dm-flakey

This target is the same as the linear target except that it exhibits unreliable behaviour periodically. It's been found useful in simulating failing devices for testing purposes.

Starting from the time the table is loaded, the device is available for <up interval> seconds, then exhibits unreliable behaviour for <down interval> seconds, and then this cycle repeats.

Also, consider using this in combination with the dm-delay target too, which can delay reads and writes and/or send them to different underlying devices.

Table parameters

```
<dev path> <offset> <up interval> <down interval> \
  [<num_features> [<feature arguments>]]
```

Mandatory parameters:

<dev path>:
: Full pathname to the underlying block-device, or a "major:minor" device-number.

<offset>:
: Starting sector within the device.

<up interval>:
: Number of seconds device is available.

<down interval>:
: Number of seconds device returns errors.

Optional feature parameters:

If no feature parameters are present, during the periods of unreliability, all I/O returns errors.

drop_writes:
: All write I/O is silently ignored. Read I/O is handled correctly.

error_writes:
: All write I/O is failed with an error signalled. Read I/O is handled correctly.

corrupt_bio_byte <Nth_byte> <direction> <value> <flags>:
: During <down interval>, replace <Nth_byte> of the data of each matching bio with <value>.

<Nth_byte>:
: The offset of the byte to replace. Counting starts at 1, to replace the first byte.

<direction>:
: Either 'r' to corrupt reads or 'w' to corrupt writes. 'w' is incompatible with drop_writes.

<value>:
: The value (from 0-255) to write.

<flags>:
: Perform the replacement only if bio->bi_opf has all the selected flags set.

Examples:

Replaces the 32nd byte of READ bios with the value 1:

```
corrupt_bio_byte 32 r 1 0
```

Replaces the 224th byte of REQ_META (=32) bios with the value 0:

```
corrupt_bio_byte 224 w 0 32
```