all-ipv6/>
</rule>
<rule action='accept' direction='out' priority='7' statematch='false'>//出方向默认放行
行
<all-ipv6/>
</rule>
<rule action='drop' direction='in' priority='10' statematch='false'>
<icmp dstipaddr='172.16.51.6' dstipmask='32' />//丢弃目的地址为 172.16.51.6 的 icmp 报
文
</rule>
</filter>

```

(4) 分析 ACL 规则，可以知道是，OVS 的 ACL 规则将报文丢弃。

```

root@cvknode112:~# ovs-appctl dpif/dump-flows vswitch0 | grep "172.16.51.6"
recirc_id(0),in_port(8),eth_type(0x0800),ipv4(dst=172.16.51.6,proto=1,tos=0/0xfc,frag=no),
packets:10426,bytes:1021748,used:0.369s,actions:drop

```

(5) 查看 ovs 的内核流表项，也可以看到报文被 drop 的信息。

4. 恢复方法

修改虚拟机的 ACL 配置。

3.6.5 虚拟防火墙导致 SSH 连接无法建立

1. 故障现象

两个虚拟机都配置了虚拟防火墙，它的策略中不存在阻止 SSH 的规则，并且虚拟机配置了相同的 VLAN，但虚拟机之间不能通过 SSH 互访。

2. 故障影响

虚拟机之间不能 SSH 互访，影响用户网络业务。

3. 故障原因

如果虚拟机网卡都绑定在 vs-app 上，虚拟机都配置相同的 VLAN 为 188 及虚拟机防火墙 FW_white，如下图所示。

```

[root@cvknode240 ~]# ovs dbg listports

```

PCI	NAME	LW	IO	NU	SRIO	MTU	A/L	SPD	MAC	IP	VEN:DEV	DESC	SUBSYS	DRIVE	FIRMWARE
0000:3d:00.0	eth0	66	0	0/32	1500	u/u	1000	34:6b:5b:29:88:63	0x8086:0x37d1	X722	0000	140e	3.33	0x80000f0c	1.1767.0
0000:3d:00.1	eth1	67	0	0/32	1500	u/u	1000	34:6b:5b:29:88:64	0x8086:0x37d1	X722	0000	140e	3.33	0x80000f0c	1.1767.0
0000:3d:00.2	eth2	68	0	0/32	1500	u/u	1000	34:6b:5b:29:88:65	0x8086:0x37d1	X722	0000	140e	3.33	0x80000f0c	1.1767.0
0000:3d:00.3	eth3	69	0	0/32	1500	u/u	1000	34:6b:5b:29:88:66	0x8086:0x37d1	X722	0000	140e	3.33	0x80000f0c	1.1767.0
0000:5f:00.0	eth4	84	0	0/63	1500	u/u	10000	74:85:c4:1d:b3:98	0x8086:0x10fb	82599	ixgbe	0x800006db			
0000:5f:00.1	pci_sf_00_1	85	0	0/63	1500	u/u	10000	74:85:c4:1d:b3:9a	0x8086:0x10fb	82599	vfiio-pci				

Type	Name	MAC	VMMAC	IP	OFPort	Vlanx	MTU	VMName	VNC Filter
vswitch0 (vsw)									
int	vswitch0	346b5b298863		10.125.32.240	65534/2		1500		
	vnet3	feda411d590e	Ocda411d590e		2/17		1500	E0730_237	5902
int	dhcp0	5eeca7ca416			20/3	44x			
	vnet0	feda411d9d9b	Ocda411d9d9b		1/14		1500	ubu14	5900
	vnet6	feda411d34f0	Ocda411d34f0		4/20		1500	kylin_v10_SP_226	5904
	vnet9	feda411d9011	Ocda411d9011		8/23		1500	kylin10_219	5908
	eth0	346b5b298863			7/1		1500	3d:00.0	66
	vnet7	feda411d1ded	Ocda411d1ded		5/21		1500	openSUSE15_218	5906
int	dhcp1	3acd4d78e10e			21/4	66x			
	vnet5	feda411d37a5	Ocda411d37a5		3/19		1500	kylin_v10_SP_226	5904
	vnet8	feda411de7c7	Ocda411de7c7		6/22		1800	openSUSE15_218	5906
vs-app (vsw)									
int	vs-app	7485c41db398			65534/11		1500		
	vnet11	feda411d8513	Ocda411d8513		3/25	188x	1500	kylin10_219	5908 FW_white
	vnet10	feda411d13c6	Ocda411d13c6		2/24	188x	1500	kylin_v10_SP_226	5904 FW_white
	eth4	7485c41db398			1/10		1500	5f:00.0	84

其中两个虚拟机的用户态流表如下：

```
[root@cvknode240 ~]# ovs-ofctl -O OpenFlow13 dump-flows vs-app | grep "in_port=3|reg1=0x3"
cookie=0x30, duration=238.433s, table=0, n_packets=18, n_bytes=6210, priority=10, ct_state=trk,ip,in_port=3 actions=ct(table=5,zone=188)
cookie=0x30, duration=238.433s, table=0, n_packets=80, n_bytes=12948, priority=10, ct_state=trk,ipv6,in_port=3 actions=ct(table=5,zone=188)
cookie=0x30, duration=238.433s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=est-rel-rpl,ip,in_port=3 actions=ct(commit,zone=188)
cookie=0x30, duration=238.433s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=est-rel-rpl,ipv6,in_port=3 actions=ct(commit,zone=188)
cookie=0x30, duration=238.433s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=new-est,ip,in_port=3 actions=drop
cookie=0x30, duration=238.433s, table=10, n_packets=45, n_bytes=10026, priority=30004, ct_state=new-est,ipv6,in_port=3 actions=drop
cookie=0x30, duration=238.433s, table=10, n_packets=0, n_bytes=0, priority=70, ct_state=inv+trk,ip,in_port=3 actions=drop

[root@cvknode240 ~]# ovs-ofctl -O OpenFlow13 dump-flows vs-app | grep "in_port=2|reg1=0x2"
cookie=0x30, duration=7.615s, table=0, n_packets=0, n_bytes=0, priority=10, ct_state=trk,ip,in_port=2 actions=ct(table=5)
cookie=0x30, duration=7.615s, table=0, n_packets=0, n_bytes=0, priority=10, ct_state=trk,ipv6,in_port=2 actions=ct(table=5)
cookie=0x30, duration=7.615s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=est-rel-rpl,ip,in_port=2 actions=ct(commit)
cookie=0x30, duration=7.615s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=est-rel-rpl,ipv6,in_port=2 actions=ct(commit)
cookie=0x30, duration=7.615s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=new-est,ip,in_port=2 actions=drop
cookie=0x30, duration=7.615s, table=10, n_packets=0, n_bytes=0, priority=30004, ct_state=new-est,ipv6,in_port=2 actions=drop
cookie=0x30, duration=7.615s, table=10, n_packets=0, n_bytes=0, priority=70, ct_state=inv+trk,ip,in_port=2 actions=drop
```

通过对比发现，流表中 zone 的值不一样，一个为 188，一个为 0，正常情况下都应该是 188，zone 值不同导致 SSH 失败。

4. 恢复方法

需要通过 set_cvm_port_flow.sh 以及 set_cvk_port_flow.sh 脚本进行修复。

- 在 VMS 中使用 set_cvm_port_flow.sh 命令执行脚本。具体操作方法如下：
 - 上传脚本 set_cvm_port_flow.sh、set_cvk_port_flow.sh 到 VMS 主机的同一目录下
 - 执行 bash set_cvm_port_flow.sh 脚本，会给所有 vks 上传脚本 set_cvk_port_flow.sh 到 /opt/bin/目录下。
 - 执行完成后，在/var/log/set_cvk_port_flow.log 查看日志，根据日志可以看出是否存在流表 error 被修复的情况。
- 在 VKS 中执行 set_cvk_port_flow.sh 命令，可以针对某个具体的虚拟机端口进行 flow 的修复，也可以不加参数对所有虚拟机端口进行 flow 的修复。具体操作方法如下：

```
bash /opt/bin/set_cvk_port_flow.sh vnetx (参数不对时，会有提示)
bash /opt/bin/set_cvk_port_flow.sh aa:bb:cc:dd:ee:ff(参数不对时，会有提示)
bash /opt/bin/set_cvk_port_flow.sh
```

- 备注：
 - set_vms_port_flow.sh 脚本只能在 VMS 上执行，脚本无参数，说明如下：
 - script can only run on VMS!
 - example:
 - set_vms_port_flow.sh check whether the virtual port configuration vfirewall and flow table are correct on cvm and cvk.
 - 所有 vks 上都有定时任务，每天 23:00，会自动执行以下脚本 set_cvk_port_flow.sh，如果出现虚拟机 flow 不对的情况，会自动修复。
 - vks 日志说明如下：

```
-----2022/08/29
11:21:50-----
cvknode116 vnet0 of vswitch0 is config FW, the flow table is correct, do not need
revalidate flow
(配置状态防火墙，流表正确，不需要重刷流表)
cvknode116 vnet1 of vswitch0 is not config FW, do not need revalidate flow
cvknode116 vnet2 of vswitch0 is config FW, the flow table is error, do need revalidate
flow
(配置状态防火墙，流表错误，需要重刷流表)
cvknode116 vnet3 of vswitch0 is config FW, the flow table is correct, do not need
revalidate flow
cvknode116 vnet4 of vswitch0 is not config FW, do not need revalidate flow
(没有配置状态防火墙，不需要重刷流表)
cvknode116 vnet5 of vswitch0 is not config FW, do not need revalidate flow
cvknode116 vnet6 of vswitch0 is not config FW, do not need revalidate flow
-----2022/08/29
11:21:50-----
```

3.6.6 主机中毒对外部发起攻击

1. 故障现象

主机中存在病毒，并且该病毒尝试向外部建立大量的非连接，这可能会导致网络资源的大量消耗，并最终导致网络故障。

2. 故障影响

用户业务网络受影响。

3. 故障原因

使用 `tcpdump` 进行抓包，判断环境是否存在异常的流量，异常流量可能具有以下特点：

- 发起 DNS 解析请求。
- 发起 TCP SYN 请求，但一般都不会建立连接。
- 目的 IP 不是用户自己环境里的，属于一些数据中心。
- 量大。

执行 `netstat -atunp` 命令查看是否有异常的进程在尝试建立网络连接。异常的进程可能具有以下特点。

- 连接的外部地址 IP，非用户环境的 IP。
- 进程名伪装成一些常用程序名：如“cd”，“ls”，“echo”，“sh”等。
- 如果 VKS 与外网不通，连接状态可能一直处在 SYN_SENT。

```
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 172.16.0.2:22          122.225.207.198:31245  ESTABLISHED 52034/5
tcp        0      0 127.0.0.1:3306         127.0.0.1:46207       ESTABLISHED 2102/mysqld
tcp        0      0 172.16.0.2:22          172.16.0.2:39393      ESTABLISHED 7183/sshd: root@not
tcp        0      0 172.16.0.2:43290       103.240.141.54:3508   SYN_SENT    48941/ls
```

执行 `top -c` 命令查看是否有异常的进程。异常进程可能有以下特点：

- CPU 或内存占用高。
- 进程伪装成一个常用程序名。

查看 `/etc/crontab` 文件，看是否被添加了异常定时任务。目前 VKS 上有以下一些定时任务，这些范围外的，可以认为是被人篡改。

```
root@cvknode89:~# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
** * * * * root    /opt/bin/cha_real_check_cvm.sh
** * * * * root    /opt/bin/cha_real_check_cvk.sh
** * * * * root    /opt/bin/libvirt_d_check.sh
** * * * * root    /opt/bin/ocfs2_iscsi_conf_chg_timer.sh
*/10 * * * * root    python /opt/bin/ocfs2_cluster_config.py -s
1 2 * * * root    /opt/bin/cas_clean_log.sh
* * * * * root    /opt/bin/novnc_check.sh
* * * * * root    /opt/bin/sys_alarmd_check.sh
* * * * * root    /opt/bin/tomcat_check.sh
```

4. 恢复方法

- 更换主机密码。

- 删除异常进程，并删除磁盘上进程可执行文件。
- 删除/etc/crontab 中的异常任务。
- 重装 VKS。

3.6.7 银河麒麟操作系统的虚拟机配置 DNS 后，重启主机，配置失效

1. 故障现象

银河麒麟虚拟机配置 DNS 后，重启主机，配置失效。

2. 故障影响

影响用户虚拟机网络。

3. 故障原因

银河麒麟操作系统虚拟机的 DNS 配置文件位于 “/etc/resolv.conf”，这是一个软链接文件。

4. 恢复方法

可删除当前软链接文件 “/etc/resolv.conf”，并重新创建一个新文件 “/etc/resolv.conf”，再在新文件中添加所需的 DNS 配置信息。

3.6.8 Ubuntu 20 虚拟机配置 IPv4 地址，界面报错：修改虚拟机 xxx 配置失败

1. 故障现象

Ubuntu 20 及之后版本的操作系统，配置 IPv4 地址时，管理平台中报错：修改虚拟机 xxx 配置失败。原因：通过 Uspheretools 配置 IPV4 和 IPV6 网络信息错误。

2. 故障影响

Ubuntu 20 以及之后的版本操作系统的虚拟机，无法在管理平台页面中配置网络。

3. 故障原因

配置 IPv4 地址时，系统会在最后设置 MTU 值，如果/var/log/set-ipv6.log 日志中有 “Can not find /etc/netplan/tools-netcfgv6.yaml to update mtu” 异常报错，则会异常退出脚本的调用，导致无法配置 IP 地址。

4. 恢复方法

在虚拟机后台手动新建文件/etc/netplan/tools-netcfgv6.yaml；或者尝试先配置 IPv6 地址，然后再配置所需的 IPv4 地址。

3.7 故障告警

3.7.1 主机网卡故障

1. 故障现象

主机出现网卡故障告警。

2. 故障影响

用户业务受影响。

3. 故障原因

排查网卡是否人为 down 掉、链路是否断开、协商是否正常、查看 syslog 中是否存在存在网卡的错误日志。

4. 恢复方法

根据情况相对应的进行 up 网卡、解决链接和协商问题、升级网卡固件、更换硬件等操作，尝试恢复。

3.7.2 主机 MCE 故障告警

1. 故障现象

主机出现 MCE 故障告警，后台查看 /var/log/mce.log 存在报错或告警。

2. 故障影响

该主机出现硬件故障，需进行紧急修复。

3. 故障原因

可能是主机服务器内存、主板等硬件故障，需检查服务器硬件日志是否存在异常。

4. 恢复方法

尽快将虚拟机迁移至其他主机，并将该主机隔离，更换故障硬件。

3.8 其他异常

3.8.1 界面无法正常登录

1. 故障现象

前台界面无法正常登录。

2. 故障影响

前台界面无法正常登录，无法完成对平台的管理。

3. 故障原因

管理平台页面的相关服务可能出现异常。

4. 恢复方法

通过 SSH 登陆 VMS 节点后台，执行 `service tomcat8 status` 命令，查看 tomcat8 是否正常运行。

- 如果 tomcat8 处于关闭状态，则执行 `service tomcat8 restart` 命令重启 tomcat8。
- 如果 tomcat8 处于运行状态，执行 `ps -ef | grep tomcat8` 命令获取 tomcat8 的 PID。

执行 `jstack -l PID > /vms/文件名称命令`，获取 tomcat8 堆栈文件。

执行 `jmap -dump:format=b,file=/vms/文件名 PID` 命令，获取 tomcat8 内存映像。

执行 `service tomcat8 restart` 命令重启 tomcat。

将上述收集文件交给工程师方便定位问题。

- 日志中报 rabbitmq 初始化失败：rabbitmq 服务正常(执行 `service rabbitmq-server status` 命令查看)，系统时间不对，早于 2021 年 5 月。尝试修改系统时间为当前时间，重启主机。
- rabbitmq 服务异常，报错“Too short cookie string”，参考下图。

```
root@VM-CAS08-CW08: /var/log/rabbitmq# service rabbitmq-server status
(error_logger {{2022,5,14},{18,27,59}}: "Too short cookie string" [])
(error_logger {{2022,5,14},{18,27,59}}: crash_report([[initial_call,{auth,init,['Argument_1']},{pid,<0.19.0>},{registered_name,[]},{error_info,{exit,{"T
oo short cookie string",{auth,init_cookie,0},{auth,init,1},{gen_server,init_it,6},{proc_lib,init_p_do_apply,3}}},{gen_server,init_it,6},{proc_lib,init_p
_do_apply,3}}],{ancestors,{inet_sup,kernel_sup,<0.9.0>}},{messages,[]},{links,<0.17.0>}},{dictionary,[]},{trap_exit,true},{status,running},{heap_size,987
},{stack_size,24},{reductions,827}]) [])
(error_logger {{2022,5,14},{18,27,59}}: supervisor_report[{supervisor,{local,net_sup}},{errorContext,start_error},{reason,"Too short cookie string",{aut
h,init_cookie,0},{auth,init,1},{gen_server,init_it,6},{proc_lib,init_p_do_apply,3}}],{offender,[{pid,undefined},{name,auth},{mfargs,{auth,start_link,[]}}
},{restart_type,permanent},{shutdown,2000},{child_type,worker}}]})
(error_logger {{2022,5,14},{18,27,59}}: supervisor_report[{supervisor,{local,kernel_sup}},{errorContext,start_error},{reason,shutdown},{offender,[{pid,undefi
ned},{name,net_sup},{mfargs,{erl_distribution,start_link,[]}},{restart_type,permanent},{shutdown,infinity},{child_type,supervisor}}]})
(error_logger {{2022,5,14},{18,27,59}}: std_info,[{application,kernel},{exited,{shutdown,{kernel,start,[normal,[]]}}},{type,permanent}}]})
"kernel pid terminated (application_controller) ({application_start_failure,kernel,{shutdown,{kernel,start,[normal,[]]}}})
kernel pid terminated (application_controller) ({application_start_failure,kernel,{shutdown,{kernel,start,[normal,[]]}}})
root@VM-CAS08-CW08: /var/log/rabbitmq#
```

– 排查磁盘根目录 (/) 是否已满。

– 删除 /var/lib/rabbitmq/.erlang.cookie 文件。

- 启动 rabbitmq (执行 `service rabbitmq-server start` 命令)，查看 cloud 用户权限。

```
15:11:51 [root@cvknode3204 ~]#rabbitmqctl list_permissions
Listing permissions for vhost "/" ...
user  configure      write  read
cloud  .*                .*     .*
guest  .*                .*     .*
```

如果 cloud 用户权限不对，需要重新初始化 rabbitmq 用户权限，执行更新包中的脚本：`rabbitmq-init.sh`。

- o rabbitmq 服务异常，执行 `rabbitmq-server` 命令启动 rabbitmq 服务，报日志目录（`/var/log/rabbitmq`）权限不足，参考下图：

```
2023-05-22 14:58:22.044059+08:00 [error] <0.224.0>
2023-05-22 14:58:22.044059+08:00 [error] <0.224.0> BOOT FAILED
2023-05-22 14:58:22.044059+08:00 [error] <0.224.0>
2023-05-22 14:58:22.044059+08:00 [error] <0.224.0> failed to open log file at '/var/log/rabbitmq/rabbit@localhost.log', reason: permission denied
2023-05-22 14:58:22.044059+08:00 [error] <0.224.0>
BOOT FAILED
failed to open log file at '/var/log/rabbitmq/rabbit@localhost.log', reason: permission denied
```

手动创建 rabbitmq 日志目录：`/var/log/rabbitmq`。

如果有该目录，尝试赋值（rabbitmq 用户）权限。

- o 启动 rabbitmq 服务。rabbitmq 服务异常，尝试执行 `rabbitmq-server` 命令启动，启动需时很长，最终超时，可参考下图。

```
{error_logger,{{2023,1,13},{11,1,1}},"Protocol: -p: register error: -p-n,['inet_tcp',{badmatch,{error,timeout}},[{inet_tcp_dist,listen,1},{net_kernel,sta
s,4},{net_kernel,start_protos,3},{net_kernel,init_node,2},{net_kernel,init,1},{gen_server,init_it,6},{proc_lib,init_p_do_apply,3}]}}
{error_logger,{{2023,1,13},{11,1,1}},Crash_report,[{initial_call,{net_kernel,init,['Argument__1']},{pid,<0.20.0>},{registered_name,[]},{error_info,{exit,{e
arg},{gen_server,init_it,6},{proc_lib,init_p_do_apply,3}}},{ancestors,[net_sup,kernel_sup,<0.9.0>}],[messages,[]],[links,[#Port<0.90>,<0.17.0>}],[dictionary
names,false]}],[trap_exit,true},{status,running},{heap_size,610},{stack_size,24},{reductions,554}],[]}
{error_logger,{{2023,1,13},{11,1,1}},supervisor_report,[{supervisor,{local,net_sup}},{errorContext,start_error},{reason,'EXIT',nodistribution},{offender,[{
fined},{name,net_kernel},{mfargs,[net_kernel,start_link,[{rabbitmqprelaunch52616,shortnames}]}},{restart_type,permanent},{shutdown,2000},{child_type,worker}]}
{error_logger,{{2023,1,13},{11,1,1}},supervisor_report,[{supervisor,{local,kernel_sup}},{errorContext,start_error},{reason,shutdown},{offender,[{pid,undefine
d,net_sup},{mfargs,[erl_distribution,start_link,[]]}},{restart_type,permanent},{shutdown,infinity},{child_type,supervisor}]}]}
```

- 排查 iptables 中禁用了 rabbitmq 需要的端口号：4369 和 5672。
- 检查 iptables 规则，放行 rabbitmq 需要的端口。
- 启动 rabbitmq 服务。

3.8.2 前台界面无法正常打开，日志中报连接数据库失败

1. 故障现象

前台界面无法正常打开，日志中报连接数据库失败。

2. 故障影响

前台界面无法正常登录，无法完成对平台的管理。

3. 故障原因

- 系统异常断电导致数据库服务异常。
- 数据库文件权限错误。
- 数据库空间满。

4. 恢复方法

- 断电导致数据库服务无法启动，请联系 UniCloud 技术支持工程师处理；
- 数据库数据目录、日志目录以及其中的文件属主需为 `mysql`，若不是，则执行 `chown mysql:mysql -R /var/lib/mysql`（双机为 `/var/lib/mysql-share`）命令修改属主，`/var/log/mariadb` 目录需做相同处理。

```
-rw-rw---- 1 mysql mysql 16K May 23 10:06 aria_log.00000001
-rw-rw---- 1 mysql mysql 52 May 23 10:06 aria_log_control
drwx----- 2 mysql mysql 12K May 23 00:10 cas_cic
-rw-rw---- 1 mysql mysql 13K May 23 10:06 ib_buffer_pool
-rw-rw---- 1 mysql mysql 76M May 23 10:06 ibdata1
-rw-rw---- 1 mysql mysql 48M May 23 10:06 ib_logfile0
-rw-rw---- 1 mysql mysql 48M May 23 10:06 ib_logfile1
drwx----- 2 mysql mysql 4.0K May 20 17:20 licc
-rw-rw---- 1 mysql mysql 0 May 20 17:20 multi-master.info
drwx--x--x 2 mysql mysql 4.0K May 20 17:16 mysql
drwx----- 2 mysql mysql 4.0K May 20 17:16 performance_schema
drwxr-xr-x 2 mysql mysql 4.0K May 20 17:16 test
drwx----- 2 mysql mysql 4.0K May 20 17:21 vmware
drwx----- 2 mysql mysql 20K May 23 07:00 vservice
```

- 空间满导致的创建表失败，请联系 UniCloud 技术支持工程师处理。

3.8.3 前台界面查看虚拟机、存储池等信息卡住

1. 故障现象

前台界面查看虚拟机、存储池等信息卡住，无法继续正常操作。

2. 故障影响

前台界面卡住，无法正常进行管理操作。

3. 故障原因

- 存在销毁虚拟机、备份、快照等未完成的任务。
- 存储 IO 压力较大。
- 存储发生故障，如 fence umount。

4. 恢复方法

- 等待任务完成后再操作，前台任务台上可查看任务进度。
- 可参考 4.2.1 章节解决存储 IO 压力大的问题。
- 在系统后台检查存储池是否正常，排查存储故障。

3.8.4 虚拟机内部操作卡顿问题

1. 故障现象

虚拟机内部操作卡顿问题，并且管理平台界面中虚拟机图标变为蓝色，不能修改虚拟机，不能访问 VNC 控制台，无法 ping 通虚拟机网络。

2. 故障影响

影响用户虚拟机业务。

3. 故障原因

- 虚拟机磁盘所在存储池异常，导致磁盘的读写不能返回，qemu 主进程一直在等待内核层的返回结果。
- 虚拟机开启了防病毒，而防病毒模块的后端驱动（杀毒厂商在 VKS 上安装的软件）异常，导致 qemu 进程在等待内核层返回。
- 云桌面环境中，一般使用 spice 组件。该组件需要虚拟机 qemu 进程配合，如果组件异常，会导致 qemu 的调用一直不能结束，引起阻塞。
- 虚拟机在线迁移存储时会对虚拟机磁盘限速，导致虚拟机内部运行速度慢，这种情况是正常现象。
- 虚拟机磁盘大小一般为若干 TB，而虚拟机内部某些业务容易产生磁盘碎片，当碎片率达到 50% 及以上时，再做快照操作，就会引起虚拟机内部卡顿，该问题是 qcow2 格式磁盘自身的缺陷。
- 由虚拟机上的其他有问题的设备引起，例如：USB 设备异常。

4. 恢复方法

- 如果虚拟机正在做迁移存储、或者删除快照操作，虚拟机运行速度慢是正常现象，尤其是磁盘数据量达到 TB 级别时。
- 查看虚拟机运行态配置，确定虚机有什么不常用的配置，如虚机开启了杀毒、挂载 ukey、U 盘、移动硬盘等 USB 设备、是否有透传设备等，可以查看虚拟机 qemu 日志 (/var/log/libvirt/qemu/虚拟机名称.log)，看看异常时间点前后有哪些报错信息，可初步了解虚拟机到底是哪个设备出问题。
- 执行 `ps aux | grep 虚拟机名称` 命令，查看其是否为 D 状态或 Z 状态。
- 执行 `top -c -H -p 虚拟机 qemu 进程 pid` 命令，看进程的 CPU、memory 使用率是否异常。执行 `cat /proc/虚拟机 qemu 进程 pid/stack` 命令，可以查看进程栈。

3.8.5 USB 插到 VKS 主机上后，主机无法识别到该设备

1. 故障现象

将 USB 设备插到主机后，在管理平台中为虚机添加 USB 设备时找不到该设备。

2. 故障影响

无法使用 USB 设备。

3. 故障原因

- USB 插入的插槽不正确。
- USB 设备本身存在问题。

4. 恢复方法

更换 USB 插槽，可通过执行 `lsusb -t` 命令，检查 USB 设备插入的插槽是否正确。例如：

```
root@cvk-163:~# lsusb -t
/: Bus 04.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/6p, 5000M
/: Bus 03.Port 1: Dev 1, Class=root_hub, Driver=xhci_hcd/15p, 480M
/: Bus 02.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
|__ Port 1: Dev 2, If 0, Class=hub, Driver=hub/8p, 480M
/: Bus 01.Port 1: Dev 1, Class=root_hub, Driver=ehci-pci/2p, 480M
|__ Port 1: Dev 2, If 0, Class=hub, Driver=hub/6p, 480M
```

命令返回结果里的 UHCI 表示 USB1.1、EHCI 表示 USB2.0、XHCI 表示 USB3.0。一般 USB1.1 最大传输速率为 12Mbps，USB2.0 最大传输速率为 480Mbps，USB3.0 最大传输速率为 5Gbps。

如果服务器支持多种 USB 总线标准，给服务器添加一个 USB2.0 设备后，在 USB2.0 (ehci-pci) 的总线下新增一个 USB 设备，则说明 USB 设备插的插槽是正确的。

如果进行上述操作后，仍然无法正常识别，则在拔插 USB 设备前后，执行 `lsusb` 命令查看是否有新增设备，例如：

```
root@ cvk:~# lsusb
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 005 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 004 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 003 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 006 Device 002: ID 03f0:7029 Hewlett-Packard
Bus 006 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

若无新增设备，说明系统识别不到，建议更换 USB 设备再进行尝试。

3.8.6 修改系统时间导致集群开启 HA 失败，报“HA 进程响应超时”

1. 故障现象

VMS 主机向过去修改超过 10 天的时间，集群开启 HA 时失败，任务台报“HA 进程响应超时”。

任务台 清除已完成

任务名称	操作对象	任务状态	详情
集群“cluster0”启用HA。	cluster0	● 失败	集群“cluster0”启用HA失败。原因：HA进程响应超时。

2. 故障影响

无法开启集群高可靠性功能。

3. 故障原因

后台 HA 进程连接数据库失败，导致 HA 进程无法正常启动。日志有报如下错误信息：**SSL connection error: protocol version mismatch.**

4. 恢复方法

- 将主机时间修改为与当前时间一致。
- 在文件/etc/mysql/mysql.conf.d/mysql.cnf 中增加 skip_ssl 选项。

4 故障信息收集

在应急恢复操作完成后，需要需要收集故障信息作为处理故障的参考，或者作为将故障信息提供给 UniCloud 技术支持人员，以便彻底定位和排除故障。

4.1 在系统后台收集日志信息

请联系 UniCloud 技术支持工程师处理。

4.2 在管理平台中收集日志文件

- (1) 选中顶端的“系统”页签，单击左侧导航树中的[操作日志/日志文件收集]菜单项，进入收集日志文件页面。
- (2) 在日志文件收集区域，输入物理主机日志文件大小，选择物理主机日志时间范围，并选择一个或多个主机，单击<日志文件收集>按钮，开始收集日志文件。

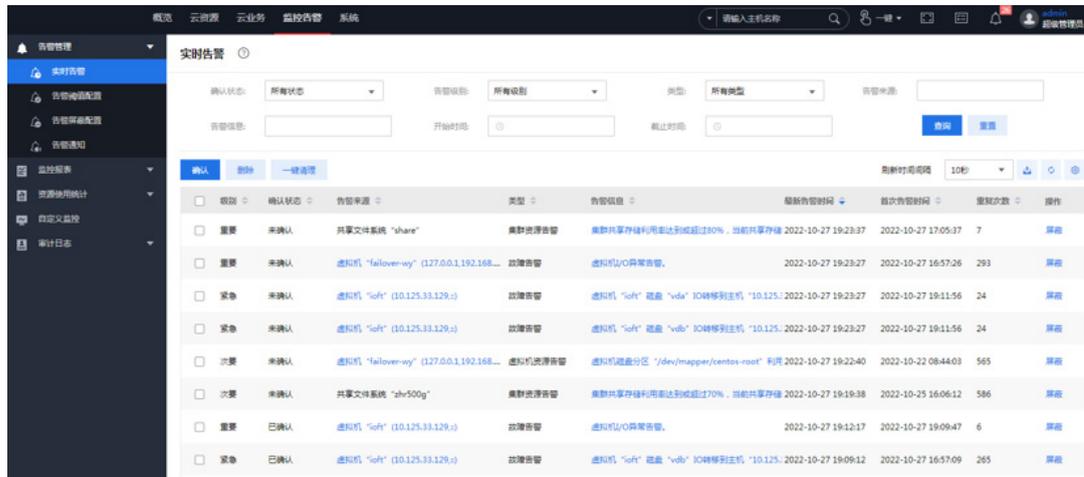


- (3) 收集完成后，单击<日志文件下载>按钮，单击<下载>按钮将日志文件下载到本地保存。

4.3 查看故障告警信息

- (1) 在管理平台中，选择顶部“监警告警”页签，单击导航树中[告警管理/实时告警]菜单项，进入实时告警列表页面。单击<告警信息>按钮，将弹出查看告警详细信息对话框。

图3 查看实时告警信息



- (2) 设置过滤查询参数后，单击<查询>按钮查询符合要求的告警信息。单击  按钮，下载当前告警列表。