

# 优化并测试实例性能

本页介绍 Filestore 的预期平均性能和建议的性能设置。它还向您展示了如何测试 Filestore 实例的性能。

## 预期表现

每个 Filestore 服务层都提供不同级别的性能。由于各种因素（例如缓存的使用、客户端 VM 的数量、客户端 VM 的机器类型以及测试的工作负载），任何给定实例的性能可能与预期数字有所不同。 (/compute/docs/machine-types)

下表显示了 Filestore 实例基于其服务层 (/filestore/docs/service-tiers)和配置容量的预期性能：

表现	容量 (TiB)	读/写IOPS	读/写吞吐量 (MiB/s)
BASIC_HDD	1–10	600/1,000	100/100
BASIC_HDD	10–63.9	1,000/5,000	180/120
BASIC_SSD	2.5–63.9	60,000/25,000	1,200/350
ZONAL（分区、低容量带）	1	9,200/2,600	260/88
ZONAL（分区、低容量带）	9.75	89,700/25,350	2,535/858
HIGH_SCALE_SSD（现在是区域性、高容量带）	10	92,000/26,000	2,600/880
HIGH_SCALE_SSD（现在是区域性、高容量带）	100	920,000/260,000	26,000/8,800
ENTERPRISE	1	12,000/4,000	120/100
ENTERPRISE	10	120,000/40,000	1,200/1,000

上表显示了具有较高容量范围的区域层（之前为大规模 SSD）和企业层在最大和最小容量下的预期性能。在这些限制之间，性能随着容量的扩展而线性扩展 (/filestore/docs/editing-instances#tier-scaling)。例如，如果您将企业实例容量从 1 TiB 加倍到 2 TiB，则实例的预期性能从 12,000/4,000 读/写 IOPS 翻倍到 24,000/8,000 读/写 IOPS。

在单客户端和少客户端场景中，必须使用 mount 选项增加 TCP 连接数 **nconnect** (<https://man7.org/linux/man-pages/man5/nfs.5.html>) 才能获得最大 NFS 性能。<sup>7</sup>我们建议为区域服务层指定最多连接数，2 为企业层指定最多连接数。一般来说，文件共享容量越大，连接的客户端虚拟机越少，通过指定附加连接获得的性能就越高**nconnect**。

## 推荐的客户端机器类型

我们建议使用 Compute Engine [机器类型](/compute/docs/machine-types) (/compute/docs/machine-types)，例如 `n2-standard-8` 提供 16 的出口带宽 Gbps。Gbps 此出口带宽允许客户端为缓存友好的工作负载实现大约 16 个读取带宽。有关其他上下文，请参阅 [网络带宽](/compute/docs/network-bandwidth) (/compute/docs/network-bandwidth)。

## Linux 客户端挂载选项

我们建议使用以下 NFS 挂载选项，尤其是 `hard mount`、`async`、和 `rsize` 选项 `wsizesize`，以在 Linux 客户端 VM 实例上实现最佳性能。有关 NFS 挂载选项的更多信息，请参阅 [nfs](https://linux.die.net/man/5/nfs) (https://linux.die.net/man/5/nfs)。

默认选项	描述
<code>hard</code>	NFS 客户端无限期地重试 NFS 请求。
<code>timeo=600</code>	NFS 客户端在重试 NFS 请求之前等待 600 分秒（60 秒）。
<code>retrans=3</code>	NFS 客户端在采取进一步恢复操作之前尝试 NFS 请求 3 次。
<code>rsizesize=262144</code>	NFS 客户端每次请求最多可以从 NFS 服务器接收 262,144 字节 READ。 注意：对于基础层实例，将 <code>rsizesize</code> 值设置为 <code>1048576</code> 。
<code>wsizesize=1048576</code>	NFS 客户端每次请求最多可以向 NFS 服务器发送 1,048,576 字节 (1 MiB) WRITE。
<code>resvport</code>	NFS 客户端在与此挂载点的 NFS 服务器通信时使用特权源端口。
<code>async</code>	NFS 客户端延迟将应用程序写入发送到 NFS 服务器，直到满足某些条件。 注意：使用该 <code>sync</code> 选项会显著降低性能。

## 单客户端和多客户端虚拟机性能

Filestore 的可扩展服务层针对多个客户端虚拟机（而不是单个客户端虚拟机）进行了性能优化。

对于 [企业级](/filestore/docs/service-tiers#enterprise_tier) (/filestore/docs/service-tiers#enterprise\_tier) 和 [区域级](/filestore/docs/service-tiers#high_scale_ssd_tier) (/filestore/docs/service-tiers#high\_scale\_ssd\_tier) 实例，至少需要四个客户端虚拟机才能充分利用性能。这可确保底层文件存储集群中的所有虚拟机都得到充分利用。

为了增加上下文，最小的可扩展文件存储集群有四个虚拟机。无论使用挂载选项指定的每个客户端的 NFS 连接数是多少，每个客户端 VM 都仅与一个 Filestore 群集 VM 进行通信

**nconnect** (<https://man7.org/linux/man-pages/man5/nfs.5.html>)。如果使用单个客户端 VM，则只能从单个 Filestore 群集 VM 执行读取和写入操作。

## 提高 Google Cloud 资源的性能

跨多个 Google Cloud 资源的操作（例如使用 将数据从 Cloud Storage 复制到 Filestore 实例）**gsutil**可能会很慢。为了帮助缓解性能问题，请尝试以下操作：

- 确保 Cloud Storage 存储桶、客户端虚拟机和 Filestore 实例均位于同一区域 (/filestore/docs/regions)。

双区域 (/storage/docs/dual-regions)为 Cloud Storage 中存储的数据提供最佳性能选项。如果使用此选项，请确保其他资源驻留在构成双区域的单个区域之一。例如，如果您的 Cloud Storage 数据驻留在 中 **us-central1,us-west1**，请确保您的客户端虚拟机和 Filestore 实例驻留在 中**us-central1**。

- 作为参考，请验证连接 PD 的虚拟机的性能并与 Filestore 实例的性能进行比较。
  - 如果与 Filestore 实例相比，PD 连接的虚拟机的性能相似或较低，则这可能表明存在与 Filestore 无关的性能瓶颈。请尝试以下操作来提高非 Filestore 资源的基准性能：
    - 应用该**gsutil -m** (/storage/docs/gsutil/addlhelp/GlobalCommandLineOptions#options) 选项。此选项允许支持的操作（包括**cp**、**mv**和 ）**rsync**并行运行。
    - 应用以下**gsutil**配置。每个都是**boto** (/storage/docs/boto-gsutil) 在配置文件中设置的可变配置值**boto**。用户可以直接修改文件（推荐）或使用以下**-o** (/storage/docs/gsutil/addlhelp/GlobalCommandLineOptions#options) 选项：
      - **parallel\_process\_count** (/storage/docs/boto-gsutil#config-file-structure)
      - **parallel\_thread\_count** (/storage/docs/boto-gsutil#config-file-structure)
      - **sliced\_object\_download\_max\_components** (/storage/docs/boto-gsutil#config-file-structure)

★ **注意：**没有对限制请求的内置支持。用户应尝试请求的值，因为最佳值可能会因网络速度、CPU 数量和可用内存等多种因素而异。 **gsutil**

- 如果 Filestore 实例的性能明显低于附加 PD 的 VM，请尝试将操作分散到多个 VM。

- 这有助于提高 Cloud Storage 读取操作的性能。
- 对于企业级 (/filestore/docs/service-tiers#enterprise\_tier)和 区域级 (/filestore/docs/service-tiers#high\_scale\_ssd\_tier) 实例，至少需要四个客户端虚拟机才能充分利用性能。这可确保底层文件存储集群中的所有虚拟机都得到充分利用。有关详细信息，请参阅单个和多个客户端 VM 性能 (/filestore/docs/performance#single-multiple-performance)。

## 测试性能

如果您使用的是 Linux，则可以使用 `fio` (<https://linux.die.net/man/1/fio>) 工具对基本层实例的读写吞吐量和 IOPS 进行基准测试。请注意，不建议将这种性能基准测试方法用于企业和区域层实例。

本节中的示例显示了您可能想要运行的常见基准测试。您可能需要从多个客户端 VM 实例运行 `fio` 才能获得最大性能。

以下示例对最大写入吞吐量进行基准测试：

```
fio --ioengine=libaio --filesize=32G --ramp_time=2s \
--runtime=5m --numjobs=16 --direct=1 --verify=0 --randrepeat=0 \
--group_reporting --directory=/mnt/nfs \
--name=write --blocksize=1m --iodepth=64 --readwrite=write
```

以下示例对最大写入 IOPS 进行基准测试：

```
fio --ioengine=libaio --filesize=32G --ramp_time=2s \
--runtime=5m --numjobs=16 --direct=1 --verify=0 --randrepeat=0 \
--group_reporting --directory=/mnt/nfs \
--name=randwrite --blocksize=4k --iodepth=256 --readwrite=randwrite
```

以下示例对最大读取吞吐量进行了基准测试：

```
fio --ioengine=libaio --filesize=32G --ramp_time=2s \
--runtime=5m --numjobs=16 --direct=1 --verify=0 --randrepeat=0 \
--group_reporting --directory=/mnt/nfs \
--name=read --blocksize=1m --iodepth=64 --readwrite=read
```

以下示例对最大读取 IOPS 进行基准测试：

```
fio --ioengine=libaio --filesize=32G --ramp_time=2s \  
--runtime=5m --numjobs=16 --direct=1 --verify=0 --randrepeat=0 \  
--group_reporting --directory=/mnt/nfs \  
--name=randread --blocksize=4k --iodepth=256 --readwrite=randread
```

## 下一步是什么

- ([/filestore/docs/troubleshooting#slow\\_performance](#)) 解决Filestore 的性能相关问题。  
([/filestore/docs/troubleshooting#slow\\_performance](#))

除非另有说明，本页面的内容根据[Creative Commons Attribution 4.0 License](#)

(<https://creativecommons.org/licenses/by/4.0/>)获得许可，代码示例根据[Apache 2.0 License](#)

(<https://www.apache.org/licenses/LICENSE-2.0>)获得许可。有关详细信息，请参阅[Google 开发者网站政策](#)

(<https://developers.google.com/site-policies>)。Java 是 Oracle 和/或其附属公司的注册商标。

最后更新时间：世界标准时间 2023 年 9 月 14 日。