Disk Safe Size Larger Than Expected on Large ext4 Filesystems with metadata_csum Enabled

KB: Reported Disk Safe Size Larger Than Expected on Large ext4 Filesystems with metadata_csum Enabled

Summary

On some Linux systems using the ext4 filesystem, the reported disk safe size may be significantly larger than expected. This behavior is often observed on large volumes where the ext4 metadata_csum feature is enabled.

The metadata_csum feature stores checksums for filesystem metadata structures to improve filesystem integrity and corruption detection. While the overhead is typically small, the cumulative metadata allocation can become noticeable on very large filesystems, resulting in a larger reported disk safe size than expected.

Symptoms

- Reported disk safe size is significantly larger than expected.
- The affected filesystem is formatted with ext4.
- The filesystem has the metadata_csum feature enabled.

Cause

The ext4 metadata_csum feature provides metadata integrity protection by storing checksums for critical filesystem metadata structures, including:

- Group descriptors
- Inodes
- Directories
- Allocation bitmaps
- Journal metadata

Unlike older checksum implementations, metadata_csum introduces checksum coverage across a broader set of filesystem structures. As a result, filesystem blocks that may appear unused can contain checksum-related metadata. On very large filesystems, the aggregate space consumed by these metadata structures can become significant and affect safe-size calculations.

This feature is enabled by default on many modern Linux distributions, including AlmaLinux 8 and later releases utilizing newer versions of e2fsprogs.

Resolution

Step 1: Verify Whether metadata_csum Is Enabled

Run the following command:

```
tune2fs -l /dev/sda | grep features
```

Example output:

```
Filesystem features: has_journal ext_attr resize_inode dir_index filetype needs_recovery extent 64bit flex_bg sparse_super large_file huge_file dir_nl
```

If metadata_csum appears in the feature list, continue with the following steps.

Step 2: Validate Filesystem Integrity

Before modifying filesystem features, ensure the filesystem successfully passes a consistency check:

```
fsck -f /dev/sda
```

Correct any reported issues before proceeding.

Step 3: Disable metadata_csum

The filesystem must be unmounted before modifying ext4 feature flags.

Disable the feature using:

```
tune2fs -O ^metadata_csum /dev/path/to/disk
```

Depending on the filesystem configuration and e2fsprogs version, the filesystem may need to be converted back to the older group descriptor checksum format:

```
tune2fs -O uninit_bg,^metadata_csum /dev/path/to/disk
```

After changing filesystem features, run:

```
e2fsck -f /dev/path/to/disk
```

Important

- This operation cannot be performed on a mounted filesystem.
- If the affected volume is the root (/) filesystem, boot from rescue media or a LiveCD environment to perform the change while the filesystem is offline.
- Ensure a valid backup exists before making filesystem-level changes.

Reference:
Ext4 Metadata Checksums

Step 4: Verify the Change

Verify that metadata_csum is no longer present:

```
tune2fs -l /dev/sda | grep features
```

Example output:

```
Filesystem features: has_journal ext_attr resize_inode dir_index filetype needs_recovery extent 64bit flex_bg sparse_super large_file huge_file dir_nl
```

Risks and Considerations When Disabling metadata_csum

Important
Disabling metadata_csum reduces the filesystem's ability to detect metadata corruption and should only be performed when the operational benefits outweigh the integrity protections provided by the feature.

Reduced Metadata Corruption Detection

The metadata_csum feature validates metadata structures through checksums. This allows ext4 to detect corruption caused by:

- Disk errors
- Storage controller issues
- Firmware defects
- Power loss events
- Unexpected system crashes

After disabling the feature, this additional checksum validation is no longer available.

Reversion to Legacy Protection Mechanisms

When metadata_csum is disabled, ext4 typically falls back to older protection mechanisms, including:

- gdt_csum (Group Descriptor Table Checksums)
- uninit_bg (Uninitialized Block Groups)

While these features provide a degree of metadata validation and filesystem optimization, they do not provide the same level of metadata coverage as metadata_csum.

Increased Risk of Undetected Metadata Corruption

Without metadata checksums, some forms of metadata corruption may not be detected immediately. Corruption may only be discovered during:

- Scheduled filesystem checks
- Manual fsck execution
- Data access operations

This can increase recovery complexity in environments where storage reliability is critical.

Compatibility Considerations

Disabling metadata_csum may improve compatibility with:

- Older Linux distributions
- Backup and Recovery tools
- Workflows sensitive to metadata overhead calculations

However, modern Linux systems generally benefit from keeping metadata_csum enabled unless a documented issue exists.

Recommendation

Disable metadata_csum only when:

- The increased safe-size calculation is causing a validated operational issue.
- A current backup of the filesystem exists.
- Filesystem integrity has been verified using fsck.
- The loss of enhanced metadata protection is considered acceptable for the environment.