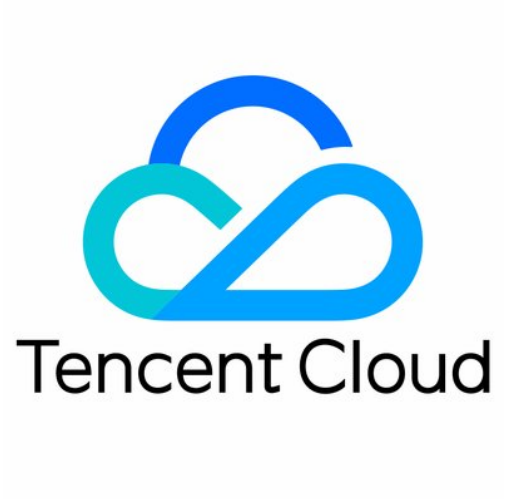
Cloud File Storage

Troubleshooting

Product Documentation



Contents

Troubleshooting
Client Use Bottleneck due to Large Number of Small Files or Parallel Requests

Last updated：2021-12-16 12:22:50

Background

CFS supports both NFS v3.0 and NFS v4.0 (later version). NFS v3.0 is compatible with Windows clients while NFS v4.0 provides file locking and other features.

Client Problem

In cases that involve a large number of small files or mixed use of large and small files, problems may occur when you mount a CFS file system onto a Tencent Kubernetes Engine（TKE） or Cloud Virtual Machine（CVM）client using NFS v4.0. After an application runs on the client for a period of time, you may find the client load remains high and keeps accumulating. Besides, the service data may be read slow or no response is returned, but the CPU utilization of the service process is not very high.

Cause

The above problems happen mainly due to the NFS v4.0 limitation as described below. If the client uses NFS v4.0 to read and write a lot of files at the same time, the large number of OPEN/CLOSE requests in parallel may result in a bottleneck on the client because the OPEN/CLOSE operations are serialized in NFS v4.0.

- 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.

Solution

In cases where your service application involves a massive number of small files, or files to operate on in parallel, it is recommended that you mount file systems onto your client using NFS v3.0 to avoid high client load. The following describes how to do so.

Mount method for CVM client

Open the Cloud File Storage console, and click on the name of the file system to mount. Select the **mount Target Info** tab, and find the NFS v3.0 mount command as shown below. Then, use this command to mount the file system.

Mount method for TKE client

You can mount file systems onto a TKE client through PVs/PVCs using NFS v3.0. The example configuration is as shown below:

```
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
```