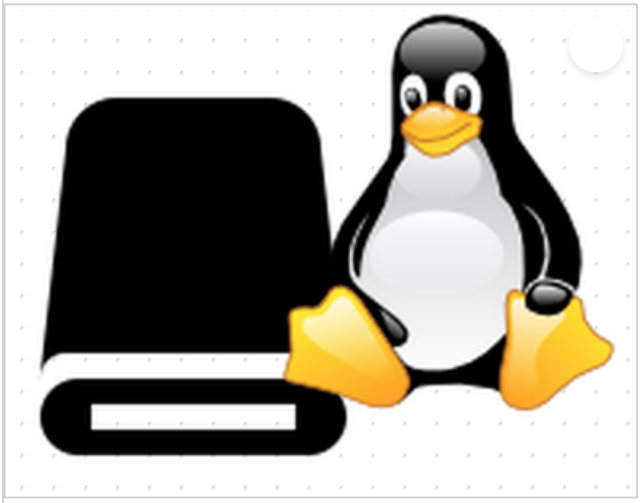
Understanding File System Superblock in Linux

Submitted by Sarath Pillai on Mon, 09/07/2015 - 19:40



Extended Filesystem being the default file system in Linux, we will be focusing ext file system in this article to understand superblocks.

Before we get to understanding Superblocks in a file system, let's understand some common terminologies and building blocks of a file system.

Blocks in File System

When a partition or disk is formatted, the sectors in the hardisk is first divided into small groups. This groups of sectors is called as blocks. The block size is something that can be specified when a user formats a partition using the command line parameters available.

```
1 mkfs -t ext3 -b 4096 /dev/sda1
```

In the above command we have specified block size while formatting /dev/sda1 partition. The size specified is in bytes. So basically one block will be of 4096 bytes.

Block Size for Ext2 can be 1Kb, 2Kb, 4Kb, 8Kb
Block Size for Ext3 can be 1Kb, 2Kb, 4Kb, 8Kb
Block Size for Ext4 can be 1Kb to 64Kb

The block size you select will impact the following things

- Maximum File Size
- Maximum File System Size
- Performance

Related: **Linux Read and Write Performance Test (http://www.slashroot.in/linux-file-system-read-write-performance-test)**

The reason block size has an impact on performance is because, the file system driver sends block size ranges to the underlying drive, while reading and writing things to file system. Just imagine if you have a large file, reading smaller blocks (which combined together makes the file size) one by one will take longer. So the basic idea is to keep bigger block size, if your intention is to store large files on the file system.

> *Less IOPS will be performed if you have larger block size for your file system.*

Related: **Monitoring IO in Linux (http://www.slashroot.in/linux-system-io-monitoring)**

And if you are willing to store smaller files on the file system, its better to go with smaller block size as it will save a lot of disk space and also from performance perspective.

Hard disk sector is nothing but a basic storage unit of the drive, which can be addressed. Most physical drives have a sector size of 512 bytes. Please keep the fact in mind that hard disk sector is the property of the drive. Block size of file systems (that we just discussed above) is a software construct, and is not to be confused with the hard disk sector.

A linux Kernal performs all its operations on a file system using block size of the file system. The main important thing to understand here is that the block size can never be smaller than the hard disk's sector size, and will always be in multiple of the hard disk sector size. The linux Kernel also requires the file system block size to be smaller or equal to the system page size.

Linux system page size can be checked by using the below command.

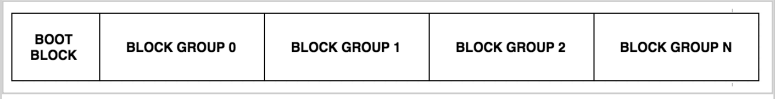
```
root@localhost:~# getconf PAGE_SIZE
4096
```

Block Groups in File System

The blocks that we discussed in the previous section are further grouped together to form block groups for ease of access during read and writes. This is primarily done to reduce the amount of time taken while reading or writing large amounts of data.

The ext file system divides the entire space of the partition to equal sized block groups(these block groups are arranged one after the other in a sequential manner).

A typical partition layout looks something like the below at a very high level.



Number of blocks per group is fixed, and cannot be changed. Generally the number of blocks per block

*groups is 8*block size.*

Lets see an output of mke2fs command, that displays few of the information that we discussed till now.

```
root@localhost:~# mke2fs /dev/xvdf
mke2fs 1.42.9 (4-Feb-2014)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
Stride=0 blocks, Stripe width=0 blocks
6553600 inodes, 26214400 blocks
1310720 blocks (5.00%) reserved for the super user
First data block=0
Maximum filesystem blocks=4294967296
800 block groups
32768 blocks per group, 32768 fragments per group
8192 inodes per group
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376, 294912, 819200, 884736
    4096000, 7962624, 11239424, 20480000, 23887872

Allocating group tables: done
Writing inode tables: done
Writing superblocks and filesystem accounting informatio
```

If you see the above output, it gives you the below details:

- Block Size (4096 bytes)
- 800 Block Groups
- 32768 Blocks per group (which is 8*4096, as mentioned earlier)

It also shows the superblock backup locations on the partition. That is the block groups where superblock backups are stored.

What is File System Superblock?

The most simplest definition of Superblock is that, its the metadata of the file system. Similar to how i-nodes stores metadata of files, Superblocks store metadata of the file system. As it stores critical information about the file system, preventing corruption of superblocks is of utmost importance.

If the superblock of a file system is corrupted, then you will face issues while mounting that file system. The

system verifies and modifies superblock each time you mount the file system.

Superblocks also stores configuration of the file system. Some higher level details that is stored in superblock is mentioned below.

- Blocks in the file system
- No of free blocks in the file system
- Inodes per block group
- Blocks per block group
- No of times the file system was mounted since last fsck.
- Mount time
- UUID of the file system
- Write time
- File System State (ie: was it cleanly unounted, errors detected etc)
- The file system type etc(ie: whether its ext2,3 or 4).
- The operating system in which the file system was formatted

The primary copy of superblock is stored in the very first block group. This is called primary superblock, because this is the superblock that is read by the system when you mount the file system. As block groups are counted from 0, we can say that the primary superblock is stored at the beginning of block group 0.

As superblock is a very critical component of the file system, a backup redundant copy is placed at each "block group".

In other words, every "block group" in the file system will have the backup superblock. This is basically done to recover the superblock if the primary one gets corrupted.

You can easily imagine that storing backup copies of superblock in every "block group", can consume a considerable amount of file system storage space. Due to this very reason, later versions implemented a feature called **"sparse_super"** which basically stores backup superblocks only on block groups 0, 1 and powers of 3,5,7. This option is by default enabled in latest system's, due to which you will see backup copies of superblock only on

several block groups(which is evident from the mke2fs output shown in the previous section).

How to view Superblock Information of a File System?

You can view superblock information of an existing file system using dumpe2fs command as shown below.

```
root@localhost:~# dumpe2fs -h /dev/xvda1
dumpe2fs 1.42.9 (4-Feb-2014)
Filesystem volume name:   cloudimg-rootfs
Last mounted on:          /
Filesystem UUID:          f75f9307-27dc-4af8-87b7-f414c0
Filesystem magic number:  0xEF53
Filesystem revision #:    1 (dynamic)
Filesystem features:      has_journal ext_attr resize_in
Filesystem flags:         signed_directory_hash
Default mount options:    (none)
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              6553600
Block count:              26212055
Reserved block count:     1069295
Free blocks:              20083290
Free inodes:              6470905
First block:              0
Block size:               4096
Fragment size:            4096
Reserved GDT blocks:      505
Blocks per group:         32768
Fragments per group:      32768
Inodes per group:         8192
Inode blocks per group:   512
Flex block group size:    16
Filesystem created:       Sat Sep 27 13:05:57 2014
Last mount time:          Mon Feb  2 14:43:31 2015
Last write time:          Sat Sep 27 13:06:55 2014
Mount count:              4
Maximum mount count:      20
Last checked:             Sat Sep 27 13:05:57 2014
Check interval:           15552000 (6 months)
Next check after:         Thu Mar 26 13:05:57 2015
Lifetime writes:          305 GB
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
First inode:              11
Inode size:              256
Required extra isize:     28
Desired extra isize:      28
Journal inode:            8
First orphan inode:       396056
Default directory hash:   half_md4
Directory Hash Seed:      2124542b-ea2f-4552-afaa-c57202
Journal backup:           inode blocks
Journal features:         journal_incompat_revoke
Journal size:             128M
Journal length:           32768
Journal sequence:         0x0151d29d
Journal start:            11415<span style="font-size:16
</span>
```

You can also view the exact locations of superblock and backups using the same dumpe2fs command as shown below.

```
root@localhost:~# dumpe2fs /dev/xvda1 | grep -i superblo
dumpe2fs 1.42.9 (4-Feb-2014)
  Primary superblock at 0, Group descriptors at 1-7
```

```
    Backup superblock at 32768, Group descriptors at 32769
    Backup superblock at 98304, Group descriptors at 98305
    Backup superblock at 163840, Group descriptors at 1638
    Backup superblock at 229376, Group descriptors at 2293
    Backup superblock at 294912, Group descriptors at 2949
    Backup superblock at 819200, Group descriptors at 8192
    Backup superblock at 884736, Group descriptors at 8847
    Backup superblock at 1605632, Group descriptors at 160
    Backup superblock at 2654208, Group descriptors at 265
    Backup superblock at 4096000, Group descriptors at 409
    Backup superblock at 7962624, Group descriptors at 796
    Backup superblock at 11239424, Group descriptors at 11
    Backup superblock at 20480000, Group descriptors at 20
    Backup superblock at 23887872, Group descriptors at 23
```

How can I use backup superblocks to recover a corrupted file system?

The first thing to do is to do a file system check using fsck utility. This is as simple as running fsck command against your required file system as shown below.

```
root@localhost:~# fsck.ext3 -v /dev/xvda1
```

If fsck output shows superblock read errors, you can do the below to fix this problem.

First step is to Identify where the backup superblocks are located. This can be done by the earlier shown method of using dumpe2fs command OR using the below command also you can find the backup superblock locations.

```
root@localhost:~# mke2fs -n /dev/xvda1
```

-n option used with mke2fs in the above example, will show the backup superblock locations, without creating an file system. Read mke2fs man page for more information on this command line switch.

Second step is to simply restore the backup copy of superblock using e2fsck command as shown below.

```
root@localhost:~# e2fsck -b 32768 /dev/xvda1
```

In the above shown example the number 32768 i have used is the location of the first backup copy of the superblock. Once the above command succeeds, you can retry mounting the file system.

> *Alternatively you can also use **sb** option available in **mount** command. **sb** option lets you specify the superblock to use while mounting the file system. As mentioned earlier in the article, when you mount a file system, by default the primary superblock is read. Instead you can force mount command to read a backup superblock in case the primary one is corrupted. Below shown is an example mount command using a backup superblock to mount a file system.*

```
root@localhost:~# mount -o -sb=98304 /dev/xvda1 /data
```

The above shown mount command will use backup superblock located at block 98304 while mounting.

Rate this article:

Average: 3.8 (558 votes)

Add new comment (/understanding-file-system-superblock-linux#comment-form)

Comments

Excellent (/comment/1214#comment-1214)

Permalink (/comment/1214#comment-1214)
Submitted by Rishi Kapoor on Tue, 10/20/2015 -

19:55

explained very well. Excellent.

reply (/comment/reply/254/1214)

linux system admin (/comment/2251#comment-2251)

Permalink (/comment/2251#comment-2251)
Submitted by srinivas karne on Mon, 05/02/2016 - 10:49

Thanks very usefull it...
I didn't answer when I was facing the interview

reply (/comment/reply/254/2251)

Trying to mount the filesystem with backup superblock (/comment/2347#comment-2347)

Permalink (/comment/2347#comment-2347)
Submitted by Bhasker Reddy on Wed, 08/17/2016 - 12:00

[root@server (mailto:root@server) ~]# fsck.ext3 -v /dev/sda8
e2fsck 1.41.12 (17-May-2010)
/dev/sda8: clean, 17/197600 files, 29971/789193 blocks

[root@server (mailto:root@server) ~]# dumpe2fs /dev/sda8 | grep -i super
dumpe2fs 1.41.12 (17-May-2010)
Filesystem features: has_journal ext_attr resize_inode dir_index filetype sparse_super large_file
Primary superblock at 0, Group descriptors at 1-1
Backup superblock at 32768, Group descriptors at 32769-32769
Backup superblock at 98304, Group descriptors at 98305-98305
Backup superblock at 163840, Group descriptors at 163841-163841
Backup superblock at 229376, Group descriptors at 229377-229377
Backup superblock at 294912, Group descriptors at 294913-294913

[root@server (mailto:root@server) ~]# mount -o sb=98304 /dev/sda8 /volume1
mount: wrong fs type, bad option, bad superblock on /dev/sda8,
missing codepage or helper program, or other error

In some cases useful info is found in syslog - try dmesg | tail or so

After running the dmesg command, found the following error.

EXT3-fs (sda8): error: can't find ext3 filesystem on dev sda8.

reply (/comment/reply/254/2347)

mount filesystem (/comment/2374#comment-2374)

Permalink (/comment/2374#comment-2374)
Submitted by gagan on Thu, 09/22/2016 - 17:12

@bhasker there will be 2 conditions the first one probably you should use mount -o -sb=98304 /dev/sda8

The second one may be that device is busy I mean may be some process is using that device so you kill that process...
lsof |grep sda8 find the pid of that process and kill it..i have faced that issue it worked for me..

reply (/comment/reply/254/2374)

Trying to mount the filesystem with backup superblock (/comment/3060#comment-3060)

Permalink (/comment/3060#comment-3060)
Submitted by Karthik on Fri, 04/20/2018 - 02:31

Please check your using small "o" or Capital "O". Seems your using small o(option) instant of capital one. :)

reply (/comment/reply/254/3060)

Add new comment

Your name

Subject

Comment *

- No HTML tags allowed.
- Web page addresses and e-mail addresses turn into links automatically.
- Lines and paragraphs break automatically.

More information about text formats (/filter/tips)

CAPTCHA

This question is for testing whether or not you are a human visitor and to prevent automated spam submissions.

I'm not a robot

reCAPTCHA
Privacy - Terms

Save Preview

Search Articles ...

Today's Most Popular

(/difference-between-iterative-and-recursive-dns-query)

difference between iterative and recursive dns query (/difference-between-iterative-and-recursive-dns-query)

Archives (/taxonomy/term/2) - 44 comment(s) (https://www.slashroot.in/difference-between-iterative-and-recursive-dns-query#comments#comments)

(/how-does-traceroute-work-and-examples-using-traceroute-command)

How Does Traceroute Work and Example's of using traceroute command (/how-does-traceroute-work-and-examples-using-traceroute-command)

Networking (/networking) - 75 comment(s) (https://www.slashroot.in/how-does-traceroute-work-and-examples-using-traceroute-command#comments#comments)


(/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

SAN vs NAS - Difference between a Storage Area Network and Network Attached Storage (/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

Archives (/taxonomy/term/2) - 57 comment(s) (https://www.slashroot.in/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage#comments#comments)


(/what-ping-sweep-and-how-do-ping-sweep)

What is ping sweep and how to do a ping sweep (/what-ping-sweep-and-how-do-ping-sweep)

Security Notes (/security-notes) - 2 comment(s) (https://www.slashroot.in/what-ping-sweep-and-how-do-ping-sweep#comments#comments)

Most Commented


(/how-does-traceroute-work-and-examples-using-traceroute-command)

How Does Traceroute Work and Example's of using traceroute command (/how-does-traceroute-work-and-examples-using-traceroute-command)

Networking (/networking) - 75 comment(s) (https://www.slashroot.in/how-does-traceroute-work-and-examples-using-traceroute-command#comments#comments)


(/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

SAN vs NAS - Difference between a Storage Area Network

Top Rated Articles


(/what-uuid-used-fstab)

What is UUID used in fstab (/what-uuid-used-fstab)

Average: 4.6 (168 votes)


(/dns-root-servers-most-critical-infrastructure-internet)

DNS Root Servers: The most critical infrastructure on the internet (/dns-root-servers-most-critical-infrastructure-internet)

Average: 4.6 (721 votes)

and Network Attached Storage (/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage)

Archives (/taxonomy/term/2) - 57 comment(s) (https://www.slashroot.in/san-vs-nas-difference-between-storage-area-network-and-network-attached-storage#comments#comments)

(/difference-between-iterative-and-recursive-dns-query)

difference between iterative and recursive dns query (/difference-between-iterative-and-recursive-dns-query)

Archives (/taxonomy/term/2) - 44 comment(s) (https://www.slashroot.in/difference-between-iterative-and-recursive-dns-query#comments#comments)

(/linux-booting-process-step-step-tutorial-understanding-linux-boot-sequence)

Linux Booting Process: A step by step tutorial for understanding Linux boot sequence (/linux-booting-process-step-step-tutorial-understanding-linux-boot-sequence)

Archives (/taxonomy/term/2) - 41 comment(s) (https://www.slashroot.in/linux-booting-process-step-step-tutorial-understanding-linux-boot-sequence#comments#comments)

(/ssh-port-forwarding-linux-configuration-and-examples)

SSH Port Forwarding in Linux: Configuration and Examples (/ssh-port-forwarding-linux-configuration-and-examples)

Average: 4.5 (134 votes)

(/alert-message-protocol-ssl)

Alert Message Protocol in SSL (/alert-message-protocol-ssl)

Average: 4.5 (95 votes)

(/puppet-tutorial-configuring-puppet-agent)

Puppet Tutorial: Configuring Puppet Agent (/puppet-tutorial-configuring-puppet-agent)

Average: 4.5 (75 votes)

(/proc-file-system-linux-explained)

PROC file System in Linux explained (/proc-file-system-linux-explained)

Average: 4.5 (353 votes)

(/lsof-command-usage-and-examples)

lsof Command Usage and Examples (/lsof-command-usage-and-examples)

Average: 4.4 (354 votes)

(/different-methods-executing-bash-shell-script-and-their-differences)

Different Methods of Executing A Bash Shell Script And Their Differences (/different-

methods-executing-bash-shell-script-and-their-differences)

Average: 4.4 (21 votes)

Jump back to navigation (#page)

Get in touch with The Authors

Sarath Pillai

Ph: +917303074400

Email: sarath@slashroot.in (mailto:sarath@slashroot.in)

Satish Tiwary

Ph: +919509452488

Email: satish@slashroot.in (mailto:satish@slashroot.in)

Follow Us

Subscribe to our RSS Feed (rss.xml)

Follow us on Twitter (https://twitter.com/slashrootin)

Be a fan on Facebook (http://www.facebook.com/pages/Slashrootin-A-technical-Blog/101642306662299)

Recent Posts

(/how-delete-git-branch-locally-and-remotely)

How To Delete A Git Branch Locally And Remotely (/how-delete-git-branch-locally-and-remotely)

Archives (/taxonomy/term/2) - *6 years 1 week* ago

(/difference-between-bashrc-and-bashprofile)

Difference Between .bashrc and .bash_profile (/difference-between-bashrc-and-bashprofile)

Archives (/taxonomy/term/2) - *6 years 2 weeks* ago

(/difference-between-process-and-thread-linux)

Difference Between Process And Thread in Linux (/difference-between-process-and-thread-linux)

Last Viewed

Needham-Schroeder protocol Explained (/needham-schroeder-protocol-explained)

Security Notes (/security-notes) - last view *47 sec* ago

(/what-dns-zone-file-complete-tutorial-zone-file-and-its-contents)

What is a DNS ZONE file: A Complete Tutorial on zone file and its contents (/what-dns-zone-file-complete-tutorial-zone-file-and-its-contents)

Archives (/taxonomy/term/2) - last view *3 min 44 sec* ago

(/examples-using-sar-command-system-monitoring-linux)

Examples of using SAR command for system monitoring in Linux (/examples-using-sar-command-system-monitoring-linux)

Archives (/taxonomy/term/2) - last view *4 min 18 sec* ago

Archives (/taxonomy/term/2) - *6 years 4 weeks* ago

(/how-does-ssltls-chain-certificates-and-its-validation-work)

How Does SSL/TLS Chain Certificates and Its Validation work? (/how-does-ssltls-chain-certificates-and-its-validation-work)

Security Notes (/security-notes) - *6 years 1 month* ago

(/linux-processes-administration-and-monitoring-tutorial)

Linux Processes:Administration and Monitoring tutorial (/linux-processes-administration-and-monitoring-tutorial)

Archives (/taxonomy/term/2) - last view *4 min 21 sec* ago