What is dm-lc

Akira Hayakawa (@akiradeveloper)
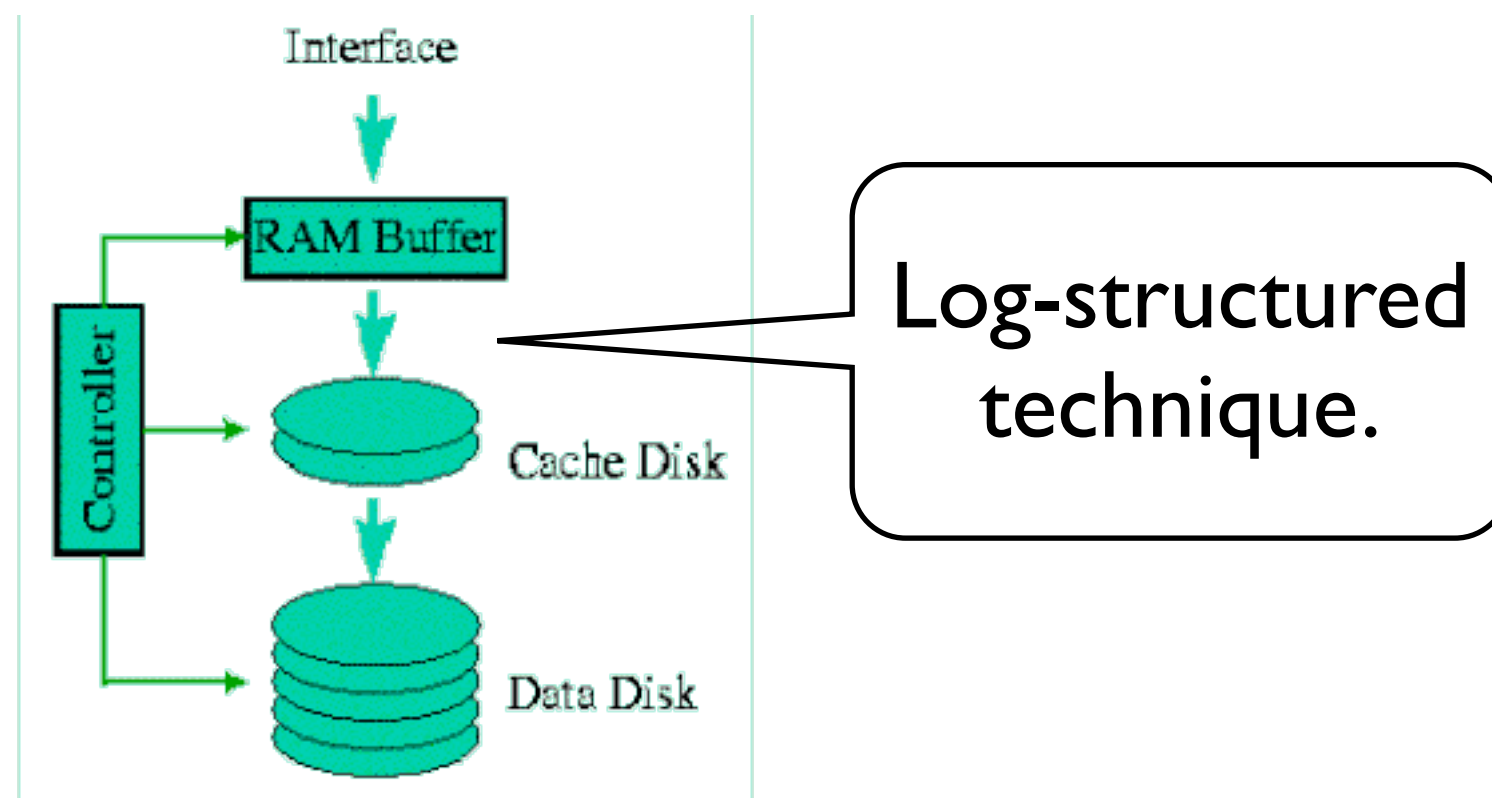
Development is real fun!

dm-lc?

- Cache module for Linux, based on Disk Caching Disk (DCD).

- caches ONLY writes.
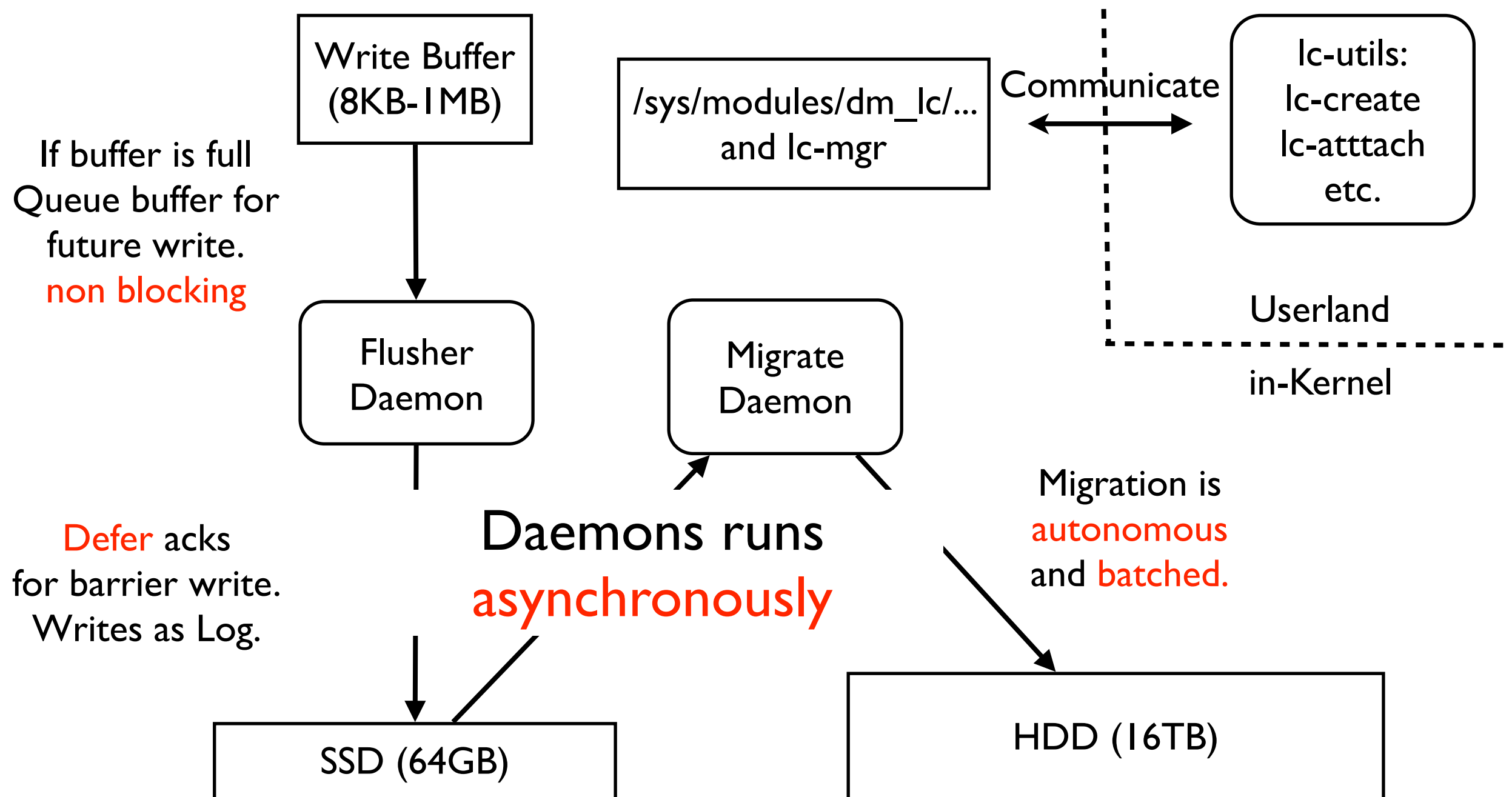
- Small SSD cache can boost random writes of backing HDD.

DCD?

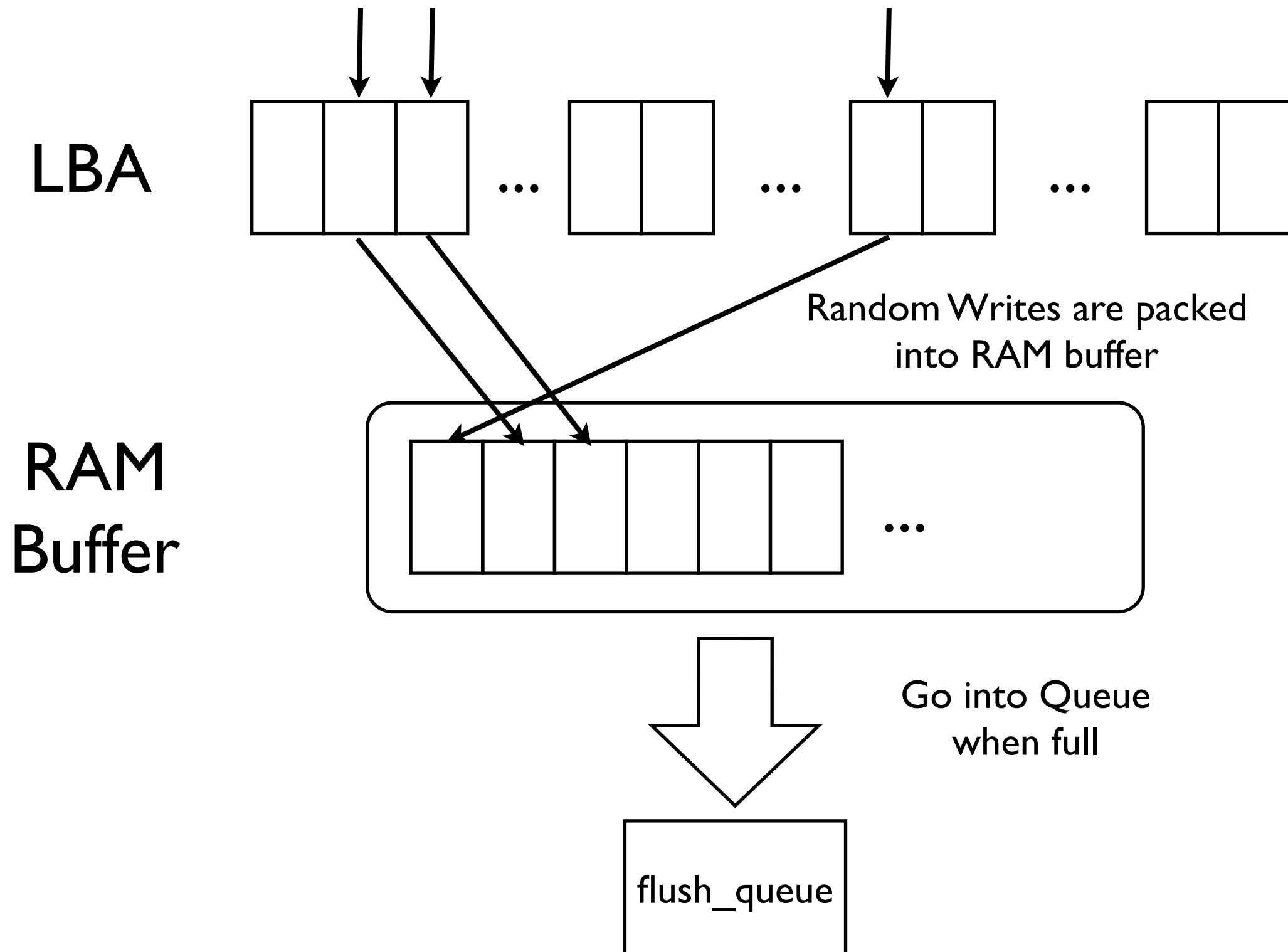- Log-structured filesystem in block layer.

Log-structured technique.

http://www.ele.uri.edu/research/hpcl/DCD/DCD.html

Architecture

Write Buffer (8KB-1MB)

/sys/modules/dm_lc/... and lc-mgr

Communicate

lc-utils: lc-create lc-atttach etc.

If buffer is full Queue buffer for future write. non blocking

Flusher Daemon

Migrate Daemon

Userland

in-Kernel

Defer acks for barrier write. Writes as Log.

Daemons runs asynchronously

Migration is autonomous and batched.

SSD (64GB)

HDD (16TB)

RAM Buffer

LBA

Random Writes are packed
into RAM buffer

RAM
Buffer

...

Go into Queue
when full

flush_queue

Queue and Flush

RAM

flush_queue -- -- flush_context -- -- flush_context

"Some flushing tasks are queued. Let's flush them."

Flusher Daemon

Log id k+3

Log id k+4

SSD

| Log id k | Log id k+1 | Log id k+2 | Log id k+3 | Log id k+4 |

Log Format

4KB

| segment_header | Data[0] | Data[1] | Data[2] | Data[3] | ... |

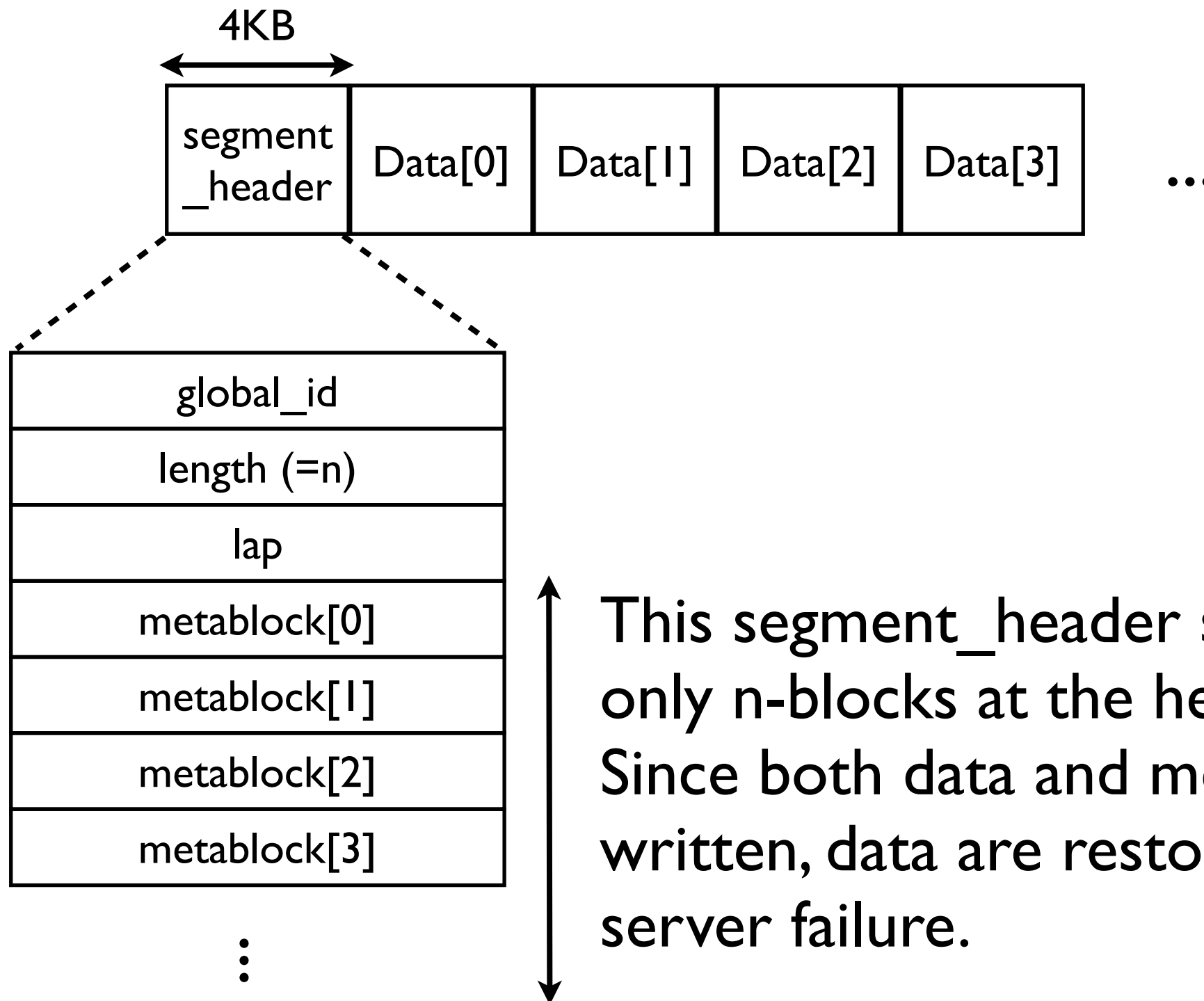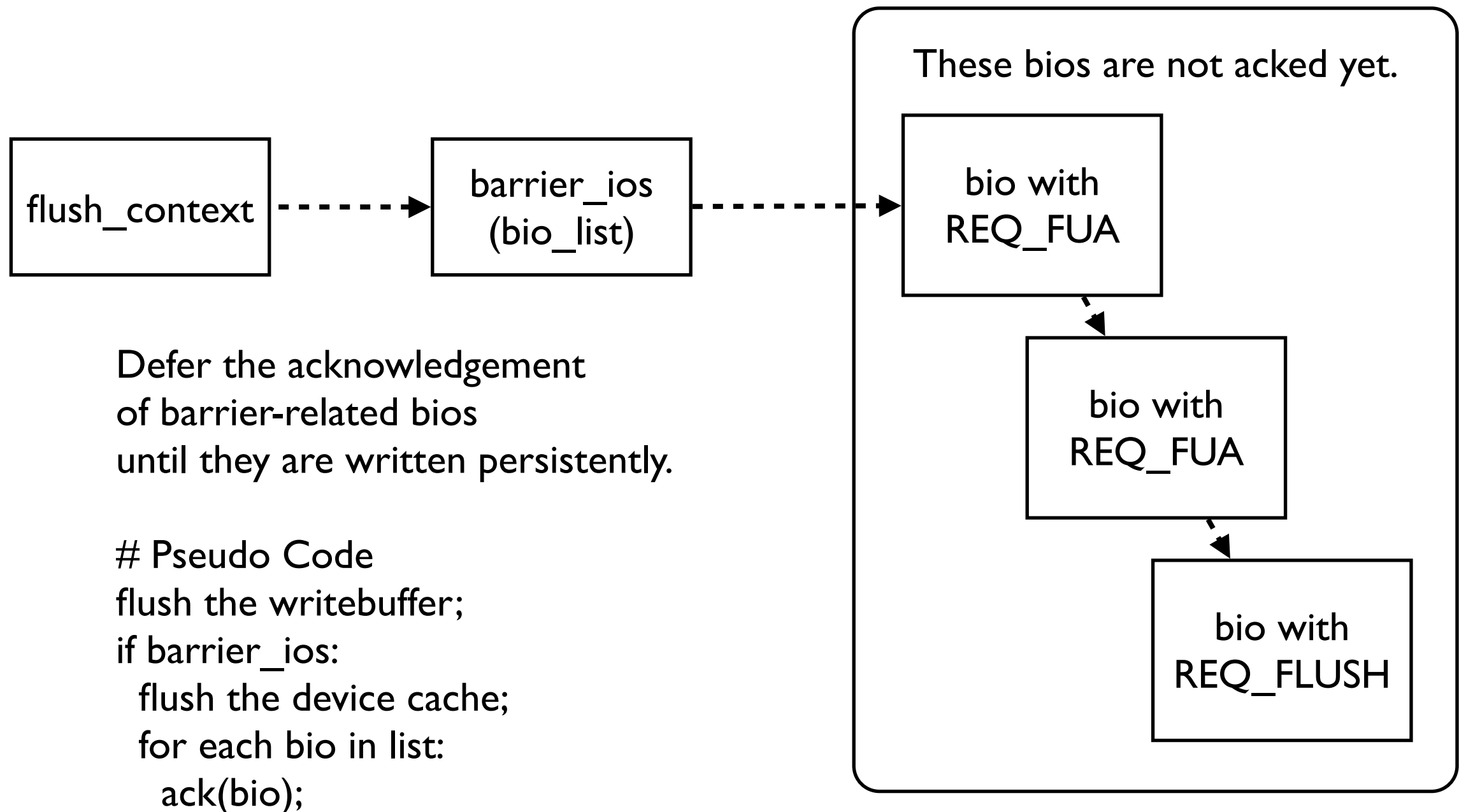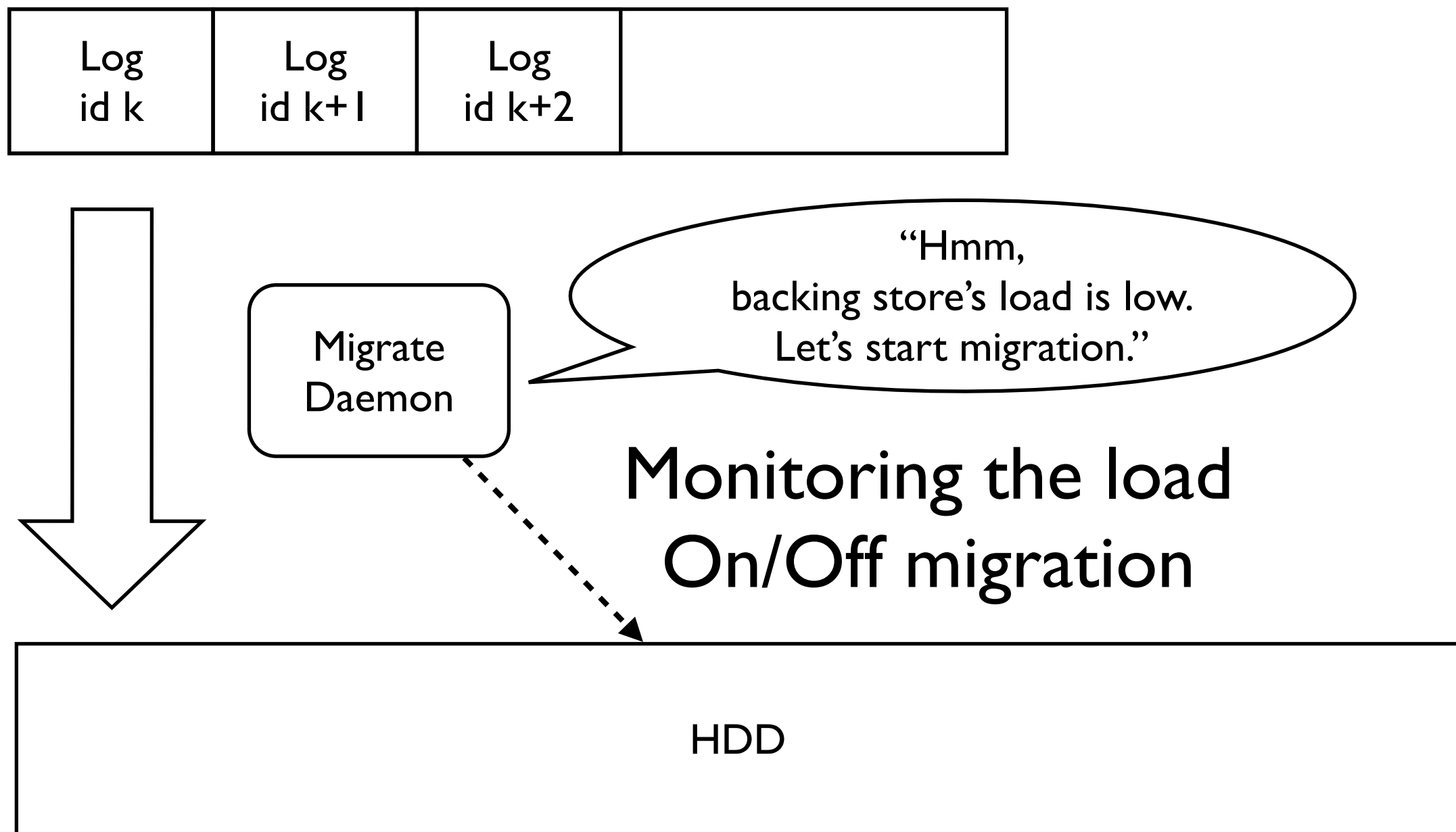| |
|---|
| global_id |
| length (=n) |
| lap |
| metablock[0] |
| metablock[1] |
| metablock[2] |
| metablock[3] |

...

This segment_header says that only n-blocks at the head are dirty. Since both data and metadata is written, data are restored after any server failure.

Deferred ack
for write barriers

These bios are not acked yet.

flush_context - - - → barrier_ios (bio_list) - - - → bio with REQ_FUA

bio with REQ_FUA

bio with REQ_FLUSH

Defer the acknowledgement
of barrier-related bios
until they are written persistently.

```
# Pseudo Code
flush the writebuffer;
if barrier_ios:
  flush the device cache;
  for each bio in list:
    ack(bio);
```
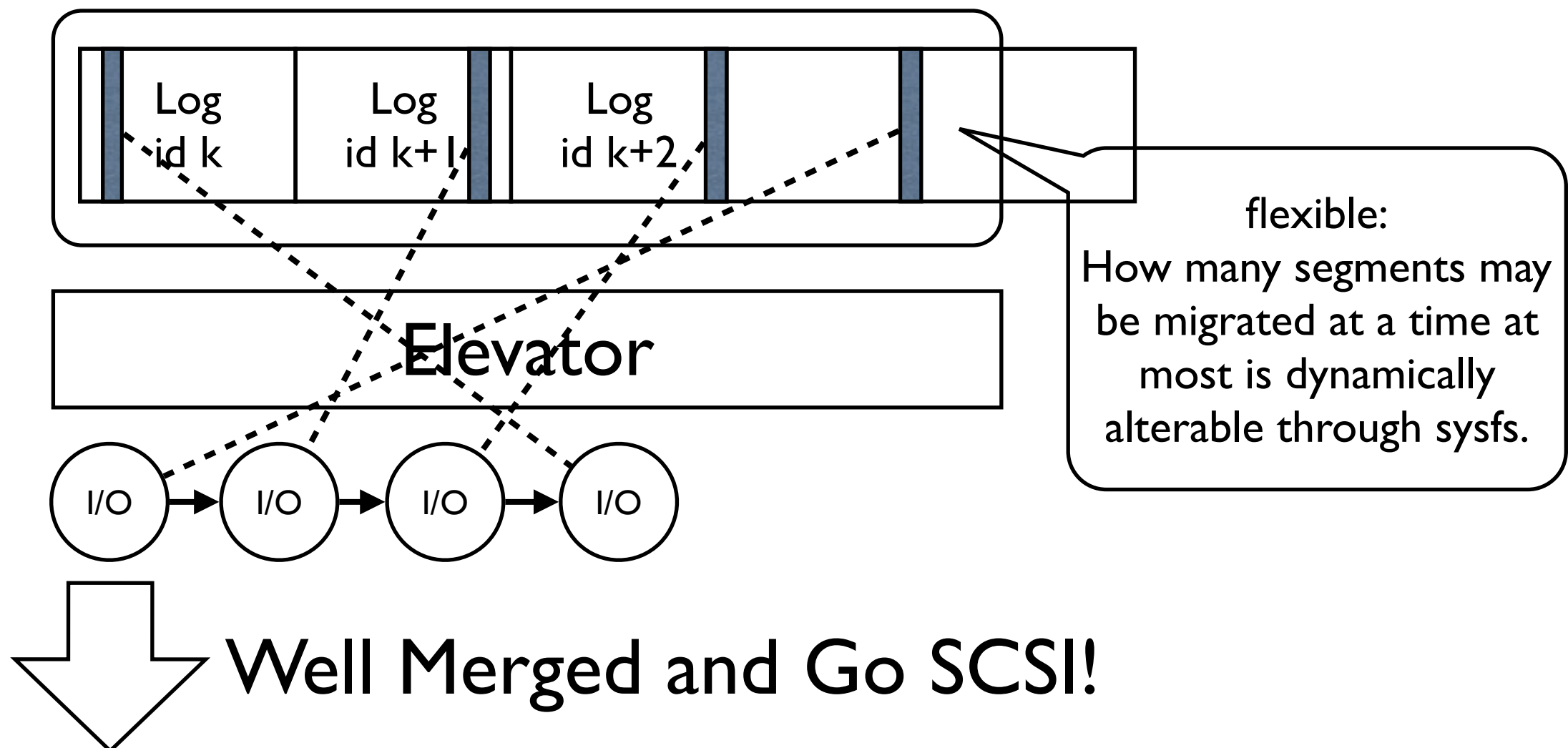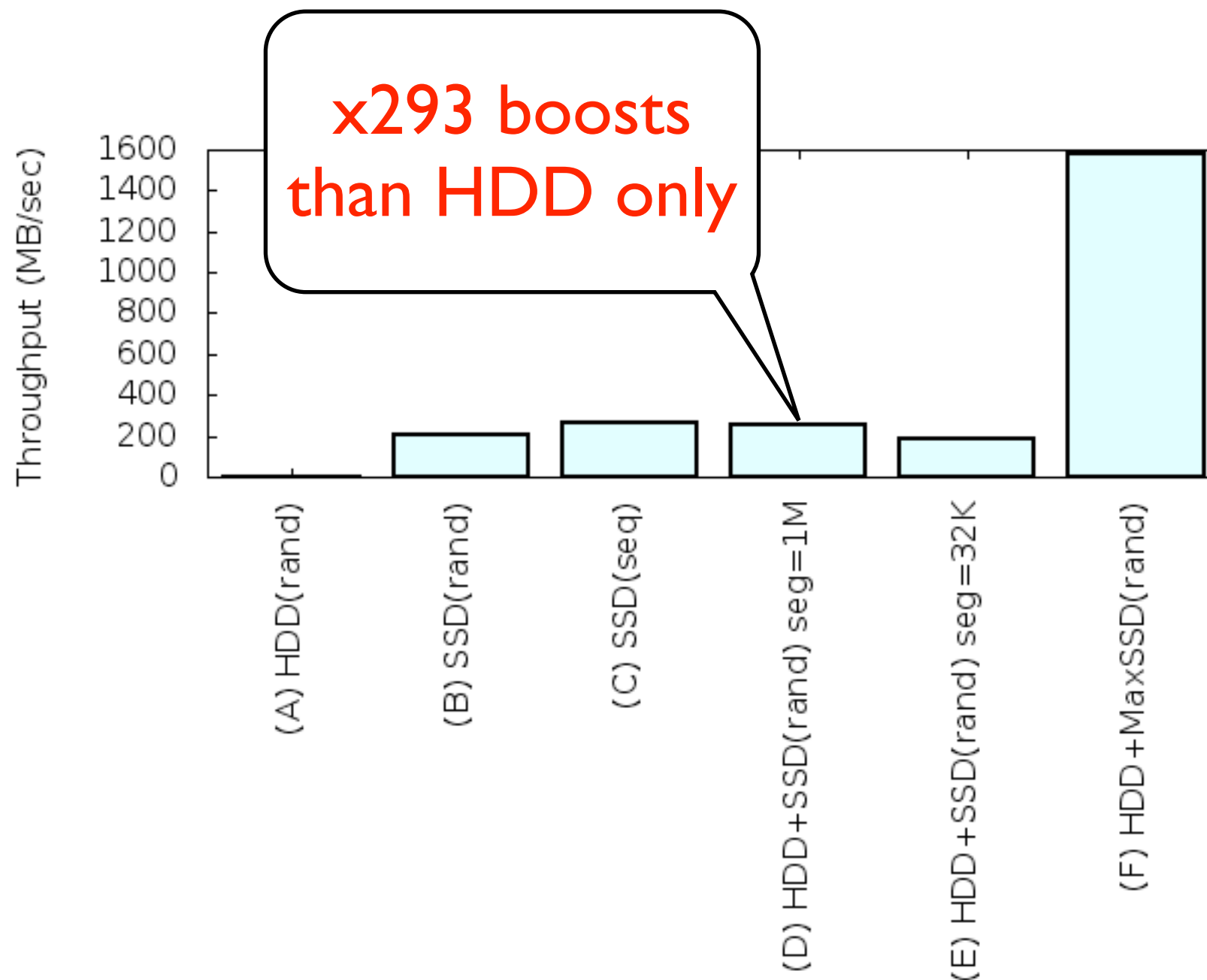
Autonomous Migration

Batched Migration

- Multiple segments are migrated at a time and bios are likely to be merged even if they are so distant in time space that they could not be merged without dm-lc's buffering aid.

Log
id k

Log
id k+1

Log
id k+2

flexible:
How many segments may be migrated at a time at most is dynamically alterable through sysfs.

Elevator

I/O   I/O   I/O   I/O

Well Merged and Go SCSI!

Performance

x293 boosts than HDD only

Throughput (MB/sec)

(A) HDD(rand)
(B) SSD(rand)
(C) SSD(seq)
(D) HDD+SSD(rand) seg=1M
(E) HDD+SSD(rand) seg=32K
(F) HDD+MaxSSD(rand)

Any question or discussion is welcome either on dm-devel or via personal e-mails.