www.photocase.com

Finding the right file system

FILE FACTS

Many users just opt for the defaults and don't think about the file system when they install Linux. But if better performance is your goal, it pays to do some shopping. **BY MARCEL HILZINGER**

With today's Linux systems, you can choose a filesystem in just a few clicks, and in some cases, you are not even asked to make a decision. Most users stick to their distribution's defaults, possibly changing these values based on past experience.

But if you have an eye on performance, it is worthwhile considering your filesystem choices before you install. We took a look at some of the popular Linux filesystem options and tested them with some real-world tasks.

Judging from Linux Magazine benchmarks, XFS on kernel 2.6 is at least as fast as ReiserFS and Ext3. Reiser4 is looking to secure as big a share of the filesystem cake as possible with record speeds and a new design, however, the Reiser4 developers still have a few fundamental bugs to iron out, which means that Reiser4 still isn't really ready for productive use.

Mainstream

The vast majority of today's Linux computers use Ext3 [1] or ReiserFS [2] as their main filesystems, as these filesystems are the most common distribution defaults. ReiserFS and Ext3 are neither particularly fast, nor do they have a particularly impressive feature scope. But distributors tend to patch their chosen filesystems up to the back teeth, and this can make for enormous performance differences. In this light, it is hard to give universally applicable advice. The box titled "It All Started with Ext2" helps you understand why most distributions use Ext3 or ReiserFS.

Ext3 and ReiserFS are two members of the family of journaling filesystems. Strictly speaking, journaling refers to the fact that the filesystem writes all data twice: first in the journal, and then where it belongs. It is easy to see where a journaling filesystem loses out speedwise.

By default, Ext3 and ReiserFS use a special mode, in which the filesystem only stores the metadata – information about changes to the filesystem – in the journal, rather than the actual data. To mount an Ext3 or ReiserFS partition in this mode, you need to specify the *data=ordered* option. This is the default setting for most distributions. Table 1 gives you more information on individual mount options.

If data integrity is your main concern, and if speed is secondary, the *data = journal* mount option is recommended. This option affects Ext3 write performance by up to 50 percent. The

performance hit is still about 20 percent with larger files (see Figure 1). This option is not recommended for the root directory, but for separate data partitions; it can't be unequivocally recommended for the */home* directory. The *data =writeback* option, which allows the filesystem to write to the journal before the data reach their final destination, gives you a performance gain of about 10 percent. This option is useful for the root filesystem, in which many write operations occur, and where data loss is not an issue. Of course, a separate */home* or data directory is recommended in this case.

Ext2 shows excellent performance for its age. In many tests, the veteran is quicker than Ext3 or ReiserFS. Again, this goes to show that Ext3's journaling abilities lead to a performance hit in comparison with Ext2.

For partitions without critical data, such as a separate */tmp* partition, you might prefer Ext2 to Ext3, if the partition size is below one Gigabyte. Ext2 is not recommended in all other cases, as the filesystem check can take longer than restoring a backup. Ext2 is still the filesystem of choice for separate boot partitions, as the ReiserFS journal takes up 33MB.

Reiser4 and XFS

If performance is at the top of list of filesystem requirements, Reiser4 has to be your choice. The Linux Magazine benchmarks confirm the claim on the Namesys homepage, stating that Reiser4 is twice as fast as ReiserFS. Reiser4 won more tests than any other filesystem. It is three times as fast as the second-placed filesystem, XFS [3], when creating 50,000 files. This is clear evidence of how much work the Reiser4 developers have put into optimizing write performance. And Reiser4 is at least 10 to 20 times faster than its nearest rival in all other write operations. It can rightly claim to be the quickest Linux filesystem.

Resource-wise, things don't look quite so good for Reiser4. For example, Reiser4 caused 26 percent CPU load during the sequential file create test with 50,000 files. All the other filesystems are happy with a CPU load of between 1 and 4 percent. As a rule, Reiser4 causes about 10 percent more CPU load than all other filesystems, but this additional load can easily rocket to 50 or more percent. In fact, ReiserFS was the only filesystem to need more CPU cycles than Reiser4 in a few test categories. If your machine has a low-powered CPU, you should stick with Ext3 or XFS.

Reiser4's feature list also lacks a number of basic features that are becoming more important to Linux operations, such as quotas and support for access control lists.

XFS has what it takes to become the number 1 filesystem. In contrast to Reiser4, it has seen its fair share of production use, and it supports quotas, ACLs, and extended attributes. XFS is the fastest of the filesystems on test, after Reiser4, and it actually beats Reiser4 in some categories. XFS performs particularly well with large volumes of data. The performance was excellent with a 4GB file size. If you intend to edit videos

It All Started with Ext2

If you have been using Linux for some time now, you may remember the days when more or less every Linux distribution used Ext2 as its default filesystem. Ext2 was the most popular Linux filesystem for no less than eight years. The earliest version of Linux used the Minix filesystem. A group of developers started programming a new filesystem in 1992, the Extended Filesystem, or ExtFS for short. Unfortunately, ExtFS was riddled with bugs, and a year later Rémy Card released the Second Extended Filesystem, Ext2.

Ext2 developed into a stable and extensible filesystem, although it had one drawback. As Ext2 didn't use journaling, time-consuming filesystem checks had to be performed if the filesystem crashed and at regular intervals. Depending on the hard disk capacity, a check could take several hours. Additionally, users required proprietary tools to extend or shrink the filesystem on the fly. And by the turn of the millennium, hard disks had become too big, and filesystem checks with Ext2 too slow.

Anticipating this issue, the kernel developers had already launched two separate projects to find a solution. One project aimed to develop Ext3 as a journaling extension for Ext2, and the other to develop a completely new filesystem with native journaling support, ReiserFS v3. ReiserFS won the race to find the first journaling filesystem in 1999. Suse, the main sponsor of the journaling code, was extremely interested in resizing on the fly, and Suse was the first distribution to introduce the new filesystem as its standard when Suse 6.4 was released in spring 2000. (It had already been made available as an update for 6.3). ReiserFS still had some teething trouble; for example, it didn't work all that well with NFS.

Ext3 was first released as an official distribution default filesystem in 2001 with Red Hat 7.2. Red Hat opted for Ext3 as the filesystem looked set to make the official kernel. It actually made it into kernel 2.2.15 (the last 2.2 series kernel before 2.4). ReiserFS had to wait until kernel 2.4.1, but since then most Linux distributions have either used Ext3 or ReiserFS as their standard filesystems, and both have a reputation of being very stable.

In the same year, two Linux ports of journaling filesystems by IBM and SGI made it into the Linux kernel. XFS by SGI is now one of the best filesystems for files of 1MB and more. JFS, a filesystem developed by IBM, failed to establish a foothold in the industry and is no longer officially supported by many distros (Suse, for example).

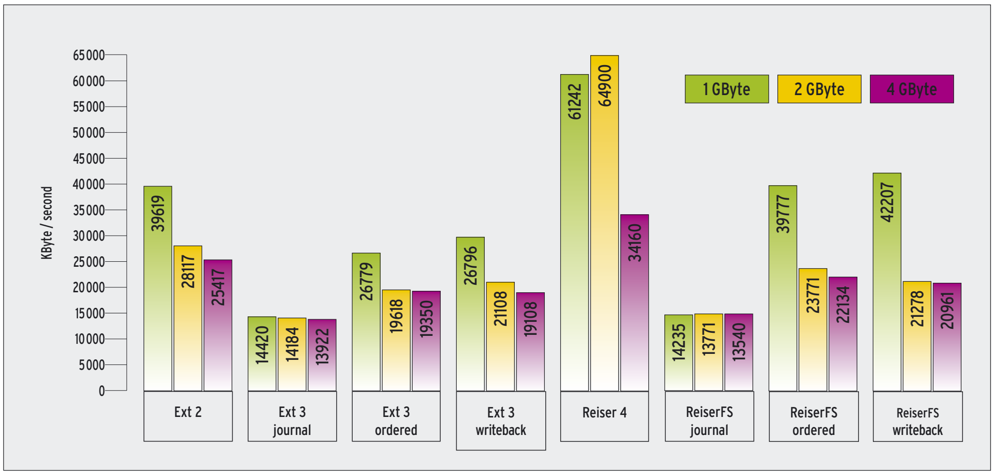

Figure 1: Write performance using the Iozone benchmark. In journaling mode, Ext3 and ReiserFS achieve similar low speeds. Writeback and Ordered modes boost data transfer speeds for Ext3 and ReiserFS by 40% and 120%, respectively.

on your computer, XFS should be your choice of filesystem.

Which Filesystem?

In most cases, you won't regret opting for Ext3 or ReiserFS. But there are a few scenarios where one or the other is preferable. ReiserFS takes a long time to mount partitions. The gap between ReiserFS and, say, Ext3 or XFS is negligible for 5 to 10 GB partitions. But if you decide to set up an 80GB partition, Ext3 will give you a multiple second speed advantage at boot time. This is also why ReiserFS is not recommended for external disks. If you connect a 200 GB hard disk with a ReiserFS filesystem to your computer, the system can take over ten seconds to mount the disk. The same procedure takes just three seconds if you format the disk with Ext3.

Small files and formatting are a big challenge for Ext3. For example, Ext3 takes over five minutes to create a 200 GB partition. If you tend to reformat your partitions quite often, you should opt for ReiserFS or XFS, which take just a few seconds to do the same job. If you typically work with office files of around 100KB, again ReiserFS is the right choice of filesystem, no matter what your distribution suggests. The filesystem by Namesys is much faster than Ext3 or XFS with files of this size. Avoid Ext3 for directories with many files, as the filesystem specifies the number of inodes during formating, and this restricts the number of files you can store on the partition. ReiserFS and XFS work with dynamic inode allocations: this completely removes the danger of having disk space left over, but no free inodes.

Ext3 and XFS are preferable if you use your PC to manage your audio or video collection. XFS is a lot quicker for gigabyte-plus file sizes than either Ext3 or ReiserFS. The space requirement is another factor you have to take into consideration. Thanks to its extremely efficient format, ReiserFS stores smaller files in far less space than either Ext3 or XFS. The kernel sources take up 250 MB of disk space on ReiserFS but 252 on XFS, and Ext3 requires 260 MB to store the exact same data.

As a final recommendation, make sure you optimize your hard disk usage. Most hard disks provide faster access to the last few cylinders of the disk than to the start of the disk. If this is true for your disk, you will want to create your root or data directories at the end of the disk.

Optimization Functions

Let's start with the good news: most distributions use a number of performance tweaks by default. However, there are a number of ReiserFS and Ext3 parameters that can make your filesystem just a tiny bit quicker. Linux filesystems store the last access time for each file. You can prevent this by stipulating the *noatime* mount option. Ext3, ReiserFS, and XFS all give you this option. There are no disadvantages to setting *noatime* on desktop systems. In a similar approach to *noatime* for files, *nodiratime* disables the last access time record for directories.

To try this option out for the partition */dev/hda6*, which you will be mounting in */test*, give the following command:

```
mount -o noatime,nodiratime ➥
/dev/hda6 /test
```

To add the option permanently, edit your */etc/fstab*. Add the *noatime,nodiratime* entries to the existing entries in column four.

In our lab, the option gave us a minor write performance boost on ReiserFS, although the test results were subject to some fluctuation. Using Suse Linux 10.0 OSS without both options, it took 110 seconds to copy the kernel sources from

Table 1: Ext3 and ReiserFS Data Options

Option	Explanation
data=journal	This option, which first copies all information to the journal area before storing the information at its final destination, guarantees maximum data safety. However, data throughput drops by about 50 percent with both Ext3 and ReiserFS as write operations take twice as long to complete.
data=ordered	This is the default option. The filesystem first writes the information at its final destination, and then creates a journal entry for the completed operation.
data=writeback	This mount option, which means a performance boost of about 10 percent on Ext3, and about 30 percent on ReiserFS compared to the default, allows the filesystem to create journal entries before the write operation has been completed. In case of a crash, old data discovered by a filesystem check may be reinstated in files. This option is only available with kernel 2.6 for Reiser.
data=notail	Only for ReiserFS. ReiserFS uses slack space in blocks to store excess data that will not fit into a block. The tail of the file is thus chopped off and stored in another block. This allows ReiserFS to store 10 to 20 percent more files on a partition of the same size in comparison with Ext3. As this causes a slight performance hit, the feature can be disabled by stipulating *data=notail.* The performance gain that this option achieves is less than 5 percent.

Table 2: Kernel Source Copy

Distribution	Filesystem	Time
Ubuntu 5.10	Ext3	70s
Ubuntu 5.10	ReiserFS	65s
Suse 10.0	ReiserFS	100s
SuSE 10.0	Ext3	70s
Suse 10.0	XFS	80s

partition A to partition B. When we enabled *noatime, nodiratime*, the same copy operation took just 100 seconds. Interestingly, Ext3 was far quicker in this test. It took just 70 seconds, no matter whether the *noatime* option was enabled or disabled. XFS took 80 seconds to copy the 20,000 plus files from one partition to another. Again, *noatime,nodiratime* did not noticeably influence the results. We repeated the tests using Ubuntu Linux 5.10 “Breezy Badger” and discovered that both Ext3 and ReiserFS are slightly quicker on Ubuntu than on Suse Linux 10.0.

Further tests revealed that using ReiserFS for the Suse Linux 10.0 root directory was slowing the whole system down. After reinstalling with Ext3 as the root filesystem, copying to the ReiserFS partition was quicker than the same process using Ext3 – as expected – achieving about the same speeds as on Ubuntu.

The *data = writeback* option is a different matter (see Table 1). The differences are easily measurable, representing a 10 percent gain with Ext3, and up to 30 percent with ReiserFS. However, it is not easy to set this option for Ext3 in the root directory – if you have Suse at least. To optimize */dev/hda7*, for example, you would need to run *tune2fs /dev/hda7 -o journal_data_writeback*.

The *dir_index* feature can improve the speed of larger Ext3 partitions. This option enables a special technique that accelerates searching in large directories. To tune an existing Ext3 partition in this way, you need the following two commands:

```
tune2fs -O dir_index /dev/hda7
fsck.ext3 -fD /dev/hda7
```

The filesystem check is not required when creating a partition. Just enter the following command:

```
mkfs.ext3 -O dir_index /dev/↵
hda7
```

Another approach to accelerating disks with Ext3, ReiserFS, or XFS is to swap the journal out onto a second disk (preferably not attached to the same IDE bus). To do this with Ext3, specify *-O journal_dev /dev/hdd1* when creating the partition (this assumes that */dev/hdd1* is the partition you will be using for the partition). The option for ReiserFS is *-j /dev/hdd1*. A kernel developer recently discovered a major bug in ReiserFS that causes the system to crash under heavy load. To avoid the bug, you might like to download the latest version of ReiserFS before you think about using an external journal. The option for XFS is *-l logdev = /dev/hdd1*.

ReiserFS has a few more mount options that can boost performance. As most of these options are highly experimental, you might prefer to move to Reiser4 straight away, if you are the adventurous type. Of course Reiser4 is partly experimental, but it is a lot quicker. On our lab machine, Reiser4 copied the test data in just 30 seconds.

Suse Linux 10.0 comes with Reiser4 packages. If you would like to install the new Reiser filesystem, first install *libaal* and *reiser4progs* using YaST, and then go on to format the target partition by entering *mkfs.reiser4 Partition*. There is a HOWTO for the install on Ubuntu on the Reiser4 mailing list [4]. The official HOWTO by Namesys is here [5].

Slow Down!

If speed is a secondary consideration, and you place more emphasis on data security, you can always slow down the filesystem. The *data = journal* tells Ext3 and ReiserFS to first write all data to the journal and then to the target space. As this means twice the amount of work, system performance is bound to take a hit. But the chance of finding all your data in one piece after a crash is much better. If your machine is prone to crashing, you might like to enable this mount option. To achieve fairly good performance despite using *data = journal*, you can always migrate the journal to a second disk.

Conclusions

Filesystem tuning takes a lot of time, and the results are often negligible. It makes much more sense to choose the right filesystem for your application during the install, than to attempt to tweak the filesystem later. Ext3 seems to be the better choice for Suse Linux 10.0: the system boots more quickly, and it mounts additional partitions faster. Apart from this, Ext3 and Reiser4 are pretty much even. XFS is a useful choice for video editing. Speed freaks will opt for Reiser4, but be aware that installing Reiser4 in the root directory is tricky. You might prefer to use a Linux distribution that supports Reiser4 as an installation option, such as Underground Linux which is discussed on page 36 in this issue. ■

THE AUTHOR

Marcel Hilzinger studied history at university. He has been working for Suse Linux’s Hungarian office in Budapest since 2001, where he translated the Suse documentation into Hungarian, among other things.

INFO

[1] Ext3: *http://www.zipworld.com.au/~akpm/linux/ext3/*

[2] ReiserFS: *http://www.namesys.com*

[3] XFS: *http://linux-xfs.sgi.com/projects/xfs/*

[4] Reiser4 for Ubuntu: *http://marc.theaimsgroup.com/?l=reiserfs&m=113270611302330&w=2*

[5] Reiser4 installation: *http://www.namesys.com/install_v4.html*