

文件存储  
故障处理  
产品文档



腾讯云

**【版权声明】**

©2013-2024 腾讯云版权所有

本文档著作权归腾讯云单独所有，未经腾讯云事先书面许可，任何主体不得以任何形式复制、修改、抄袭、传播全部或部分本文档内容。

**【商标声明】**

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。

**【服务声明】**

本文档意在向客户介绍腾讯云全部或部分产品、服务的当时的整体概况，部分产品、服务的内容可能有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或模式的承诺或保证。

---

## 文档目录

故障处理

小文件及高并发场景下客户端使用卡顿

# 故障处理

## 小文件及高并发场景下客户端使用卡顿

最近更新时间：2024-01-22 22:15:48

### 背景

文件存储（Cloud File Storage, CFS）支持 NFS v3.0 及 NFS v4.0 协议，其中 NFS v3.0 是 NFS 协议较早期版本，兼容 Windows 客户端；NFS v4.0 协议为稍后期版本，支持文件锁等功能。

### 客户端现象

在大量小文件或者大小文件混合场景下，用户在容器服务（Tencent Kubernetes Engine, TKE）或者云服务器（Cloud Virtual Machine, CVM）等客户端使用 NFS v4.0 协议挂载 CFS，可能在应用运行一段时间后出现：客户端负载居高不下，无限累加，业务读取数据慢或无响应，但是业务进程的 CPU 使用率并不是很高的情况。

### 问题原因

上述问题出现的主要原因是 NFS v4.0 协议的限制，客户端使用 NFS v4.0 同时大批量读写文件时由于协议的 OPEN/CLOSE 操作为串行化，因此在客户端大量操作请求并发情况下，协议处理会形成阻塞。具体协议描述如下：  
There is a limitation to the Linux NFS4.0 client implementation that an "open\_owner" is mapped to a userid. This results in a bottleneck if one user opens and closes a lot of files in a short period of time. Each OPEN / CLOSE operation has to wait for a sequence id, which essentially serializes each OPEN / CLOSE request. If an NFS server's response time for OPEN / CLOSE requests increases due to some secondary load or complication, this NFS4 client limitation can become pronounced, and in some cases, cause an unresponsive machine.

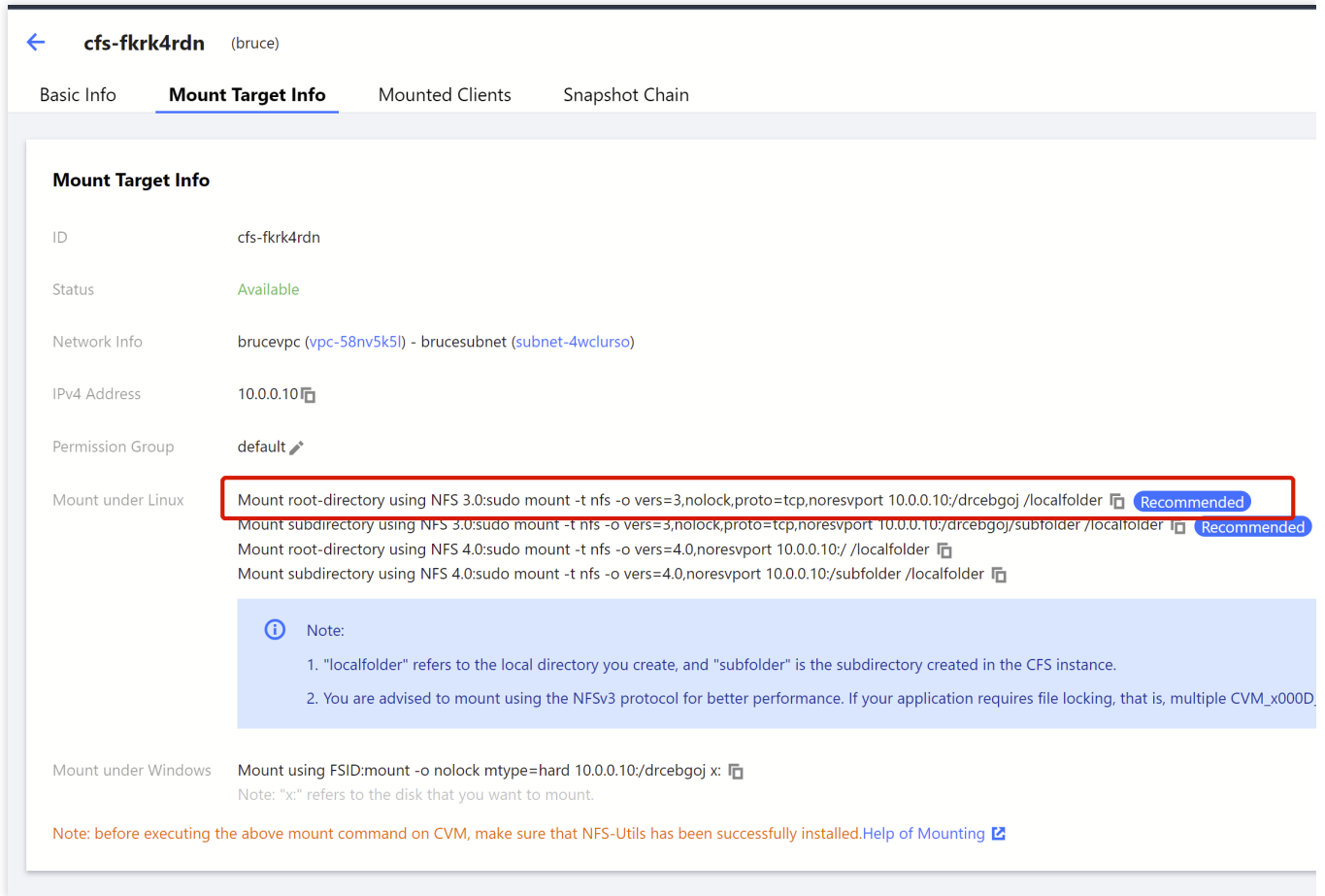
The NFS4.1 protocol addresses the limitation of serialization of OPENs per open\_owner. For more information, see [RFC 5661 Section 9.10](#).

### 使用优化

如果业务应用存在大量小文件的场景，或者并发操作文件数量巨大，推荐客户端使用 NFS v3.0 协议挂载。以下为规避大量小文件及大并发请求下客户端高负载的问题的方法。

#### 云服务器等客户端挂载方法

进入 [文件存储](#) 控制台，打开待挂载的文件系统详情中的[挂载点信息](#)，找到如下图的 NFS v3.0 挂载命令。使用该挂载命令挂载文件系统即可。



## 容器客户端挂载方法

容器可以通过 pv/pvc 的方式使用 NFS v3.0 协议进行挂载，参考配置如下。



```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: cfs-test-pv
spec:
  accessModes:
    - ReadWriteMany
  capacity:
    storage: 9000Gi
  mountOptions:
    - vers=3
```

```
- nolock
- proto=tcp
- noatime
- nodiratime
- noexec
- hard
- rsize=524288
- wsize=524288
nfs:
  path: /[cfs-id]/[mount-path]/
  server: [cfs-server-ip]
persistentVolumeReclaimPolicy: Retain
storageClassName: cfs-test-pv
volumeMode: Filesystem
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: cfs-test-pvc
  namespace: default
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 9000Gi
  storageClassName: cfs-test-pv
  volumeMode: Filesystem
  volumeName: cfs-test-pv
```