Linux: IO Performance Tuning with noatime, nodiratime, relatime

25 Oct 2014 Mohamed Ibrahim

The POSIX standard mandates that operating systems maintain file system metadata that records when each file was last accessed. This timestamp is known as `atime` and `atime` comes with a performance penalty – every read operation on a filesystem generates a write operation.

Linux keeps three timestamps for each file on the filesystem – `mtime`, `ctime` and `atime` (modified time, change time and access time). These can be displayed with the `stat` command.

```
$ stat /bin/ls
  File: `/bin/ls'
  Size: 105840          Blocks: 208        IO Block: 4096   regular file
Device: fc02h/64514d    Inode: 1701255     Links: 1
Access: (0755/-rwxr-xr-x)  Uid: (    0/    root)   Gid: (    0/    root)
Access: 2014-10-25 16:07:49.360248485 +0800
Modify: 2012-11-20 06:25:16.000000000 +0800
Change: 2014-05-11 13:31:24.443355991 +0800
```

Linux have the following time related mount options available.

- **`noatime`** – disables updating of `atime` for both files and directories
- **`nodiratime`** – disables updating of `atime` for directories
- **`relatime`** – updates `atime` attribute only if the previous `atime` is <= `mtime` or `ctime`, or the previous `atime` is over 24 hours old or if inode is dirty.
- **`strictatime`** – allows full `atime` updates

Let's look at the mount point options in more detail.

Linux Mount Option – noatime

Linux kernel developer Ingo Molnár claimed that it (`atime`) was "perhaps the most stupid Unix design idea of all times." To disable the tracking of `atime`, the `noatime` option can be used to mount filesystems. For IO intensive tasks, the performance reward for turning off `atime` can be immediately apparent. But, turning off `atime` unconditionally will occasionally break certain software like mail tools that compare `mtime` and `atime` to determine whether there is unread mail or not. The `tmpwatch` utility and some backup tools also use `atime` and can misbehave if `atime` is incorrect. Audit requirements are another reason for keeping `atime` enabled.

Linux Mount Option – nodiratime

This is the same as the `noatime` option but this only applies to directories. Note that turning on `noatime` implicitly means that `nodiratime` is enabled as well.

Linux Mount Option – relatime

The `relatime` option was introduced to mitigate the issues with the `noatime` option. If `relatime` option is used, the `atime` is updated when:

1. old `atime` is <= than `mtime`/`ctime`

2. at least 24 hours (hardcoded in fs/inode.c (https://github.com/torvalds/linux/blob/master/fs/inode.c)) have passed since the last atime update
3. inode is dirty (or the file contents have been changed but it's in the buffer and have not been written to the physical disk yet)

Improve Linux IO Performance

Since Linux 2.6.30, the kernel defaults to the behavior provided by the `relatime` option. So IO performance has been sort of optimised, but further improvements can be done.

To further improve the IO performance, you can choose to use the `noatime` mount option for selected partitions like `/` or `root` partition but leave the `/var/spool`, `/tmp` and other required partitions with `relatime` option. Below is a sample of the fstab file with the mount point options.

/etc/fstab Vim

```
1 /dev/mapper/vg00-root       /       ext4    defaults,noatime        1       1
2 /dev/mapper/vg01-home       /       ext4    defaults,relatime       1       1
3 /dev/mapper/vg02-tmp        /tmp    ext4    defaults,relatime       1       1
4 /dev/mapper/vg03-var        /var    ext4    defaults,relatime       1       1
```

Partitions like `/home` could either be mounted with `relatime` or `noatime`. It depends on the needs of the users and the applications or services being run on the system.

If you have an application that uses `atime`, study the possibility of using `mtime` or `ctime` instead. If the application works well without the `atime` option, edit the mount point options accordingly.

There are no rules to dictate which partitions will be mapped with which mount option. It depends on the applications and services being run on the system in question.

Finally, I would suggest that no matter which Linux distribution you use, explicitly define the mount point options for the behavior you need. This will prevent any unwanted issues from happening should the Linux kernel defaults are changed.

linux (https://stackpointer.io/tag/linux/), performance tuning (https://stackpointer.io/tag/performance-tuning/)