

FS 作业 4

钟赧 2016K8009915009

1. 假设一个 NFS 服务器使用 5 个磁盘构成的一个 RAID-5，每个条带为 64KB，每个磁盘的寻道时间是 3ms，旋转速度是 10,000RPM，传输带宽是 200MB/s。假设网络传输 4KB 的延迟是 0.2ms，传输 1MB 的延迟是 9ms，请问：

- 1) 客户端用 NFS 一次读一个数据块（4KB）的有效带宽是多少？

假设平均旋转半圈，平均旋转延迟为： $0.5/10000 \times 60000\text{ms} = 3\text{ms}$ ；

传输 4KB 文件所需时间： $4\text{K}/200(\text{M/s}) = 0.0195\text{ms}$ ；

网络延迟为 0.2ms；寻道时间为 3ms；

总延迟时间为： $3 + 0.0195 + 0.2 + 3 = 6.2195\text{ms}$ ；

有效带宽： $4\text{KB}/6.2195\text{ms} = 643.14\text{KB/s}$ 。

- 2) 客户端用 NFS 一次读 1MB 数据的有效带宽是多少？

RAID-5 可并行传输，传输 1MB 文件所需时间： $1\text{M}/200(\text{M/s})/5 = 1\text{ms}$ ；

网络延迟为 9ms；寻道时间为 3ms；平均旋转延迟仍为 3ms；

总延迟时间为： $3 + 1 + 9 + 3 = 16\text{ms}$ ；

有效带宽： $1\text{MB}/16\text{ms} = 62.5\text{MB/s}$ 。

2. 机器 X0 是 NFS 服务器，并输出目录 /share，机器 X1、X2、X3 和 X4 都通过下面命令挂载了一个 NFS 文件系统，使用 NFS v3 协议。

`mount -t nfs X0:/share /share`

机器 X1 上的进程 P1 打开文件 /share/foo，读取文件 foo 的第一块（块 0），假设读操作成功返回的时刻为 t_0 ；之后，X2 上的进程 P2 也打开文件 /share/foo，在 $t_0 + 3\text{sec}$ 时刻读取了 foo 的块 0。然后，P2 向块 0 写入新数据，假设写入成功返回的时刻为 $t_0 + 10\text{sec}$ 。之后，P2 读写的其它块，在 $t_0 + 30\text{min}$ 时刻关闭 /share/foo。请问：

- 1) P1 最早在什么时刻能够读到 P2 写入的新数据？

P2 写入成功的时刻为 $t_0 + 10\text{sec}$ ；脏数据 30s 之后写回服务器，故服务器端数据会在 $t_0 + 40\text{sec}$ 更新。

而客户端缓存的有效期为 60s，即 X1 上的 P1 在 t_0 时刻打开文件，故至少在 $t_0 + 60\text{sec}$ 才会更新缓存和服务器同步。

所以 P1 至少在 $t_0 + 60\text{sec}$ 时才能读到 P2 写入的数据。

- 2) X3 上的进程 P3 最早在什么时刻打开 /share/foo 能够读到 P2 写入的新数据？

X3 是新机器，只要服务器端更新数据 P3 就可以读到 P2 新写入的数据。故 P3 最早在 $t_0 + 40\text{sec}$ 时就可以读到 P2 写入的数据。

- 3) X1 上的进程 Q1 最早在什么时刻打开 /share/foo 能够读到 P2 写入的新数据？

同一台机器的文件缓存在进程间共享，因此 X1 上的进程 Q1 也会使用和 P1 一样的文件缓存。故需要在 $t_0 + 60\text{sec}$ 时才能读到 P2 写的的数据

3. NFS 的 file handle 包含三个信息：volumn ID, ino, generation number。其中 generation number 是 i-node 的版本号，每次分配该 i-node 给一个文件或者目录，版本号都增长

1. 请举一个例子来说明为什么 NFS 的 file handle 必须包含 generation number。

举例：机器 X1 打开现有文件 1，获得一个 inode。然后因为某些原因文件 1 被删除，新建文件 2，其文件句柄与前一个相同。此时可以使用之前文件 1 的 inode 访问文件 2。这是不合法的

4. WAFL 的块大小都为 4KB，指针都为 4 字节，i-node 中有 16 个指针用于文件块索引。请问：

1) WAFL 最大能支持多大的文件？

每个块可以存储 $4K/4=1024$ 个间接指针。采用一级间接，共计有 $16*1024$ 个间接指针。故一共可以索引的文件大小为： $16*1024*4KB=64MB$ 。

2) 如果采用两级间接的话，最大能索引多大的文件？

共计 $16*1024*1024$ 个间接指针。

故总共可以索引的文件大小为： $16*1024*1024*4KB=64GB$ 。

3) 对于一个 10GB 的文件，WAFL 如何定位偏移(offset)为 5G 所在的文件块？

$64MB < 10GB < 64GB$ ，故需要两级间接。

每个 inode 中的指针可以指 $1*1024*1024*4KB=4GB$ ，因此 5GB 应该在 inode 中第二个指针指向的一级间接块上。一级间接块每个指针可以索引 $1*1024*4KB=4MB$ 的文件，因此是第 256 个间接指针索引的块。

故偏移为 5G 所在的文件块是 inode 中第 2 个指针指向的间接块上第 256 个指针指向的块。

5. 假设 WAFL 平均 1000 个操作的日志需要大约 1MB 的 NVRAM 空间，如果一个 NFS 服务器的吞吐率是 5000 IOPS（即每秒 3000 I/O 操作），操作的读写比是 6:4，如果每 10 秒创建一个一致点的话，它需要多大的 NVRAM？

一个 NFS 服务器每秒 3000 次 I/O 操作，操作的读写比为 6:4，那么读操作为 1800 次/秒，写操作为 1200 次每秒。每 10 秒创建一个一致点，所以 10 秒内有 12000 次写操作，日志的大小： $12000/1000*1MB=12MB$ 。