Silicon Graphics, Inc.

XFS Overview & Internals
04 - mkfs

November 2006

Creating XFS Filesystems

- mkfs.xfs supports a large number of options for configuration a large number of different XFS filesystems
  - See mkfs.xfs(8)

- Units
  - `100s = 100 sectors = 100 x 512 bytes*`
  - `100b = 100 blocks = 100 x 4 kilobytes*`
  - `100k = 100 * 1024 bytes`
  - `Assuming 512 bytes sectors and 4 KB filesystem block`

- -N option can be used to show filesystem parameters without creating a filesystem
- Also provides the capability to pre-initialise the filesystem with directories and inodes, which is useful for testing

mkfs – Stripe Alignment

- Aligning file data on stripe width boundaries can significantly improve performance on large RAIDs
  - A 2MB write to filesystem with a 2MB stripe width and 512KB stripe unit will result in four I/Os, one to each lun. Without alignment this would often require two I/Os to one disk, and one I/O to the other three, taking longer than it should.

- To create a filesystem with a stripe unit of 1MB for an 8+1 RAID you would use:

```
mkfs -d sunit=2048,swidth=16384 device
```

or

```
mkfs -d su=1m,sw=8m device
```

mkfs – Inode Options

- The mkfs inode options allow the user to change
  - The size of the inode from the default 256 bytes to up 2048 bytes
  - The maximum number of inodes in the filesystem, defaults to 25%
  - Extended attribute allocation policy

- These options may impact
  - where the inode is allocated in filesystem
  - how much data needs to be read from disk to read the inode
  - how much data may be associated with the inode before data goes out of line

- To allocate 512 bytes to each inode:

```
mkfs -i size=512 device
```

mkfs – Directory Options

- TODO

mkfs – Sector Size

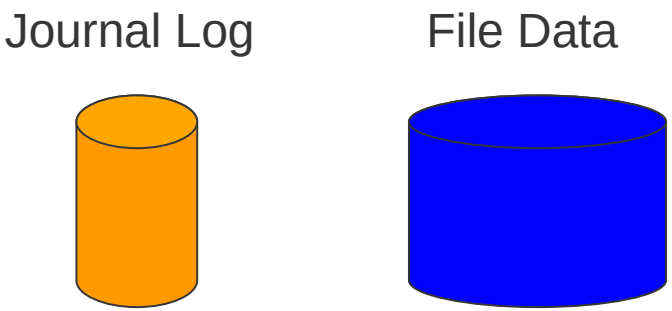
- TODO

mkfs – External Log

- The journal log can be on a different device to the rest of the filesystem
  - At least 512 blocks.
  - No more than 64K blocks or 128MB, whichever is smaller
  - Defaults to maximum size for >1TB filesystems
- Log device could be a device with better IOPS performance
  - 15K RPM disk or battery-backed memory

`mkfs.xfs -l logdev=`*`log_device device`*

`mount -o logdev=`*`log_device device path`*



mkfs – Log Alignment

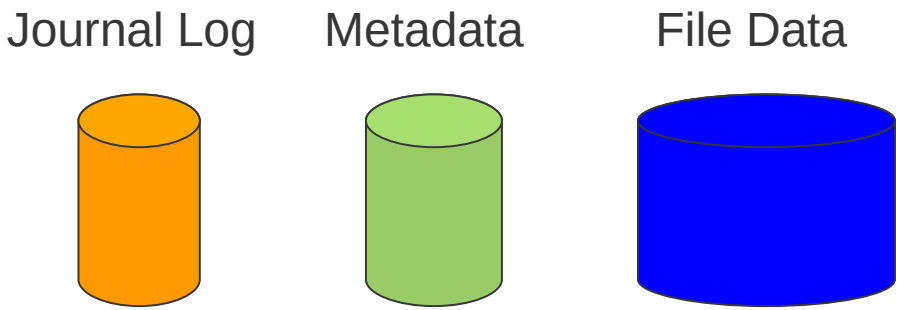
- TODO

mkfs - Realtime

- Originally designed for media streaming requiring predictable latencies
  - Realtime volume tuned for large files without small inode clusters
- Metadata and file data on separate volumes
  - Bitmap allocator only has to manage file data allocations

```
mkfs.xfs -l logdev=log_device -r rtdev=rt_device device
mount -o logdev=log_device,rtdev=rt_device device path
```

- *rt_device* is the device for the file data, *device* is for the metadata
- Receives limited testing and support in Linux



mkfs – Filesystem Image

- mkfs allows you to create a filesystem as a regular file
- This can be used to create a filesystem on a loop-back device

```
mkfs -d file=1,name=filename,size=2g filename
```

mkfs – Prototype Filesystem

- TODO