

Re: libata in 2.4.24?

Source: <http://linux.derkeiler.com/Mailing-Lists/Kernel/2003-12/0355.html>

From: Jeff Garzik (jgarzik_at_pobox.com)

Date: 12/02/03

Date: Tue, 2 Dec 2003 14:06:46 -0500

To: Greg Stark <gsstark@mit.edu>

On Tue, Dec 02, 2003 at 01:51:17PM -0500, Greg Stark wrote:

>

> *Jeff Garzik <jgarzik@pobox.com> writes:*

>

> > *If true, this is an IDE driver bug... assuming the drive itself*

> > *doesn't lie about FLUSH CACHE results (a few do).*

>

> *I don't think the IDE drivers issue FLUSH CACHE after every write on O_SYNC,*

> *or after fsync calls. The "lying" discussed on the database lists is when a*

> *normal write is issued, IDE disks report immediate success even before the*

> *write hits disk. As far as I know from the lists it seems *all* IDE disks*

> *behave this way unless write caching is disabled.*

The way CONFIG_IDE (the traditional IDE driver) and libata work right now, when the drive indicates that the read/write is complete, the OS driver indicates to the filesystem that the data transaction is complete.

So, today, no acknowledgement occurs until the data *_really_* is in the drive's buffers.

That said, "the database lists" may be seeing page cache effects. write(2) will certainly report success long before the data transaction is even sent to the driver! You must fsync(2) to flush data from the page cache to the IDE driver.

> *This doesn't happen with SCSI disks where multiple requests can be pending so*
> *there's no urgency to reporting a false success. The request doesn't complete*
> *until the write hits disk. As a result SCSI disks are reliable for database*
> *operation and IDE disks aren't unless write caching is disabled.*

This is not really true.

Regardless of TCQ, if the OS driver has not issued a FLUSH CACHE (IDE) or SYNCHRONIZE CACHE (SCSI), then the data is not guaranteed to be on the disk media. Plain and simple.

Linux-Kernel: Re: libata in 2.4.24?

If fsync(2) returns without a flush-cache, then your data is not guaranteed to be on the disk. And as you noted, flush-cache destroys performance.

> *I'm unclear on which of your #2 or #3 will be the solution though. Do either
> or both of them require that writes actually hit disk before the drive reports
> success? Do either of them allow that semantic without destroying concurrent
> performance?*

There are three levels:

- a) Data is successfully transferred to the controller/drive queue (TCQ).
- b) Data is successfully transferred to the drive's internal buffers.
- c) The drive successfully transfers data to the media.

Acknowledgement of (a) is basically instantaneous. The OS driver simply adds a drive read/write command to a list that the host controller can see.

Acknowledgement of (b) happens fairly rapidly, limited by the device's throughput and seek times, internal buffer load (amount of work todo), and internal algorithms.

Acknowledgement of (c) *_never_* occurs. One must issue the flush-cache drive command to be certain that the drive has flushed its write buffers.

Jeff

—

To unsubscribe from this list: send the line "unsubscribe linux-kernel" in the body of a message to majordomo@vger.kernel.org

More majordomo info at <http://vger.kernel.org/majordomo-info.html>

Please read the FAQ at <http://www.tux.org/lkml/>