The computer wants to lose your data

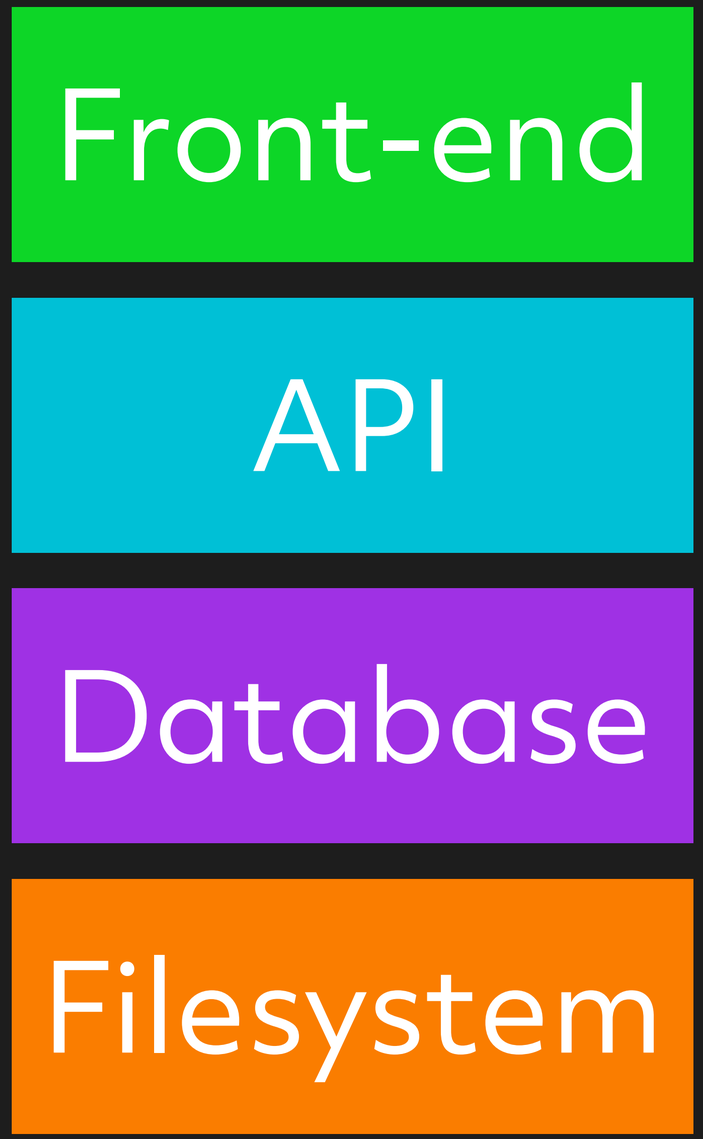
sinjo.dev

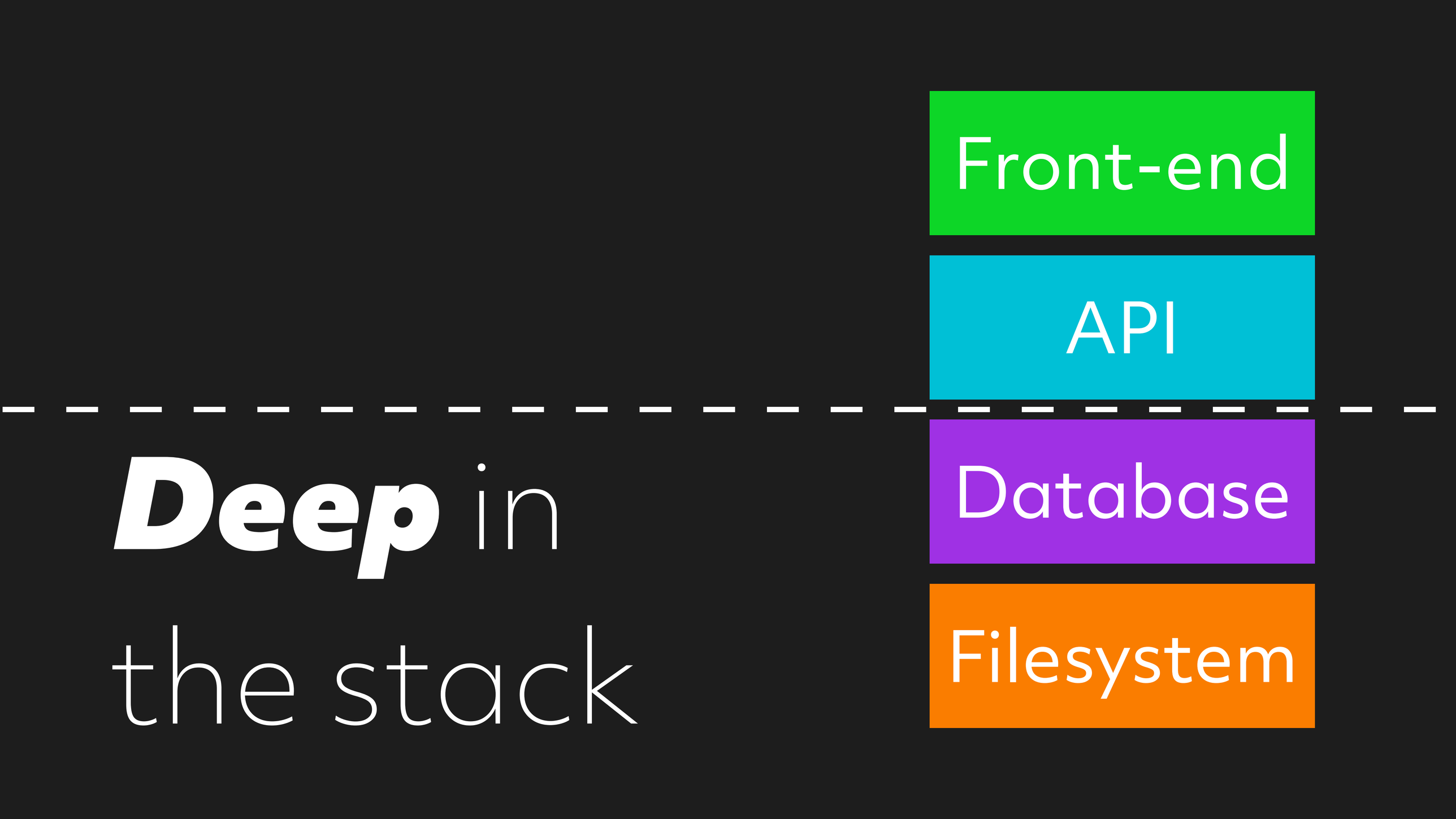
```
INSERT INTO products VALUES ("sunglasses", 27.99);
```

```
INSERT INTO products VALUES ("sunglasses", 27.99);

SELECT * FROM products;
 id |     name     | price
----------------------------
  1 | sunglasses | 27.99
```

```
INSERT INTO products VALUES ("sunglasses", 27.99);

SELECT * FROM products;
 id |     name     | price
----------------------------
  1 | sunglasses | 27.99
```

Deep in the stack




Front-end
API
Database
Filesystem
Deep in
the stack

Deep in
the stack

Database

Filesystem

Hi

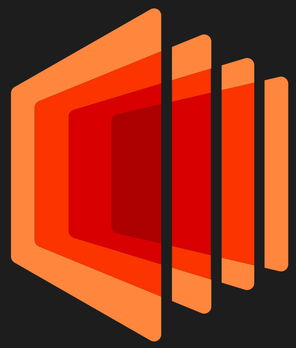
sinjo.dev

sinjo.dev

Infra Engineer

PlanetScale

PostgreSQL



Vitess

I ❤️ database

(most of the time)

```
INSERT INTO products VALUES ("sunglasses", 27.99);
```

Case studies

Case studies

- **MySQL:** doublewrite buffer

Case studies

- **MySQL:** doublewrite buffer
- **Postgres:** fsyncgate

Case studies

- **MySQL:** doublewrite buffer
- **Postgres:** fsyncgate
- **Disk:** write-back caches

Caveats

I am not a:

I am not a:

- Hardware engineer

I am not a:

- Hardware engineer
- Filesystem engineer

I am not a:

- Hardware engineer
- Filesystem engineer
- Database storage engineer

Someone who cares

a lot

about database reliability

Simplifying

lies

ahead

Do ***your own*** research

Case 1

MySQL doublewrite buffer

The promise

```
BEGIN;
INSERT INTO products VALUES ("sunglasses", 27.99);
INSERT INTO products VALUES ("jorts", 10.99);
COMMIT;
```

The promise

```
BEGIN;
INSERT INTO products VALUES ("sunglasses", 27.99);
INSERT INTO products VALUES ("jorts", 10.99);
COMMIT;

SELECT * FROM products;
 id |     name     | price
----------------------------
  1 | sunglasses | 27.99
  2 |      jorts | 10.99
```

The promise

```
BEGIN;
INSERT INTO products VALUES ("sunglasses", 27.99);
INSERT INTO products VALUES ("jorts", 10.99);
COMMIT;
```

Data must be durable

```
SELECT * FROM products;
 id |     name     | price
----------------------------
  1 | sunglasses | 27.99
  2 |      jorts | 10.99
```

The promise

```
BEGIN;
INSERT INTO products VALUES ("sunglasses", 27.99);
INSERT INTO products VALUES ("jorts", 10.99);
COMMIT;

SELECT * FROM products;
 id |     name     | price
----------------------------
  1 | sunglasses | 27.99
  2 |      jorts | 10.99
```

Data must be durable

All or nothing

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

Table

id	name	price

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

Table

id	name	price
1	sunglasses	27.99

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

Table

id	name	price
1	sunglasses	27.99
2	jorts	10.99

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

Table

id	name	price
1	sunglasses	27.99
2	jorts	10.99

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

Table

id	name	price
1	sunglasses	27.99
2		

Table

id	name	price
1	sunglasses	27.99
2		

Now what?

Write-Ahead Logs

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

WAL

Table

id	name	price

SQL

```
➜ BEGIN;
  INSERT ("sunglasses", 27.99);
  INSERT ("jorts", 10.99);
  COMMIT;
```

WAL

```
➜ BEGIN TX 1
```

Table

id	name	price

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

WAL

```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
```

Table

id	name	price
1	sunglasses	27.99

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

WAL

```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
INS 1 (2, "jorts", 10.99)
```

Table

id	name	price
1	sunglasses	27.99
2	jorts	10.99

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

WAL

```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
INS 1 (2, "jorts", 10.99)
COMMIT TX 1
```

Table

id	name	price
1	sunglasses	27.99
2	jorts	10.99

SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

WAL

```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
INS 1 (2, "jorts", 10.99)
```

Table

id	name	price
1	sunglasses	27.99
2		

Never committed

No partial data

WAL

```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
INS 1 (2, "jorts", 10.99)
```

Table

id	name	price

All good!

All good!

Right?

That is where the neat, theoretical version stops…

That is where the neat,
theoretical version
stops...

...but
real computers
are more complicated

Log of every operation

Up-to-date version of data

WAL

```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
INS 1 (2, "jorts", 10.99)
COMMIT TX 1
```

Table

id	name	price
1	sunglasses	27.99
2	jorts	10.99

Logical

id	name	price
1	sunglasses	27.99
2	jorts	10.99

Logical

vs

Physical

id	name	price
1	sunglasses	27.99
2	jorts	10.99

16kB

16kB

16kB

16kB

⋮

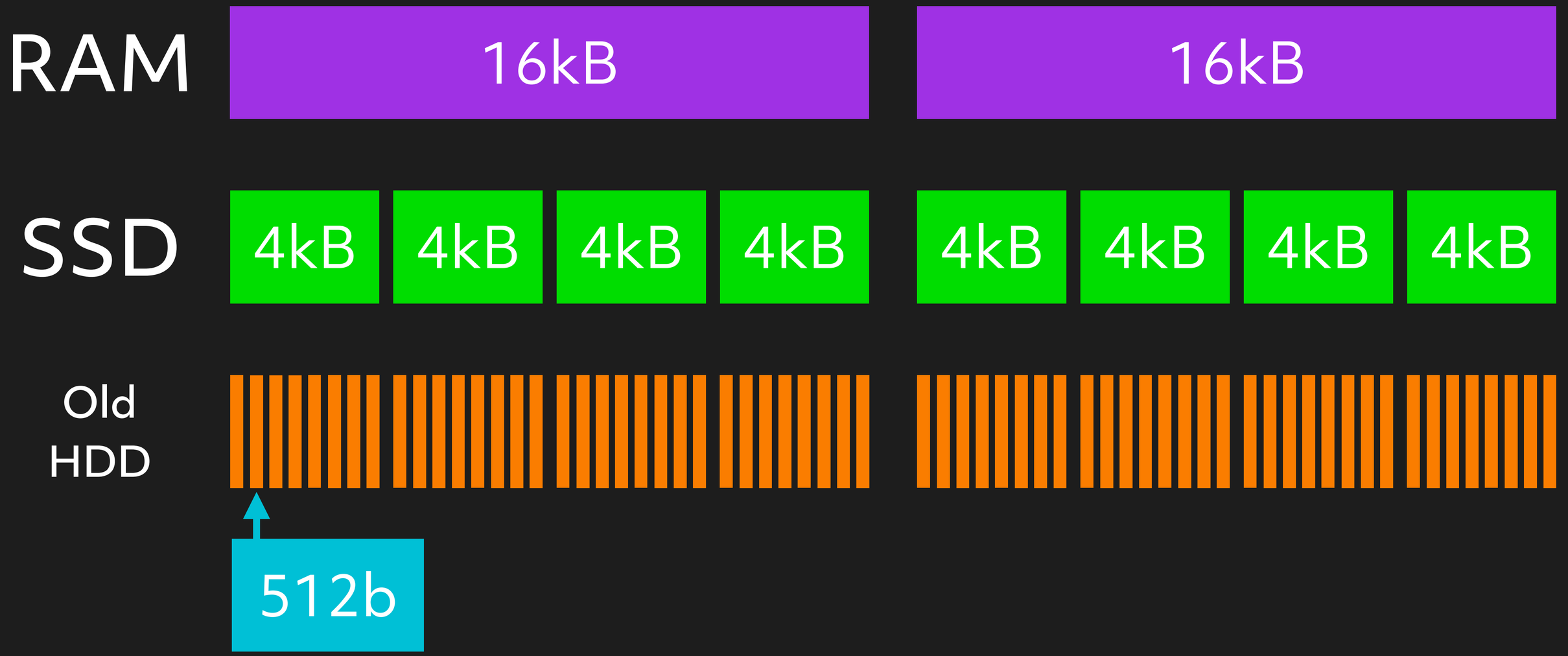
Atomic writes

All or

Nothing

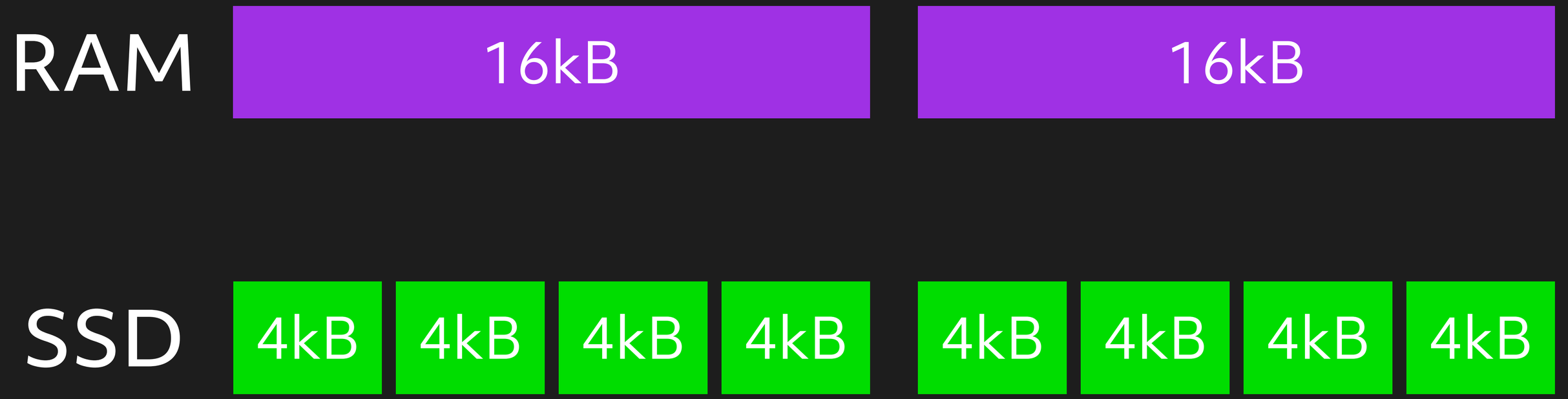
Pages *vs* Sectors

Pages *vs* Sectors

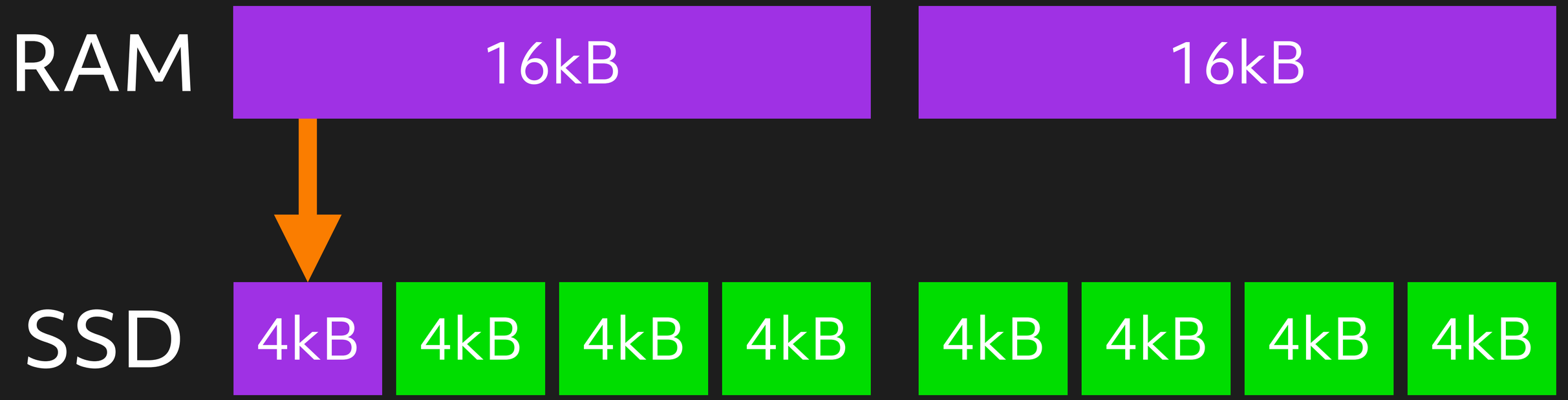


Pages *vs* Sectors

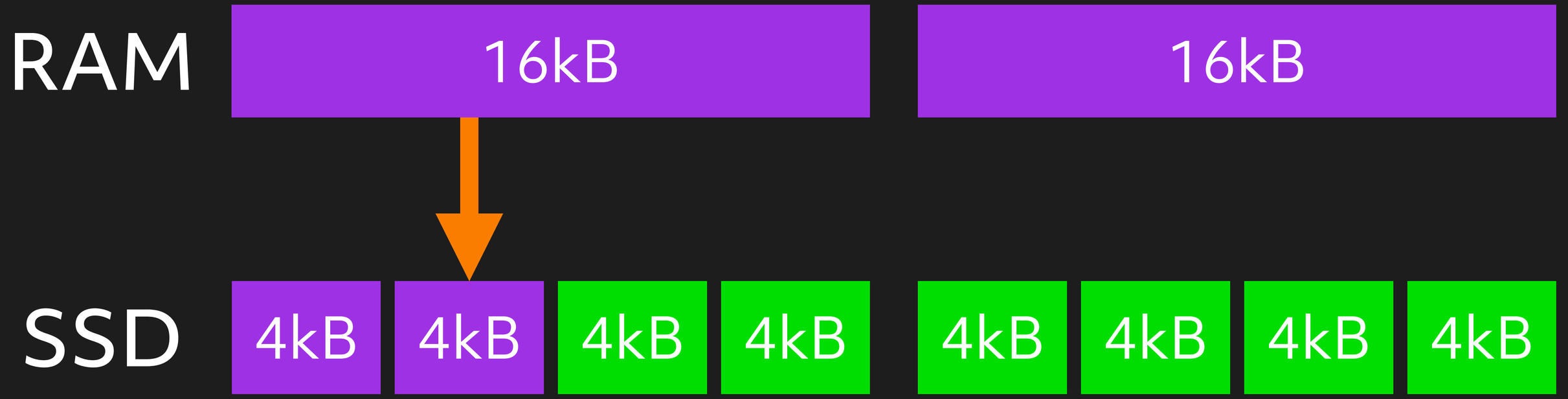
Pages *vs* Sectors



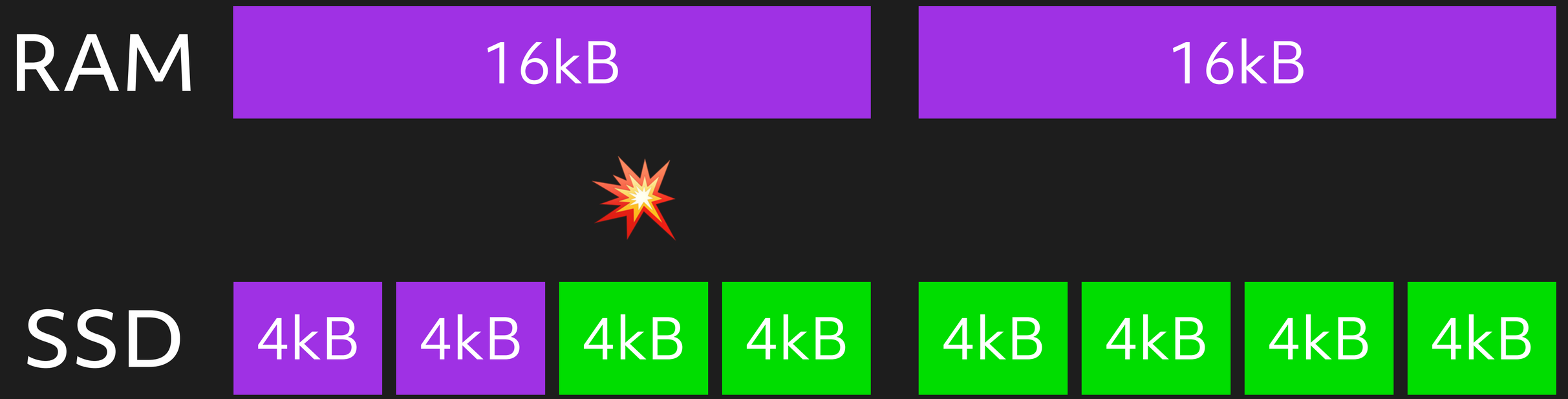
Pages *vs* Sectors



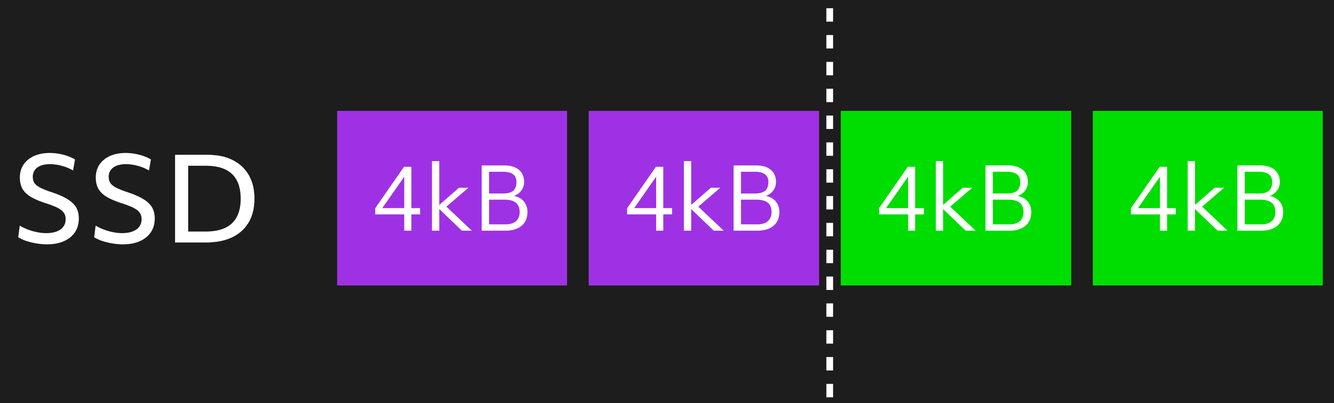
Pages *vs* Sectors



Pages *vs* Sectors



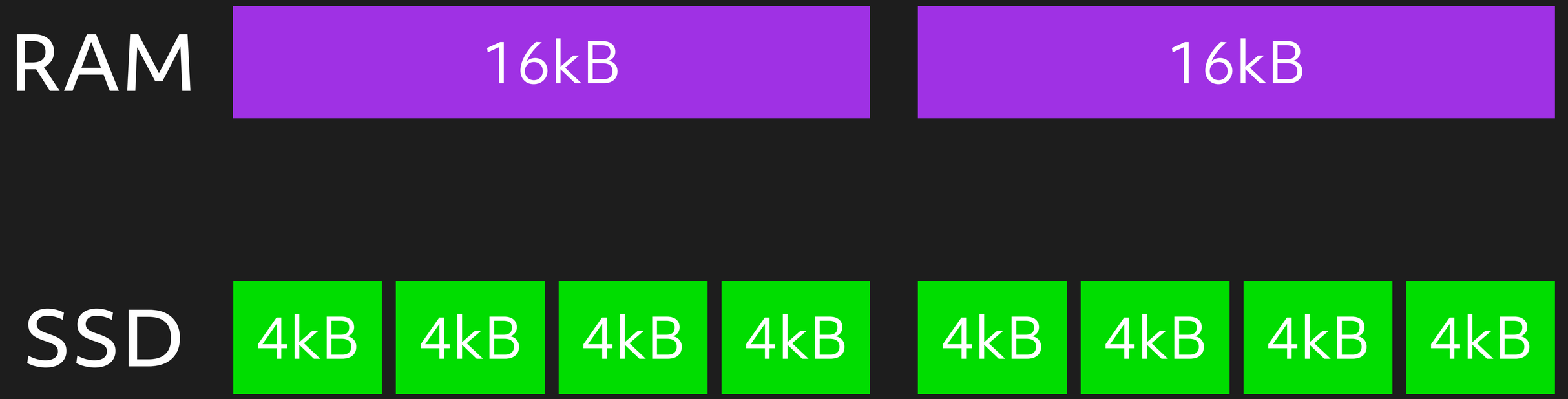
Pages *vs* Sectors



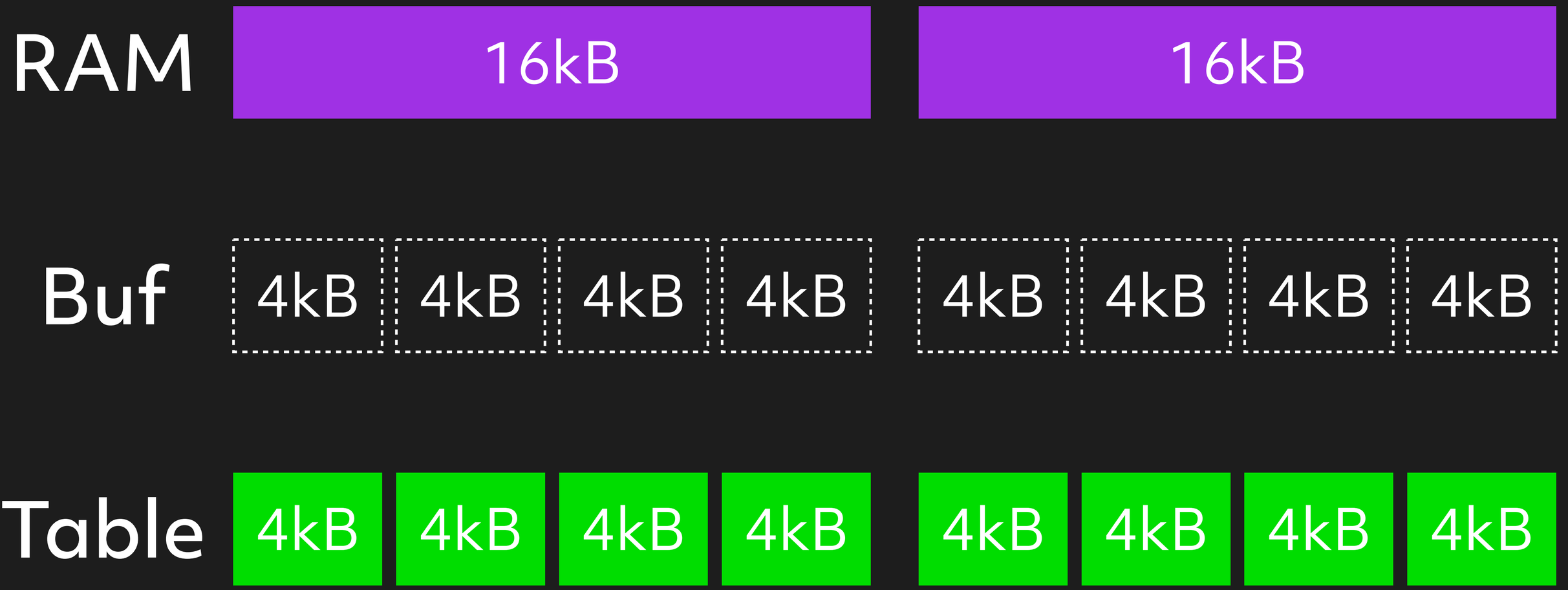
Torn Write

Write-ahead logs
only work
if table data
isn't corrupted

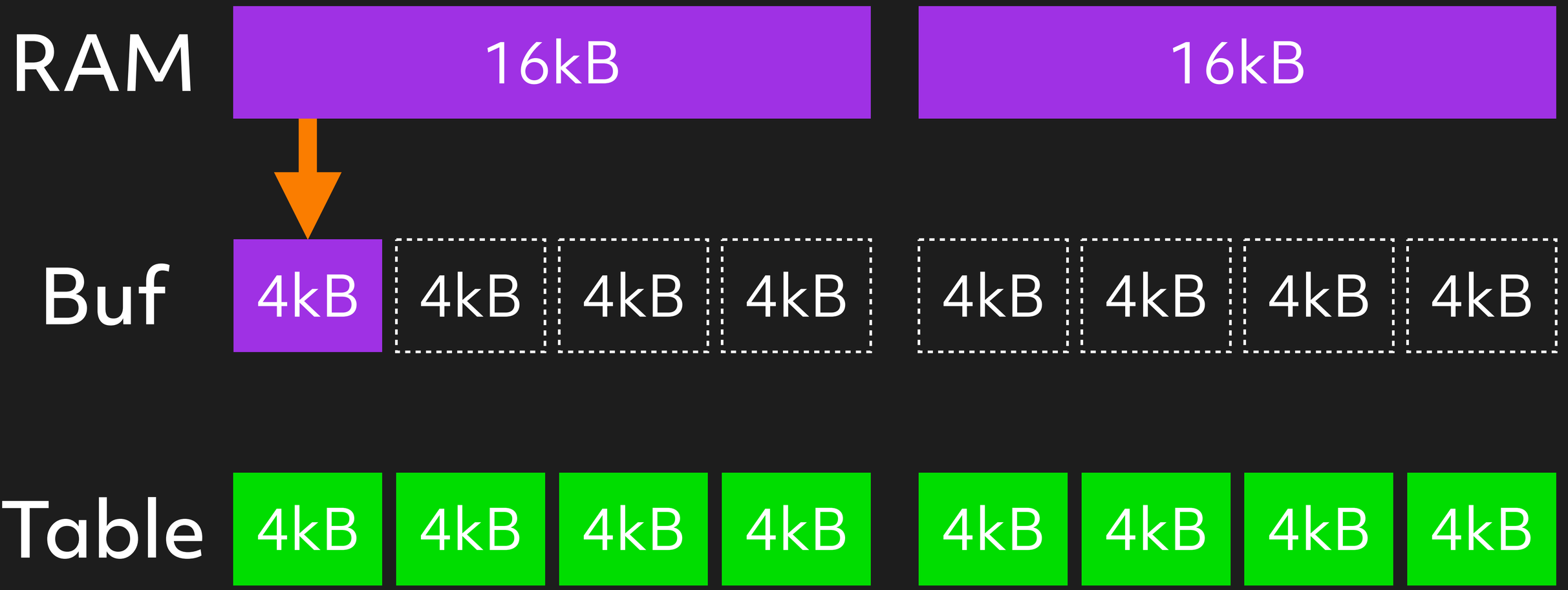
Pages *vs* Sectors



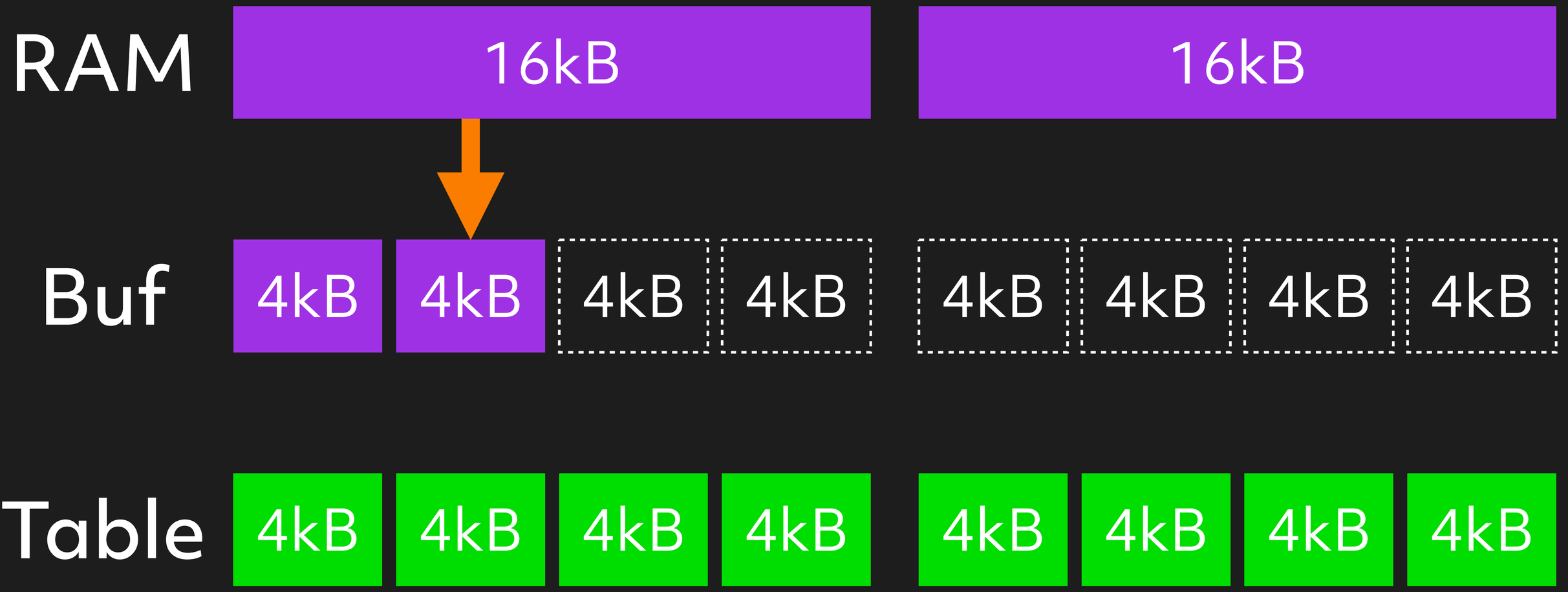
Doublewrite buffer



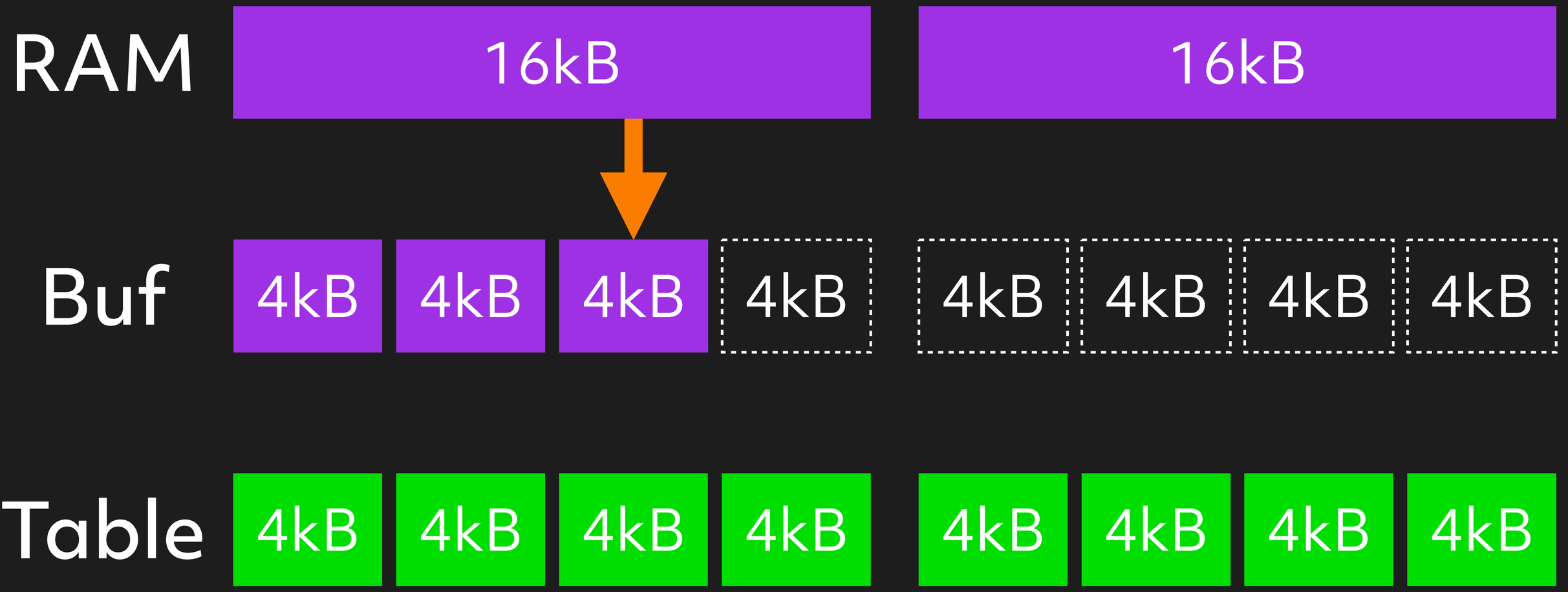
Doublewrite buffer



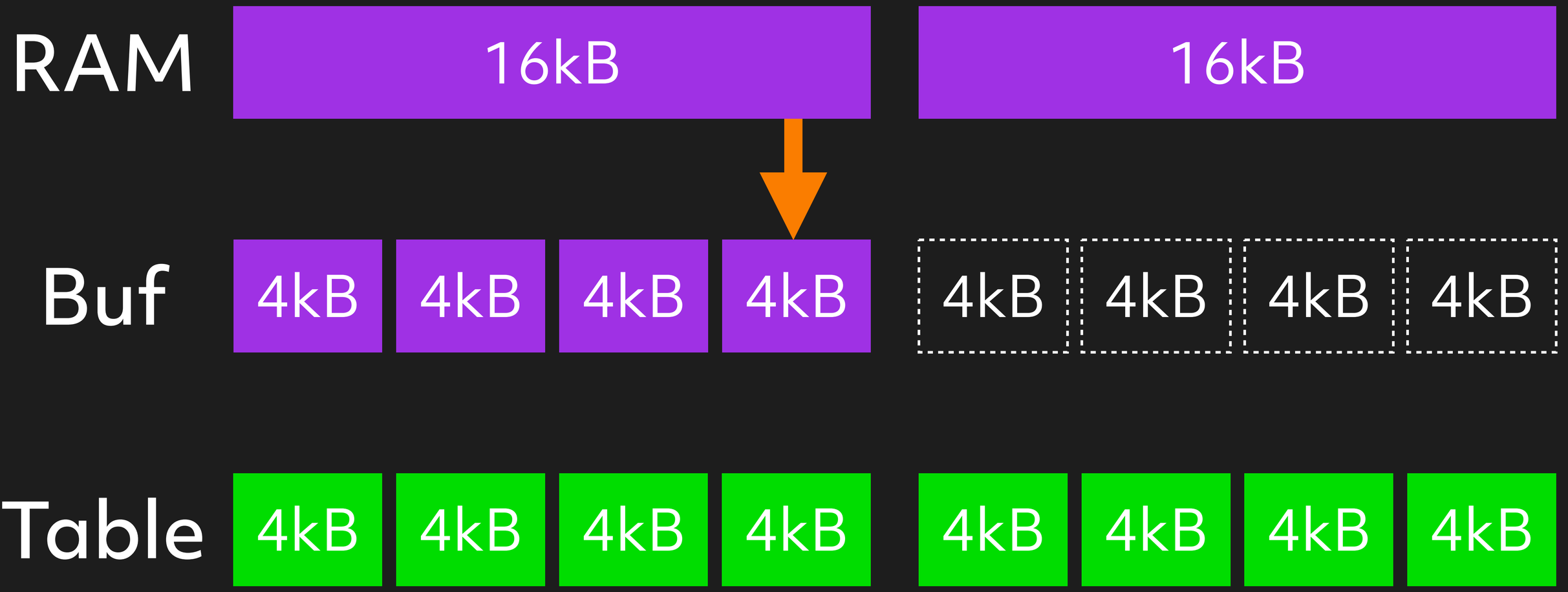
Doublewrite buffer



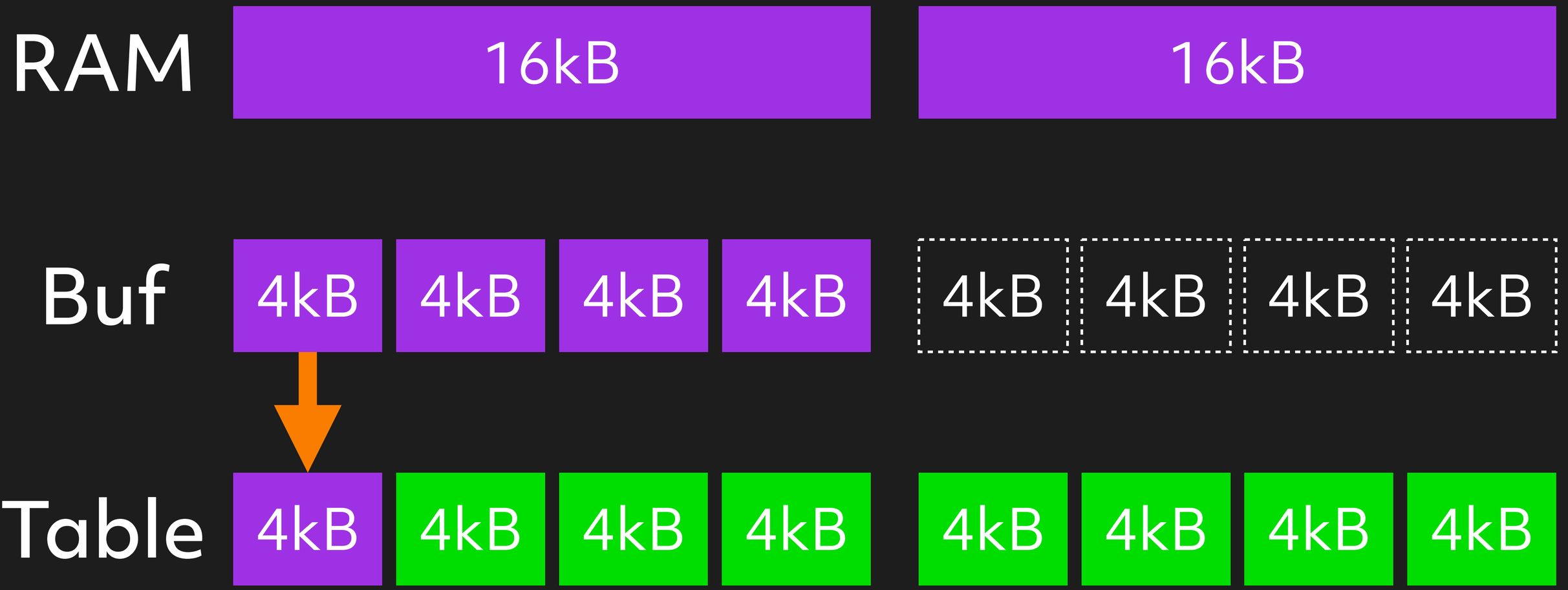
Doublewrite buffer



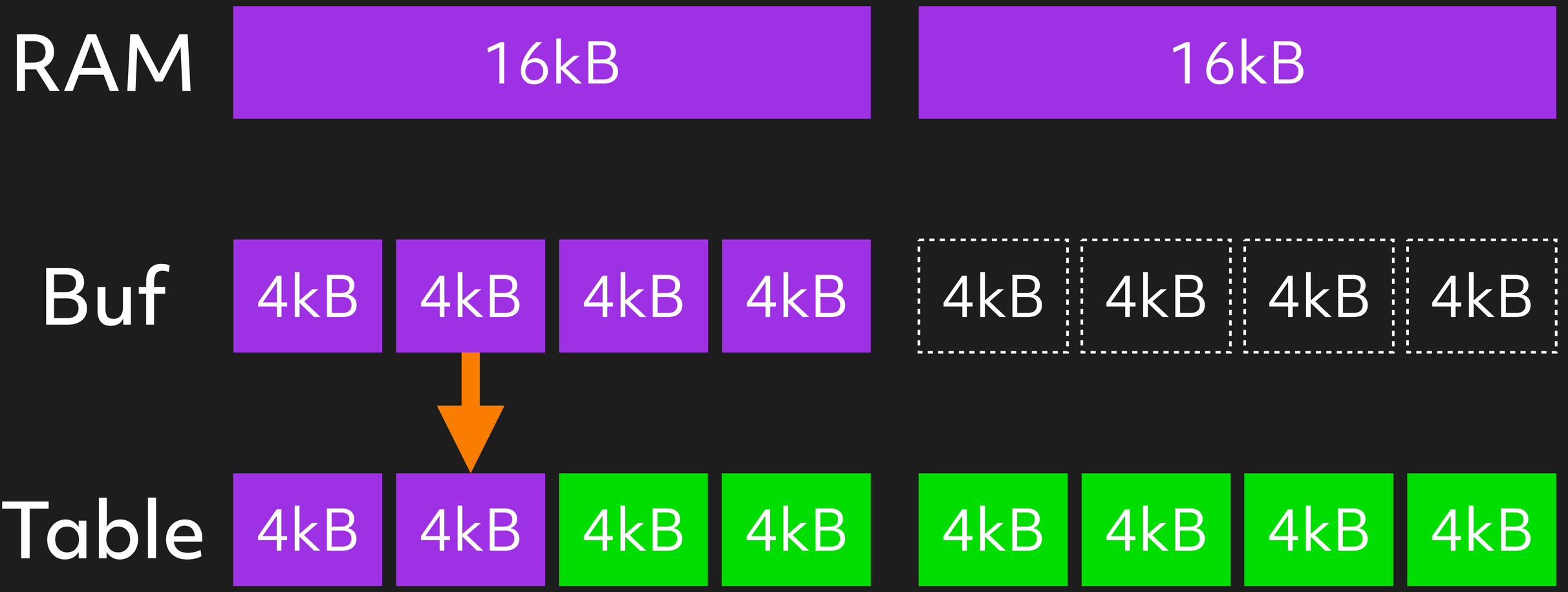
Doublewrite buffer



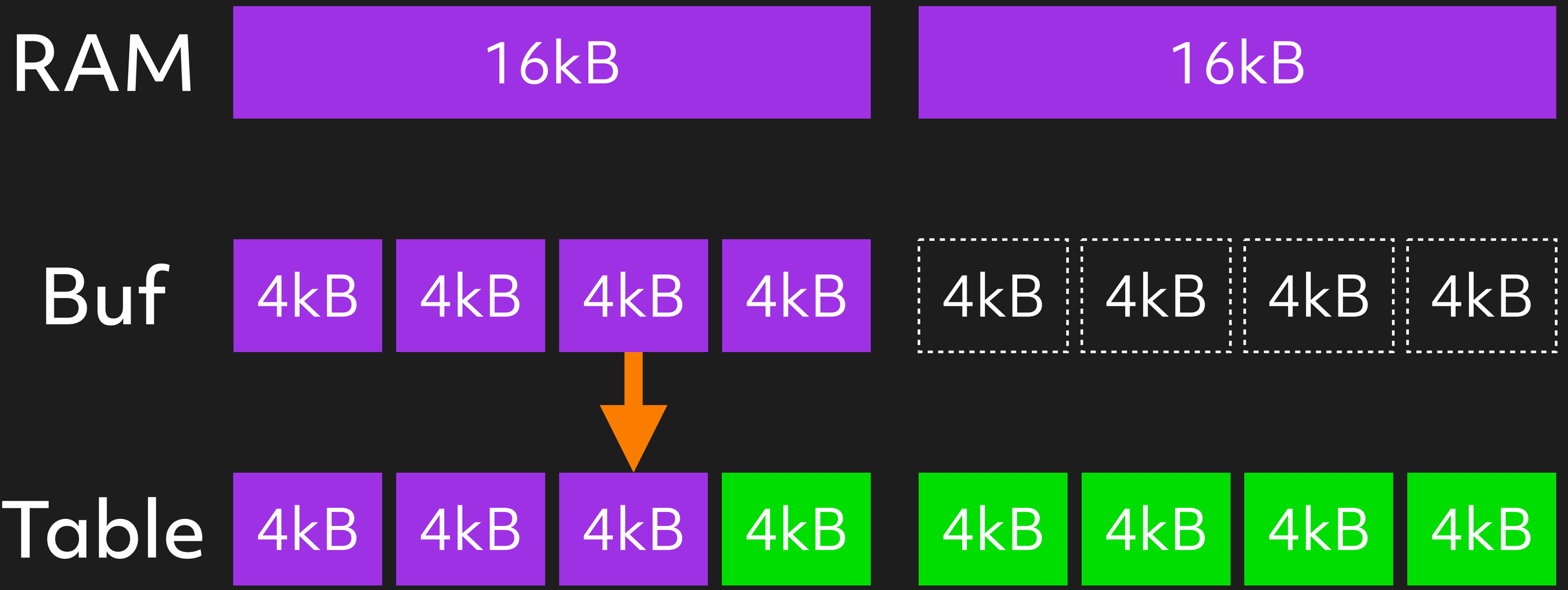
Doublewrite buffer



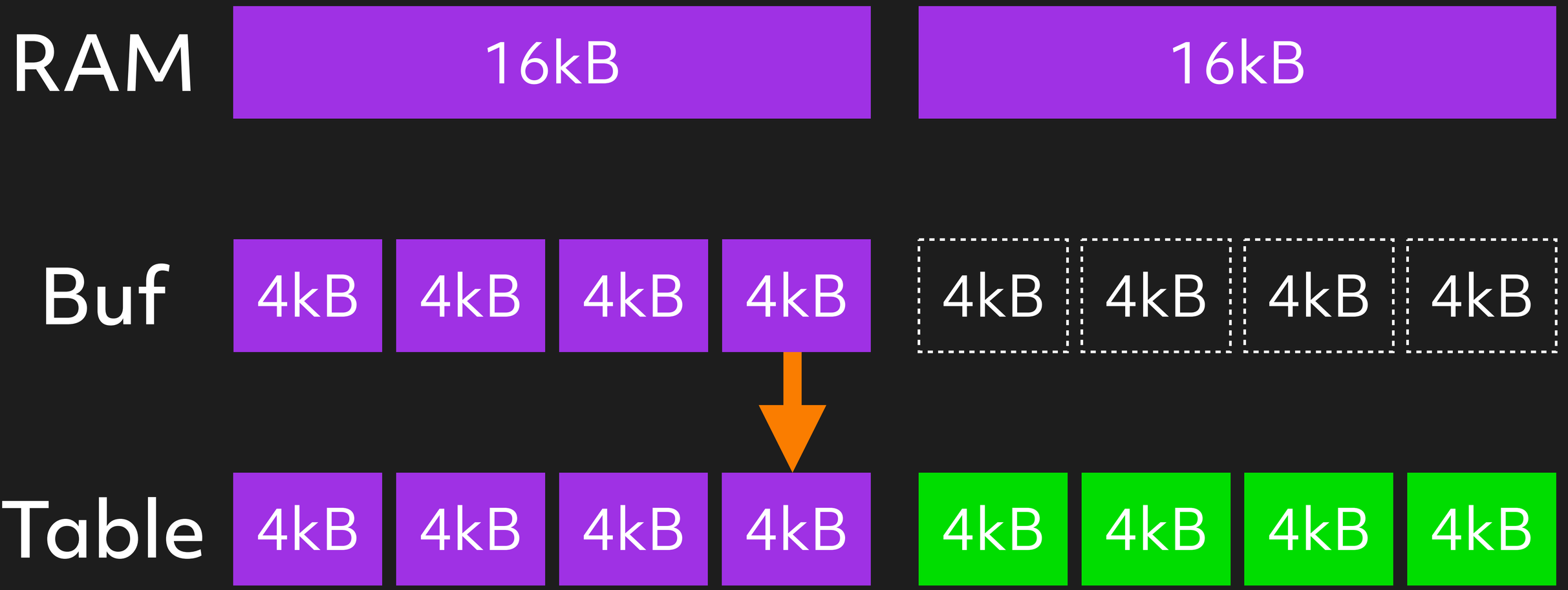
Doublewrite buffer



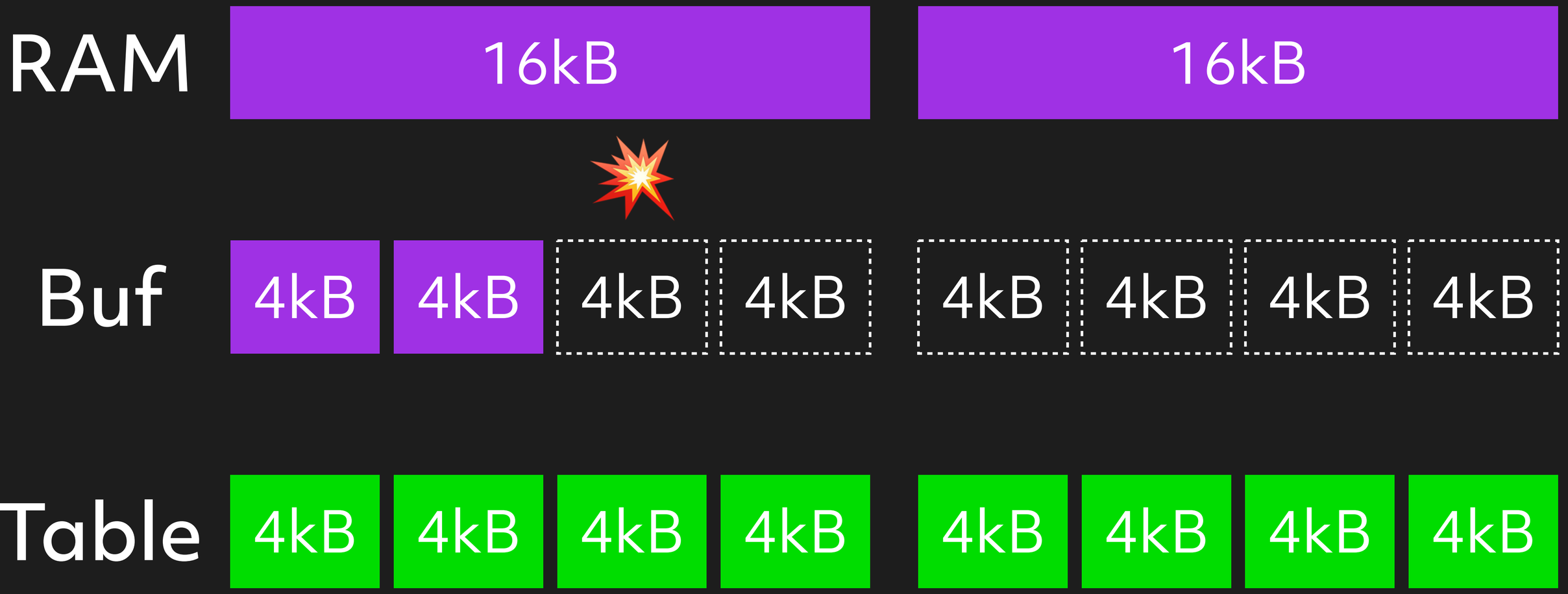
Doublewrite buffer



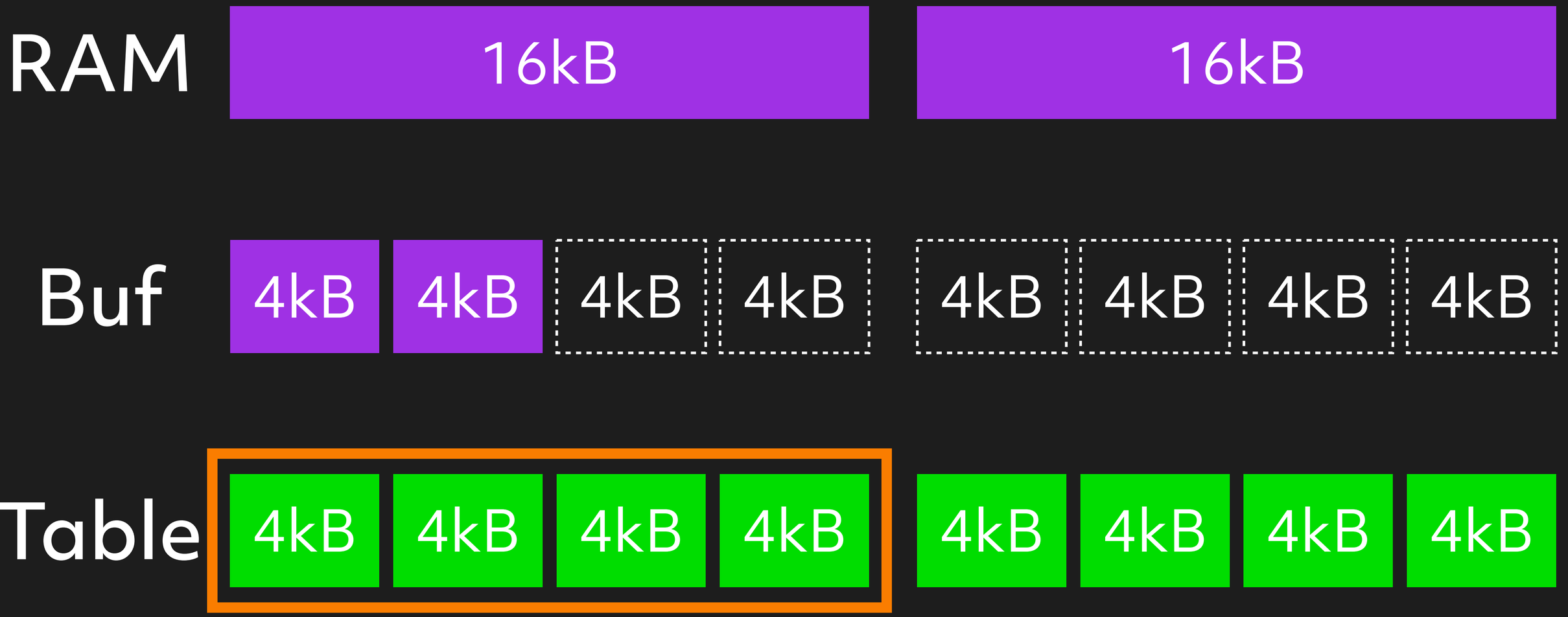
Doublewrite buffer



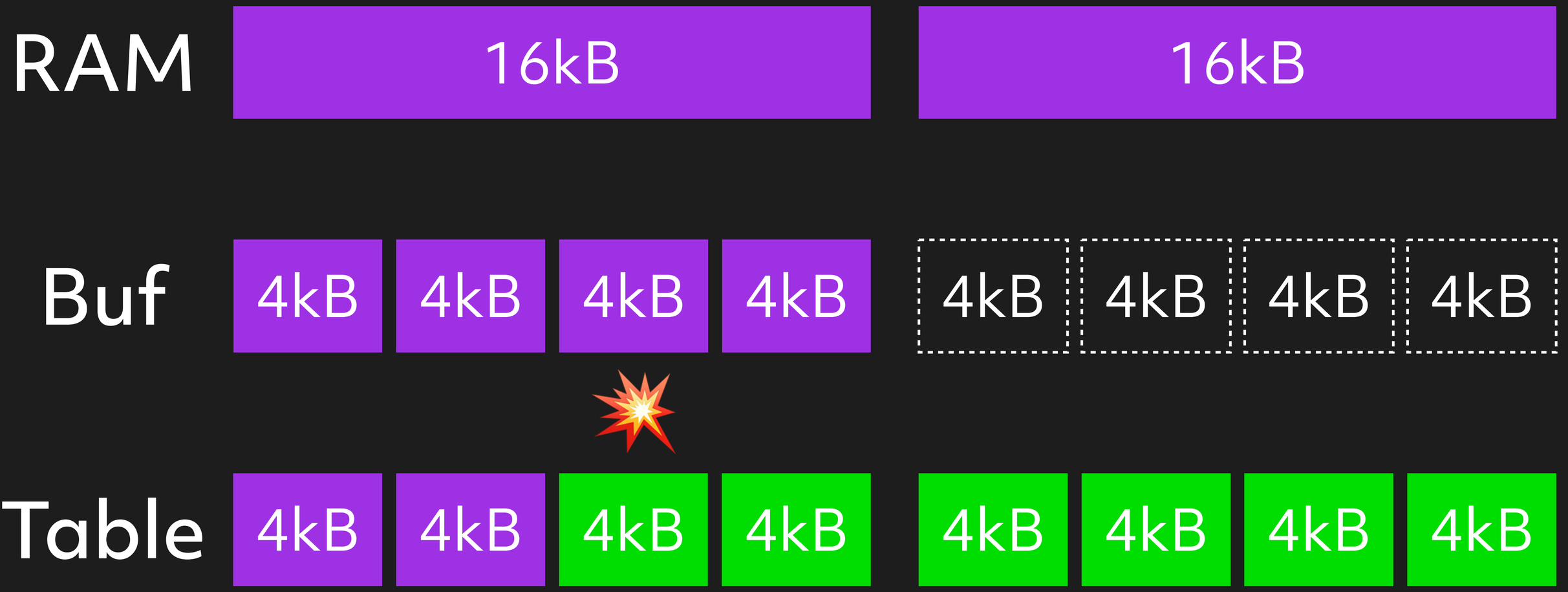
Doublewrite buffer



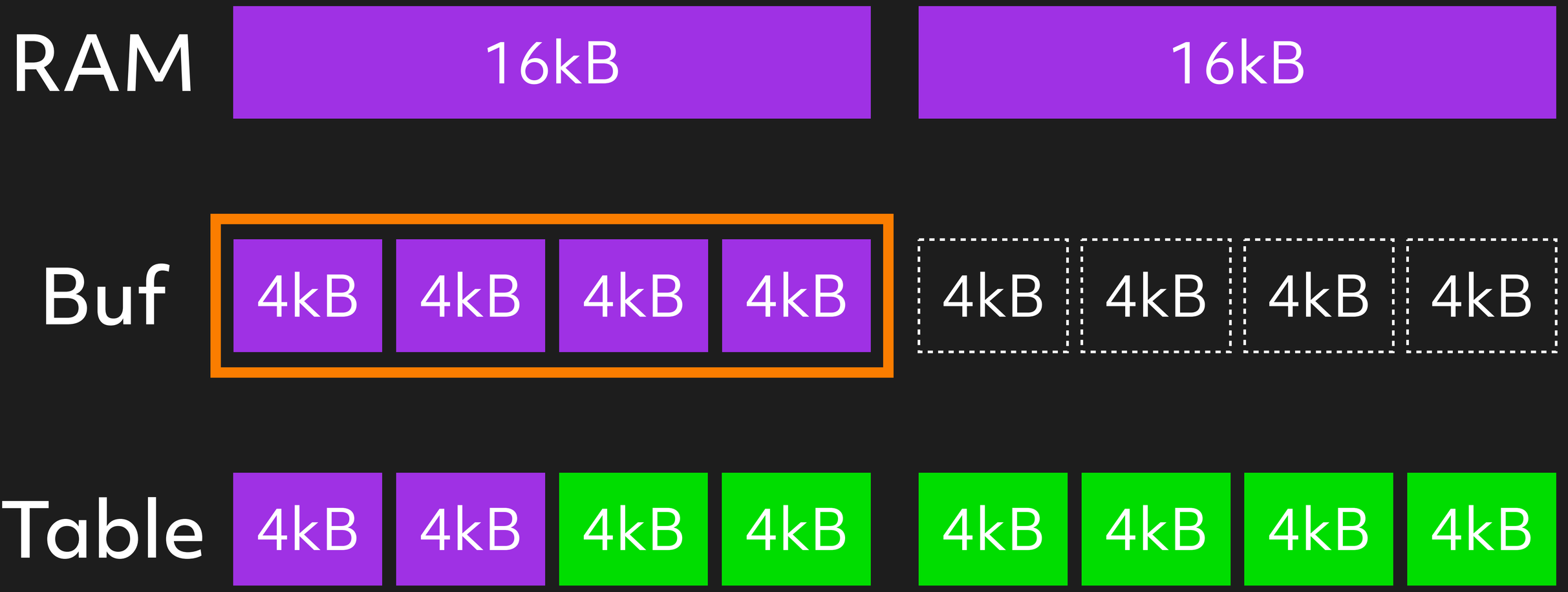
Doublewrite buffer



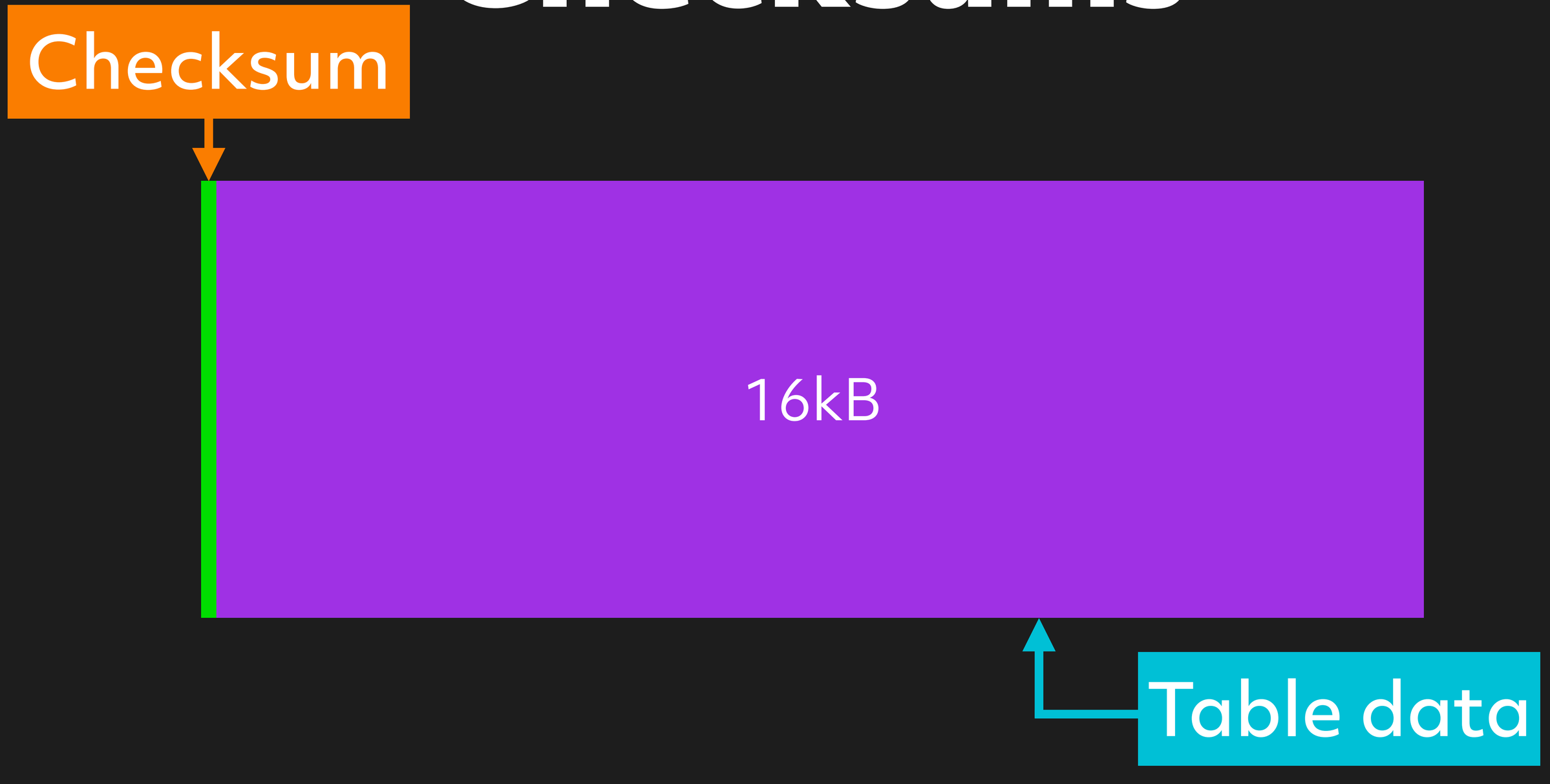
Doublewrite buffer



Doublewrite buffer



Checksums



On restart:

On restart:

- Read doublewrite buffer pages

On restart:

- Read doublewrite buffer pages
- Copy to table if checksum good

On restart:

- Read doublewrite buffer pages
- Copy to table if checksum good
- Ignore if checksum bad

Postgres equivalent:

`full_page_writes`

That's ***a lot*** of work!

Can we ***skip***

it?

Fancy filesystems

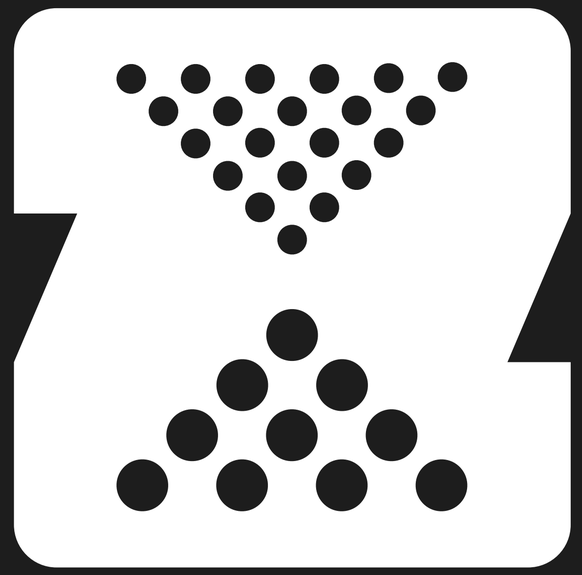
or

Fancy disks

Fancy filesystems

or

Fancy disks



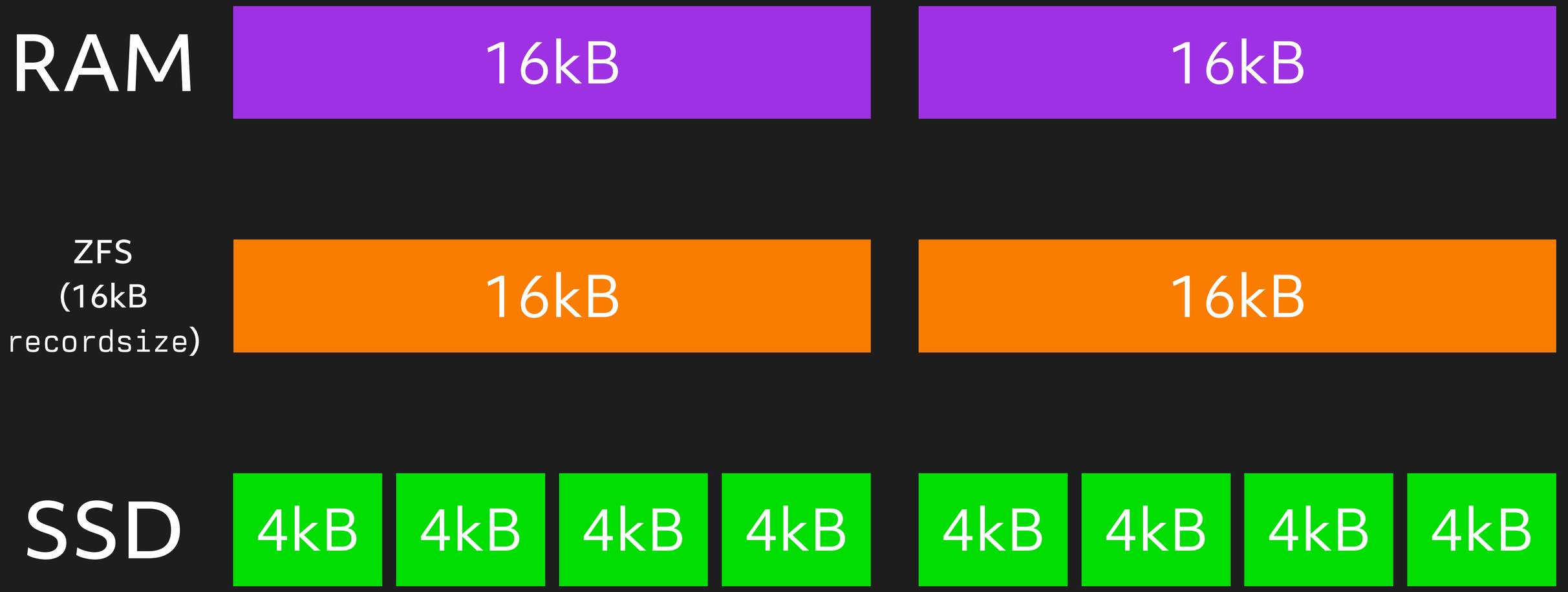
OpenZFS

ZFS

Custom

atomic write size

ZFS


RAM
16kB
16kB
ZFS
(16kB
recordsize)
16kB
16kB
SSD
4kB
4kB
4kB
4kB
4kB
4kB
4kB
4kB

Problem:

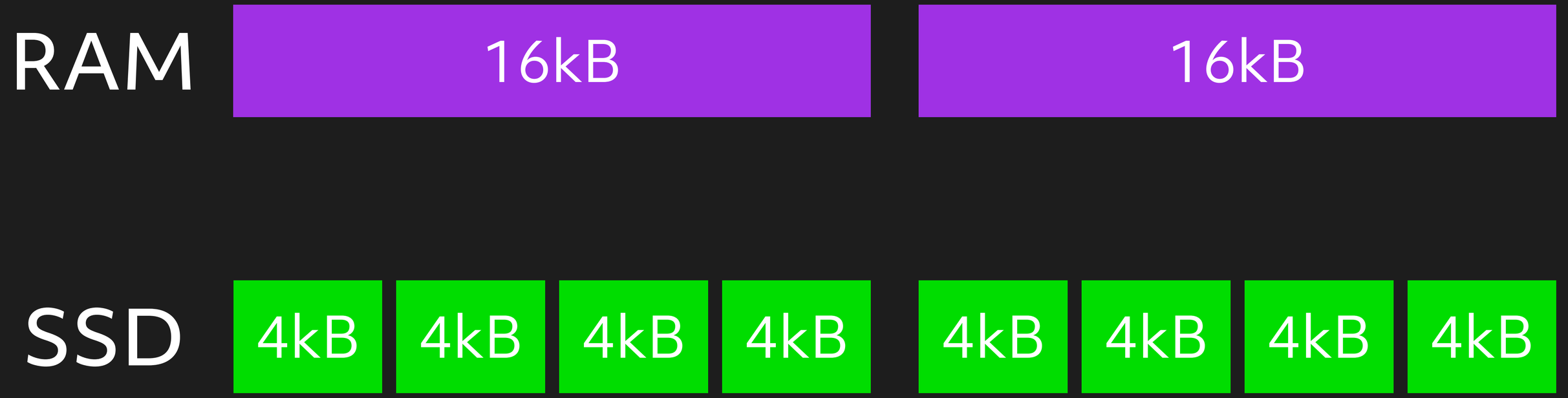
We still pay in performance to do it in software

Fancy filesystems

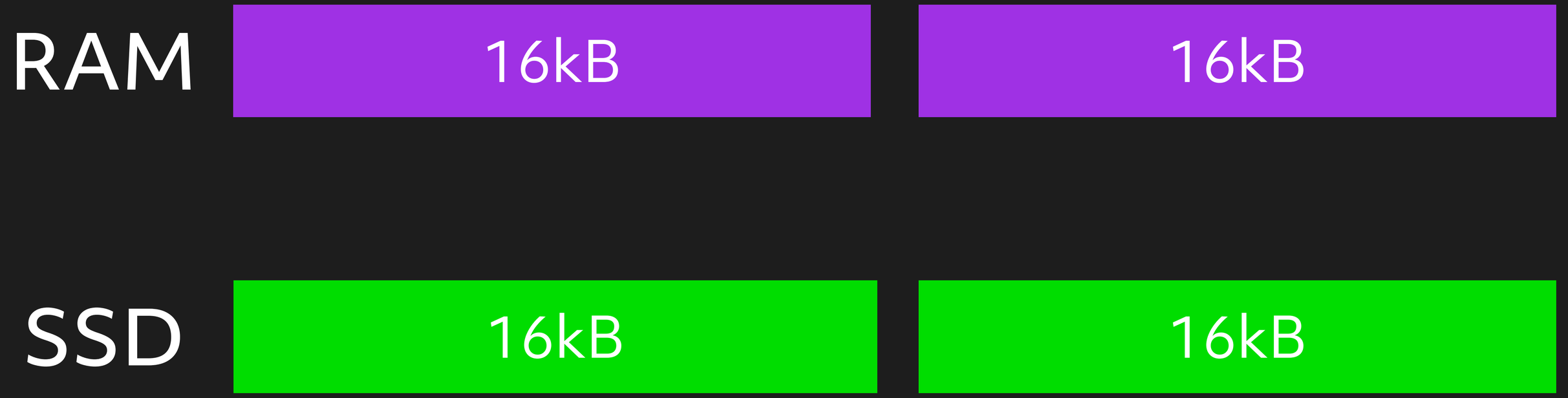
or

Fancy disks

Fancy disks



Fancy disks



NVM Express®

NVM Command Set Specification

Revision 1.2

July 30th, 2025

https://nvmexpress.org/specification/nvm-command-set-specification/

529:528	M	**Atomic Write Unit Power Fail (AWUPF):** This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during a power fail or error condition. If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUPF field in the Identify Namespace data structure. Refer to section 2.1.4. This field is specified in logical blocks and is a 0's based value. The AWUPF value shall be less than or equal to the AWUN value. If a write command is submitted that has a size less than or equal to the AWUPF value, the host is guaranteed that the write is atomic to the NVM with respect to other read or write commands. If a write command is submitted that is greater than this size, there is no guarantee of command atomicity, but atomicity is guaranteed for portions of the command if the command is processed in Multiple Atomicity Mode (refer to section 2.1.4.5). If the write size is less than or equal to the AWUPF value and the write command fails, then subsequent read commands for the associated logical blocks shall return data from the previous successful write command. If a write command is submitted that has a size greater than the AWUPF value, then there is no guarantee of data returned on subsequent reads of the associated logical blocks.

https://nvmexpress.org/specification/nvm-command-set-specification/

529:528	M	**Atomic Write Unit Power Fail (AWUPF):** This field indicates the size of the write operation guaranteed to be written atomically to the NVM across all namespaces with any supported namespace format during a power fail or error condition. If a specific namespace guarantees a larger size than is reported in this field, then this namespace specific size is reported in the NAWUPF field in the Identify Namespace data structure. Refer to section 2.1.4. This field is specified in logical blocks and is a 0's based value. The AWUPF value shall be less than or equal to the AWUN value. If a write command is submitted that has a size less than or equal to the AWUPF value, the host is guaranteed that the write is atomic to the NVM with respect to other read or write commands. If a write command is submitted that is greater than this size, there is no guarantee of command atomicity, but atomicity is guaranteed for portions of the command if the command is processed in Multiple Atomicity Mode (refer to section 2.1.4.5). If the write size is less than or equal to the AWUPF value and the write command fails, then subsequent read commands for the associated logical blocks shall return data from the previous successful write command. If a write command is submitted that has a size greater than the AWUPF value, then there is no guarantee of data returned on subsequent reads of the associated logical blocks.

https://nvmexpress.org/specification/nvm-command-set-specification/

On Linux

```
$ sudo nvme id-ctrl /dev/nvme0n1 | grep awupf
```

On Linux

```
$ sudo nvme id-ctrl /dev/nvme0n1 | grep awupf
awupf     : 0

$ # Not a fancy drive
$ # 1 sector -> 4kB atomicity
$
```

On Linux

```
$ sudo nvme id-ctrl /dev/nvme0n1 | grep awupf
awupf     : 3

$ # Fancy drive :D
$ # 4 sectors -> 16kB atomicity
$
```

It is ***possible***
to turn off the
doublewrite buffer
safely, but…

Caveat

If you're wrong,
the computer might lose your data

Case 2

Postgres fsyncgate (2018)

fsync

In a nutshell

man 2 fsync

fsync(2) System Calls Manual fsync(2)

NAME

fsync, fdatasync - synchronize a file's in-core state with storage device

DESCRIPTION

fsync() transfers ("flushes") all modified in-core data of (i.e., modified buffer cache pages for) the file referred to by the file descriptor fd to the disk device (or other permanent storage device) so that all changed information can be retrieved even if the system crashes or is rebooted. This includes writing through or flushing a disk cache if present. The call blocks until the device reports that the transfer has completed.

man 2 fsync (simplified)

fsync(2) System Calls Manual fsync(2)

NAME

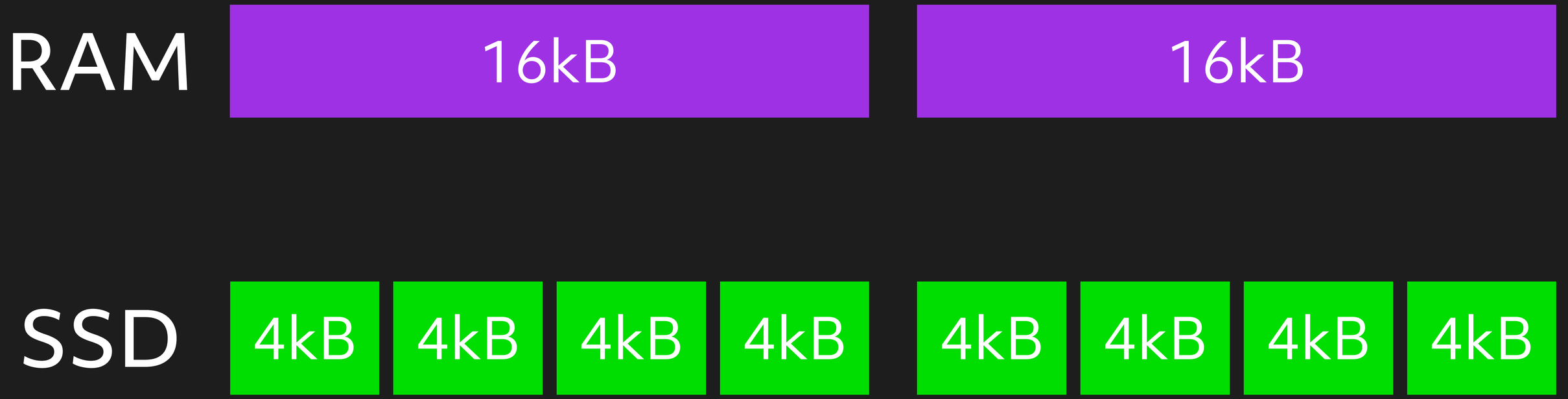
fsync, fdatasync - synchronize a file's in-core state with storage device

DESCRIPTION

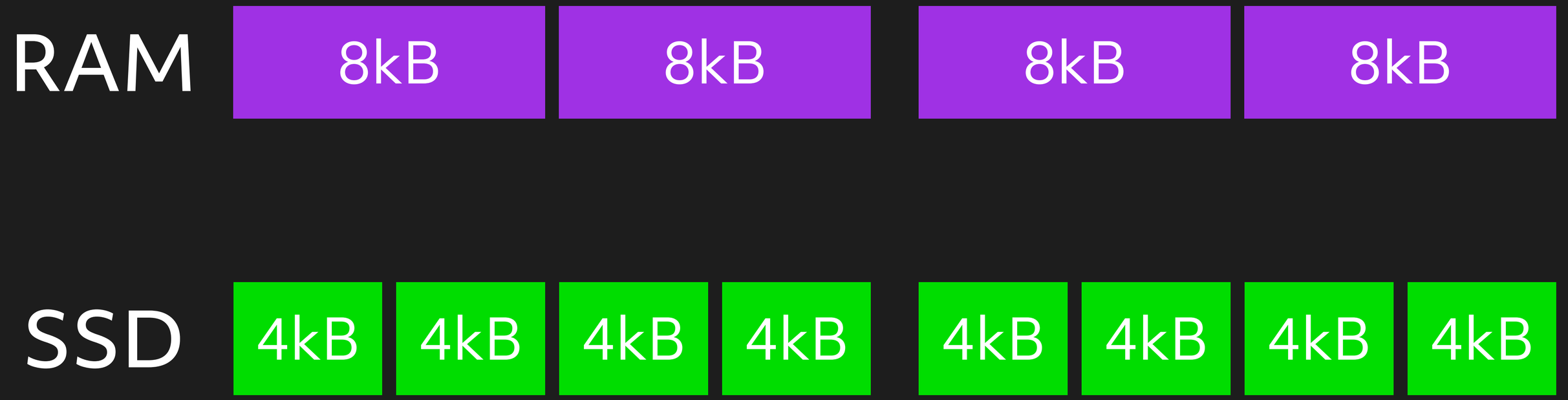
fsync() transfers all modified data of the file to the disk device so that all changed information can be retrieved even if the system crashes or is rebooted.

The call blocks until the device reports that the transfer has completed.

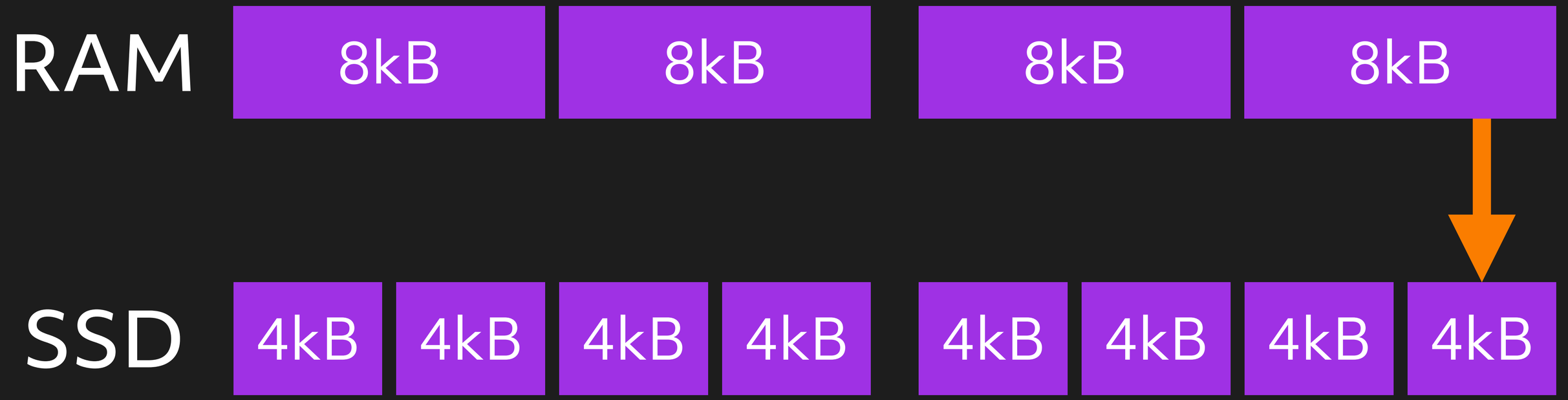
fsync



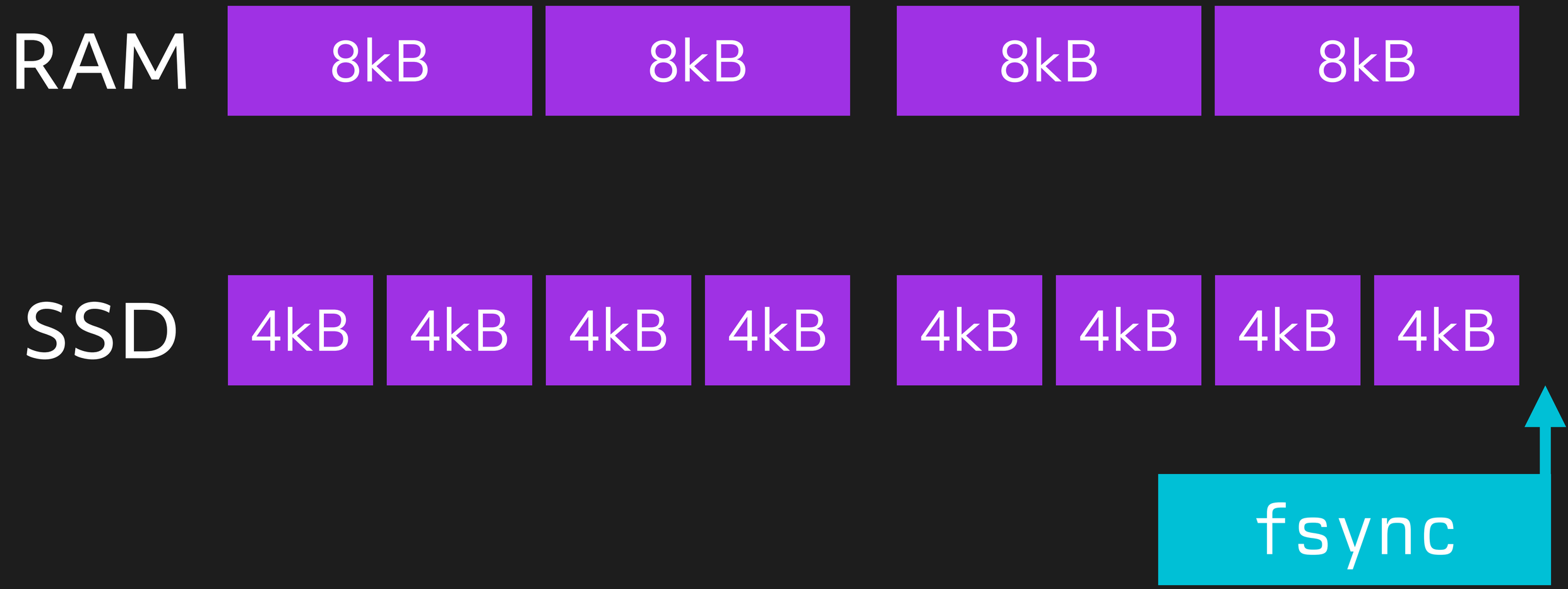
fsync



fsync



fsync



fsync pseudocode

```
file = File.open("/data/base")
file.write("some data")
file.fsync
```

Postgres mailing list

From: Craig Ringer <craig(at)2ndquadrant(dot)com>
To: PostgreSQL Hackers <pgsql-hackers(at)postgresql(dot)org>
Subject: PostgreSQL's handling of fsync() errors is unsafe and risks data loss at least on XFS
Date: 2018-03-28 02:23:46
Message-ID: CAMsr+YHh+5Oq4xziwwoEfhoTZgr07vdGG+hu=1adXx59aTeaoQ@mail.gmail.com
Lists: pgsql-hackers

Hi all

Some time ago I ran into an issue where a user encountered data corruption after a storage error. PostgreSQL played a part in that corruption by allowing checkpoint what should've been a fatal error.

TL;DR: Pg should PANIC on fsync() EIO return. Retrying fsync() is not OK at least on Linux. [redacted]

...

Postgres mailing list

From: Craig Ringer <craig(at)2ndquadrant(dot)com>
To: PostgreSQL Hackers <pgsql-hackers(at)postgresql(dot)org>
Subject: PostgreSQL's handling of fsync() errors is unsafe and risks data loss at least on XFS
Date: 2018-03-28 02:23:46
Message-ID: CAMsr+YHh+5Oq4xziwwoEfhoTZgr07vdGG+hu=1adXx59aTeaoQ@mail.gmail.com
Lists: pgsql-hackers

Hi all

Some time ago I ran into an issue where a user encountered data corruption after a storage error. PostgreSQL played a part in that corruption by allowing checkpoint what should've been a fatal error.

TL;DR: Pg should PANIC on fsync() EIO return. Retrying fsync() is not OK at least on Linux. [redacted]

...

2018-03-28 02:23:46 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-28 03:53:08 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
2018-03-29 02:30:59 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-03-29 02:48:27 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 05:00:31 from Justin Pryzby <pryzby(at)telsasoft(dot)com>
2018-03-29 05:06:22 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 05:25:51 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-21 19:21:39 from Gasper Zejn <zejn(at)owca(dot)info>
2018-03-29 05:32:43 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 05:35:47 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 05:58:45 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 12:07:56 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 13:15:10 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 16:20:00 from Catalin Iacob <iacobcatalin(at)gmail(dot)com>
2018-03-29 21:18:14 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-31 13:24:28 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-03-31 16:13:09 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-31 16:38:12 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
2018-04-01 00:20:38 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-01 01:14:46 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 18:13:46 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 18:53:20 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 19:32:45 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 20:38:06 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 20:58:08 from Stephen Frost <sfrost(at)snowman(dot)net>
2018-04-02 23:05:44 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 23:23:24 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 23:27:35 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-03 00:03:39 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-03 00:05:09 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-03 00:07:41 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-03 00:48:00 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-03 02:54:26 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-03 03:45:30 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-03 10:35:39 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 11:26:05 from Greg Stark <stark(at)mit(dot)edu>
2018-04-03 13:36:47 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 14:37:30 from Greg Stark <stark(at)mit(dot)edu>
2018-04-03 16:52:07 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 21:47:01 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-04 00:56:37 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 01:54:50 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 02:05:19 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 02:14:28 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 02:44:22 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 05:29:28 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 06:00:21 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 07:32:04 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 13:49:38 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 15:23:51 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 17:51:03 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-05 23:37:42 from Andrew Gierth <andrew(at)tao11(dot)riddles(dot)org(dot)uk>
2018-04-06 01:27:05 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-06 02:53:56 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-06 03:20:22 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 02:16:07 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-08 02:33:37 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 02:37:47 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 03:27:45 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 03:37:06 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-08 03:46:17 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 10:30:31 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 16:38:03 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 22:29:16 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 23:10:24 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 23:16:25 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-08 23:27:57 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-09 01:55:10 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 02:00:41 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 02:06:12 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 03:15:01 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 13:54:19 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 01:35:06 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 13:42:35 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 13:47:03 from Abhijit Menon-Sen <ams(at)2ndQuadrant(dot)com>
2018-04-09 18:02:21 from Gasper Zejn <zejn(at)owca(dot)info>
2018-04-17 21:29:17 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:34:53 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-18 09:52:22 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:32:45 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:41:42 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-18 11:56:57 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 09:41:06 from Andreas Karlsson <andreas(at)proxel(dot)se>
2018-04-08 10:31:24 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 21:23:21 from Greg Stark <stark(at)mit(dot)edu>
2018-04-08 21:28:43 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-09 01:31:56 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 21:47:04 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 08:45:40 from Greg Stark <stark(at)mit(dot)edu>
2018-04-09 10:50:41 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 12:03:28 from Geoff Winkless <pgsqladmin(at)geoff(dot)dj>
2018-04-09 12:31:27 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 13:33:18 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 14:22:06 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 15:29:36 from Greg Stark <stark(at)mit(dot)edu>
2018-04-09 19:26:21 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:29:16 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:44:31 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:37:03 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:51:12 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:54:05 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:04:20 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 20:30:00 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:37:31 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 01:59:03 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 02:00:59 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 16:54:40 from Greg Stark <stark(at)mit(dot)edu>
2018-04-10 18:58:37 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-10 19:51:01 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-10 20:57:34 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-11 12:23:49 from Greg Stark <stark(at)mit(dot)edu>
2018-04-17 21:19:53 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-18 10:04:30 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 11:46:15 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-18 12:45:53 from Craig Ringer <craig(at)2ndquadrant(dot)com>

2018-03-28 02:23:46 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-28 03:53:08 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
2018-03-29 02:30:59 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-03-29 02:48:27 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 05:00:31 from Justin Pryzby <pryzby(at)telsasoft(dot)com>
2018-03-29 05:06:22 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 05:25:51 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-21 19:21:39 from Gasper Zejn <zejn(at)owca(dot)info>
2018-03-29 05:32:43 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 05:35:47 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 05:58:45 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 12:07:56 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 13:15:10 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 16:20:00 from Catalin Iacob <iacobcatalin(at)gmail(dot)com>
2018-03-29 21:18:14 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-31 13:24:28 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-03-31 16:13:09 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-31 16:38:12 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
2018-04-01 00:20:38 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-01 01:14:46 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 18:13:46 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 18:53:20 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 19:32:45 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 20:38:06 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 20:58:08 from Stephen Frost <sfrost(at)snowman(dot)net>
2018-04-02 23:05:44 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 23:23:24 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 23:27:35 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-03 00:03:39 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-03 00:05:09 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-03 00:07:41 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-03 00:48:00 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-03 02:54:26 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-03 03:45:30 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-03 10:35:39 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 11:26:05 from Greg Stark <stark(at)mit(dot)edu>
2018-04-03 13:36:47 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 14:37:30 from Greg Stark <stark(at)mit(dot)edu>
2018-04-03 16:52:07 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 21:47:01 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-04 00:56:37 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 01:54:50 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 02:05:19 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 02:14:28 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 02:44:22 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 05:29:28 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 06:00:21 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 07:32:04 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 13:49:38 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 15:23:51 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 17:51:03 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-05 23:37:42 from Andrew Gierth <andrew(at)tao11(dot)riddles(dot)org(dot)uk>
2018-04-06 01:27:05 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-06 02:53:56 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-06 03:20:22 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 02:16:07 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-08 02:33:37 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 02:37:47 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 03:27:45 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 03:37:06 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-08 03:46:17 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 10:30:31 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 16:38:03 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 22:29:16 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 23:10:24 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 23:16:25 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-08 23:27:57 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-09 01:55:10 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 02:00:41 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 02:06:12 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 03:15:01 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 13:54:19 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 01:35:06 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 13:42:35 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 13:47:03 from Abhijit Menon-Sen <ams(at)2ndQuadrant(dot)com>
2018-04-09 18:02:21 from Gasper Zejn <zejn(at)owca(dot)info>
2018-04-17 21:29:17 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:34:53 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-18 09:52:22 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:32:45 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:41:42 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-18 11:56:57 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 09:41:06 from Andreas Karlsson <andreas(at)proxel(dot)se>
2018-04-08 10:31:24 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 21:23:21 from Greg Stark <stark(at)mit(dot)edu>
2018-04-08 21:28:43 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-09 01:31:56 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 21:47:04 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 08:45:40 from Greg Stark <stark(at)mit(dot)edu>
2018-04-09 10:50:41 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 12:03:28 from Geoff Winkless <pgsqladmin(at)geoff(dot)dj>
2018-04-09 12:31:27 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 13:33:18 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 14:22:06 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 15:29:36 from Greg Stark <stark(at)mit(dot)edu>
2018-04-09 19:26:21 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:29:16 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:44:31 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:37:03 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:51:12 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:54:05 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:04:20 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 20:30:00 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:37:31 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 01:59:03 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 02:00:59 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 16:54:40 from Greg Stark <stark(at)mit(dot)edu>
2018-04-10 18:58:37 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-10 19:51:01 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-10 20:57:34 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-11 12:23:49 from Greg Stark <stark(at)mit(dot)edu>
2018-04-17 21:19:53 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-18 10:04:30 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 11:46:15 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-18 12:45:53 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 23:31:50 from Mark Kirkwood <mark(dot)kirkwood(at)catalyst(dot)net(dot)nz>
2018-04-19 00:44:33 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-20 20:49:08 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-09 19:47:44 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 22:33:16 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-10 00:32:20 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-09 12:16:38 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 12:54:16 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:41:19 from Justin Pryzby <pryzby(at)telsasoft(dot)com>
2018-04-09 19:59:34 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 01:44:59 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 01:52:21 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-09 16:45:00 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-09 17:26:24 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-09 18:29:42 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-09 19:22:58 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 19:02:11 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-09 19:13:14 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:25:33 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-17 21:49:42 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-09 20:25:54 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-09 20:34:15 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 20:43:03 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:55:29 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-09 21:08:29 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 21:25:52 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 21:33:29 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-10 01:54:30 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 15:15:46 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-10 15:40:05 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-10 16:38:27 from Greg Stark <stark(at)mit(dot)edu>
2018-04-11 12:05:27 from Jonathan Corbet <corbet(at)lwn(dot)net>
2018-04-11 14:29:09 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-11 14:40:31 from Jonathan Corbet <corbet(at)lwn(dot)net>
2018-04-10 05:04:13 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-10 05:37:19 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 06:10:21 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-10 12:15:15 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 10:19:28 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 15:28:07 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-10 00:41:10 from Andreas Karlsson <andreas(at)proxel(dot)se>
2018-04-10 02:02:48 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 17:23:58 from Gasper Zejn <zejn(at)owca(dot)info>
2018-04-04 07:51:53 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 14:00:15 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 14:09:09 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 14:25:47 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 14:42:18 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 15:23:31 from Antonis Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-04 21:28:09 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 22:14:24 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 02:40:16 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 13:53:01 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-03 14:29:10 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-01 00:58:22 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-01 18:24:51 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-02 15:03:42 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-03 01:29:28 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-03 23:59:27 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-05 07:09:57 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-05 08:46:08 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-05 19:33:14 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-23 20:14:48 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-24 00:09:23 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-27 01:18:55 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-26 02:16:52 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-30 04:55:22 from Craig Ringer <craig(at)2ndquadrant(dot)com>

man 2 fsync

fsync(2) System Calls Manual fsync(2)

ERRORS

The fsync() system call will fail if:

...

[EIO] An error occurred during synchronization. This error may relate to data written to some other file descriptor on the same file.

...

Postgres mailing list

From: Craig Ringer <craig(at)2ndquadrant(dot)com>
To: PostgreSQL Hackers <pgsql-hackers(at)postgresql(dot)org>
Subject: PostgreSQL's handling of fsync() errors is unsafe and risks data loss at least on XFS
Date: 2018-03-28 02:23:46
Message-ID: CAMsr+YHh+5Oq4xziwwoEfhoTZgr07vdGG+hu=1adXx59aTeaoQ@mail.gmail.com
Lists: pgsql-hackers

Hi all

Some time ago I ran into an issue where a user encountered data corruption after a storage error. PostgreSQL played a part in that corruption by allowing checkpoint what should've been a fatal error.

TL;DR: Pg should PANIC on fsync() EIO return. Retrying fsync() is not OK at least on Linux. When fsync() returns success it means "all writes since the last fsync have hit disk" but we assume it means "all writes since the last SUCCESSFUL fsync have hit disk".

...

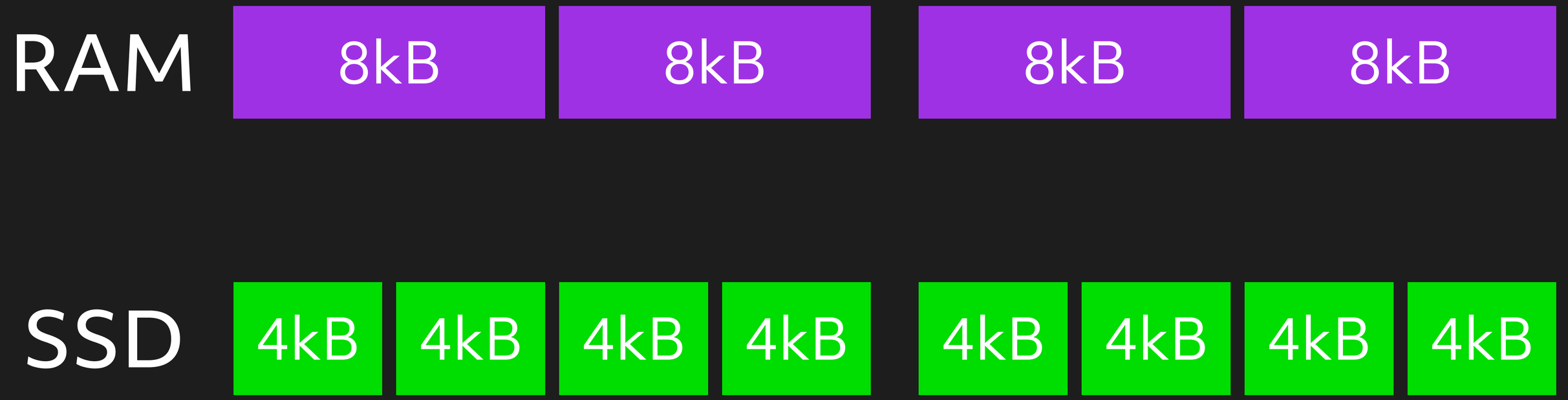
fsync pseudocode

```
file = File.open("/data/base")
file.write("some data")
file.fsync
```

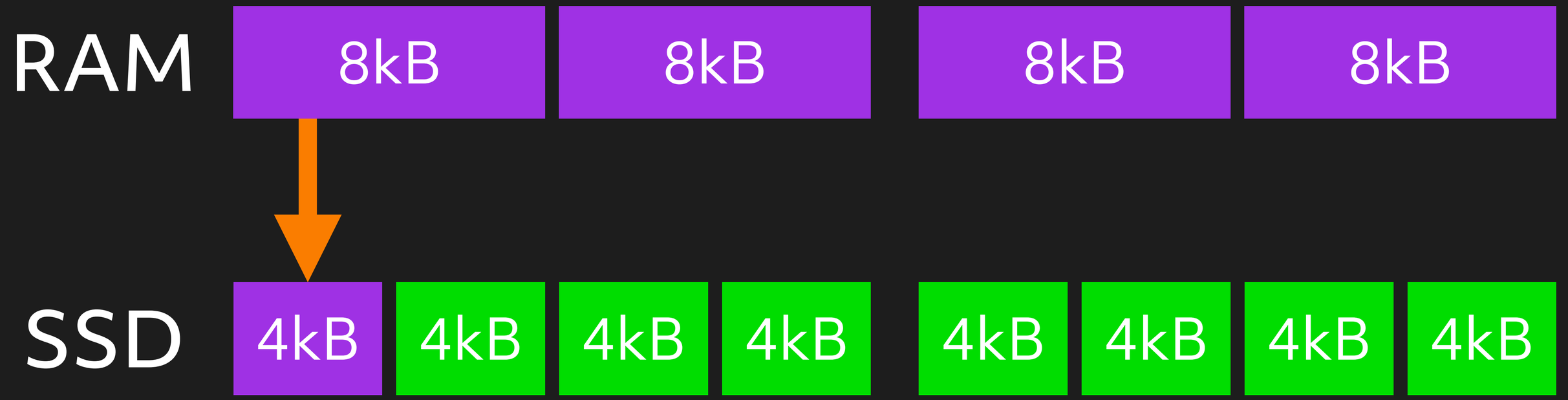
fsync pseudocode

```
file = File.open("/data/base")
file.write("some data")
err = file.fsync

# Mark the data for a retry
if !err.nil? && err.type == EIO
  file.mark_for_retry
end
```

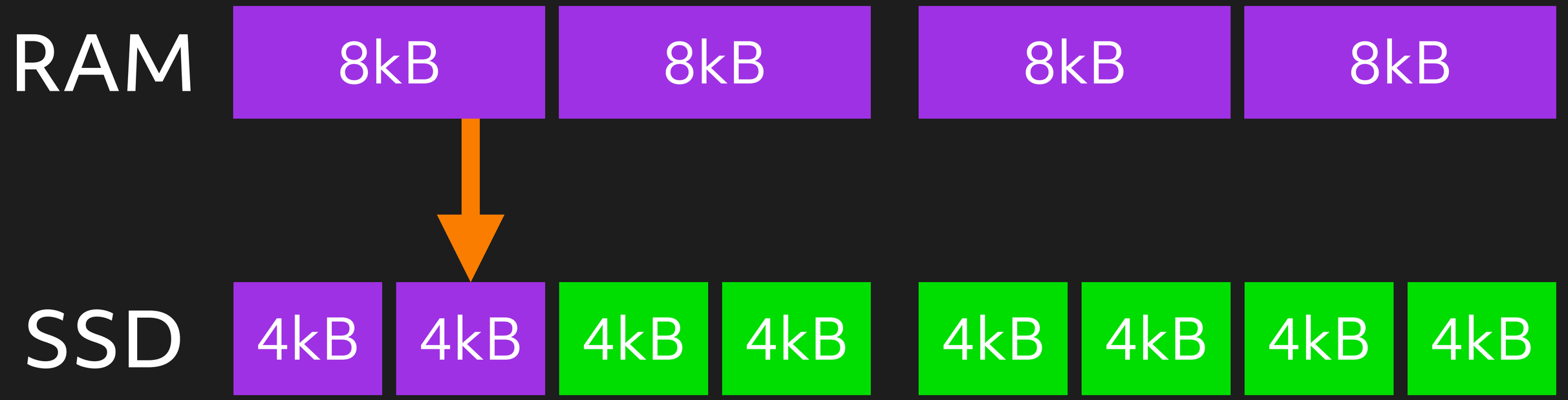
fsync



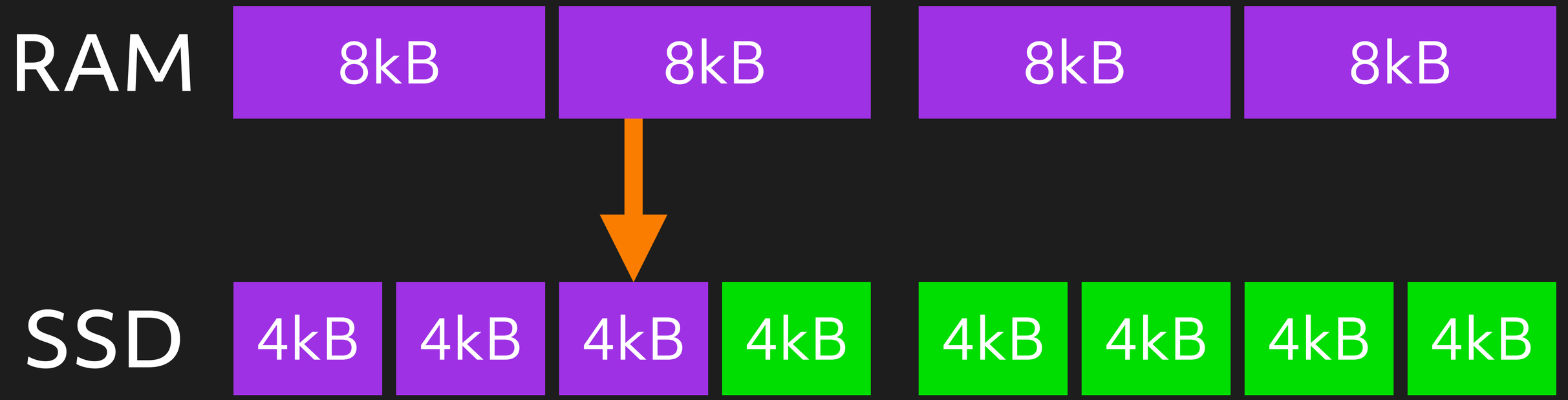
fsync



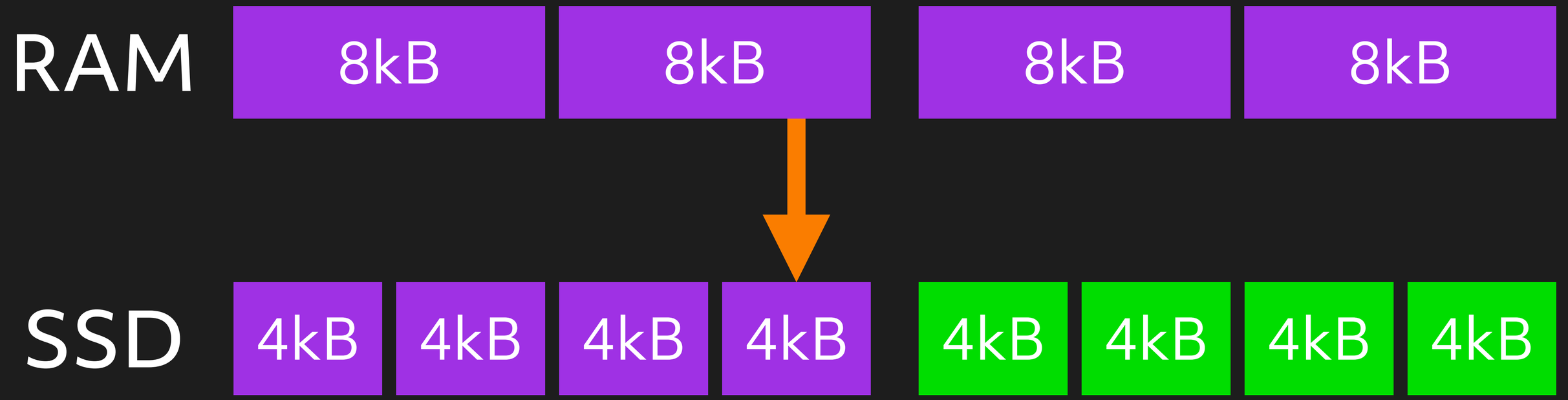
fsync



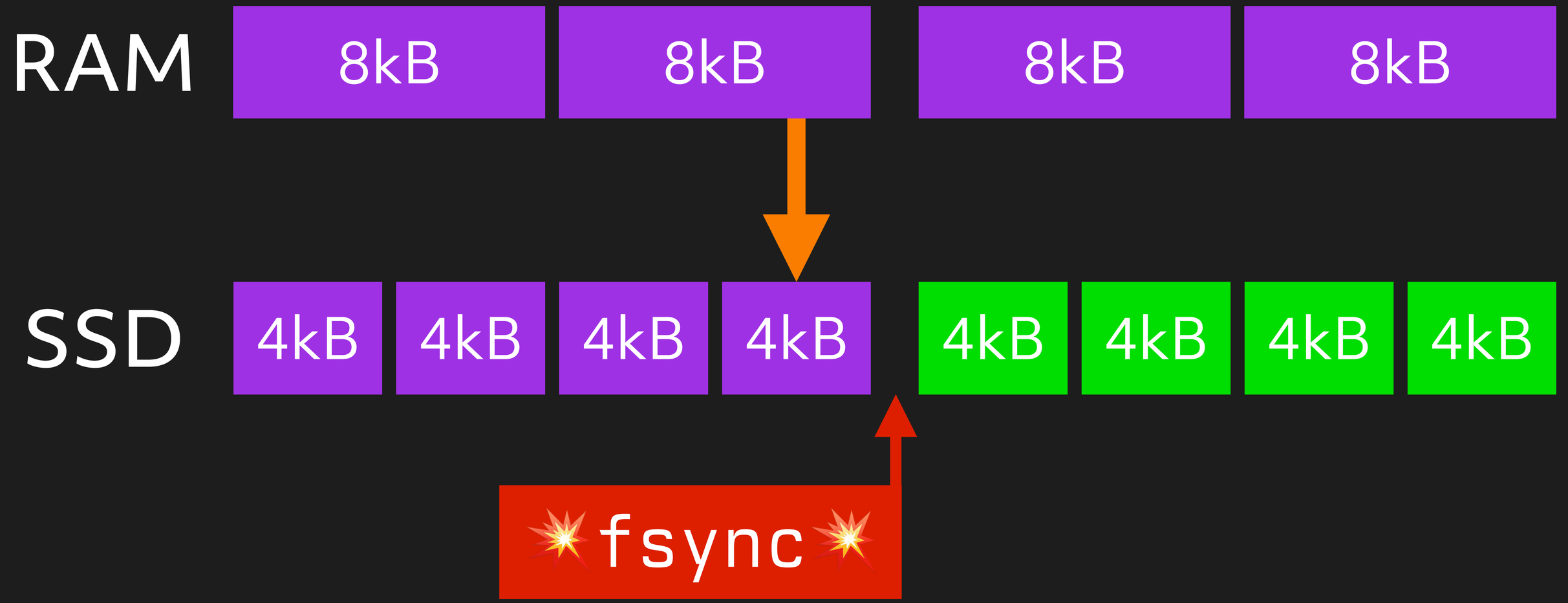
fsync



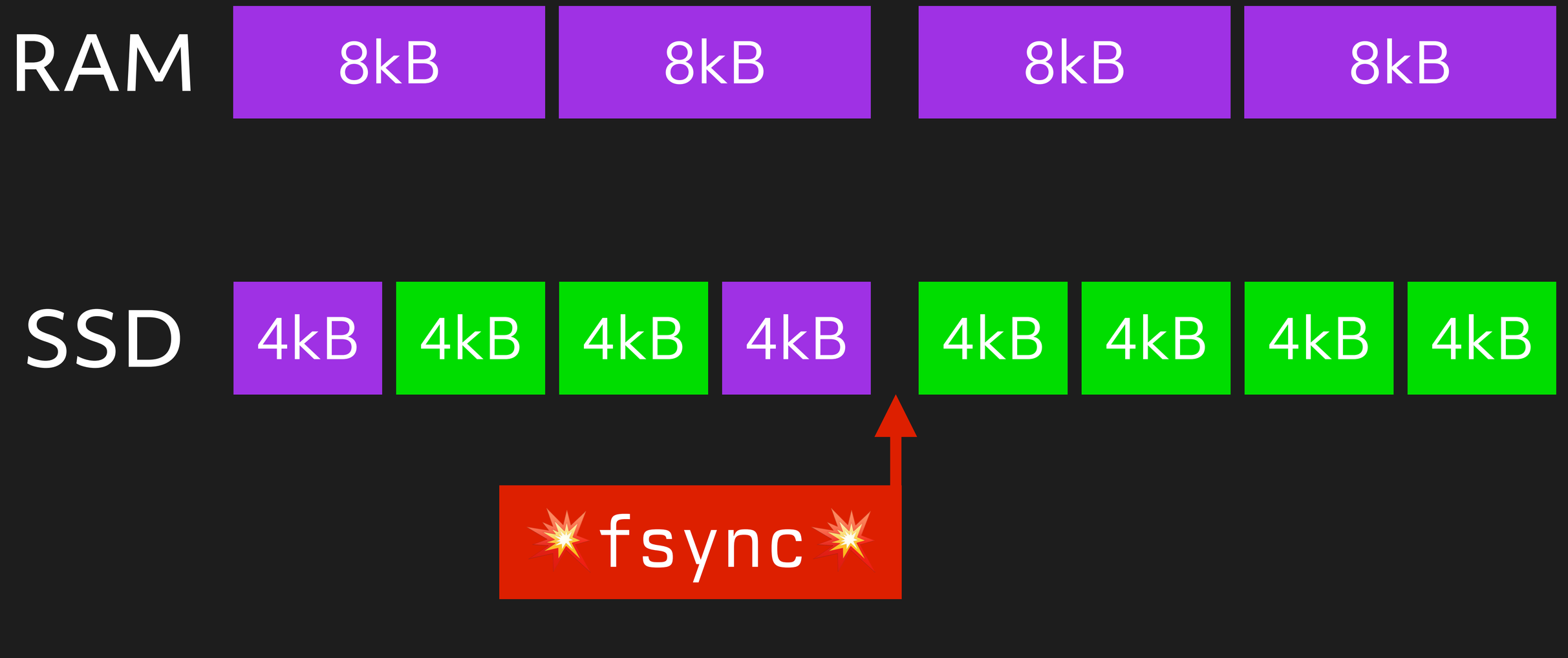
fsync



fsync

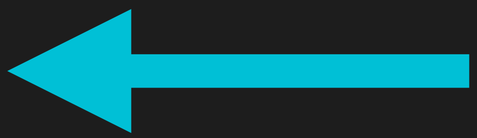


fsync

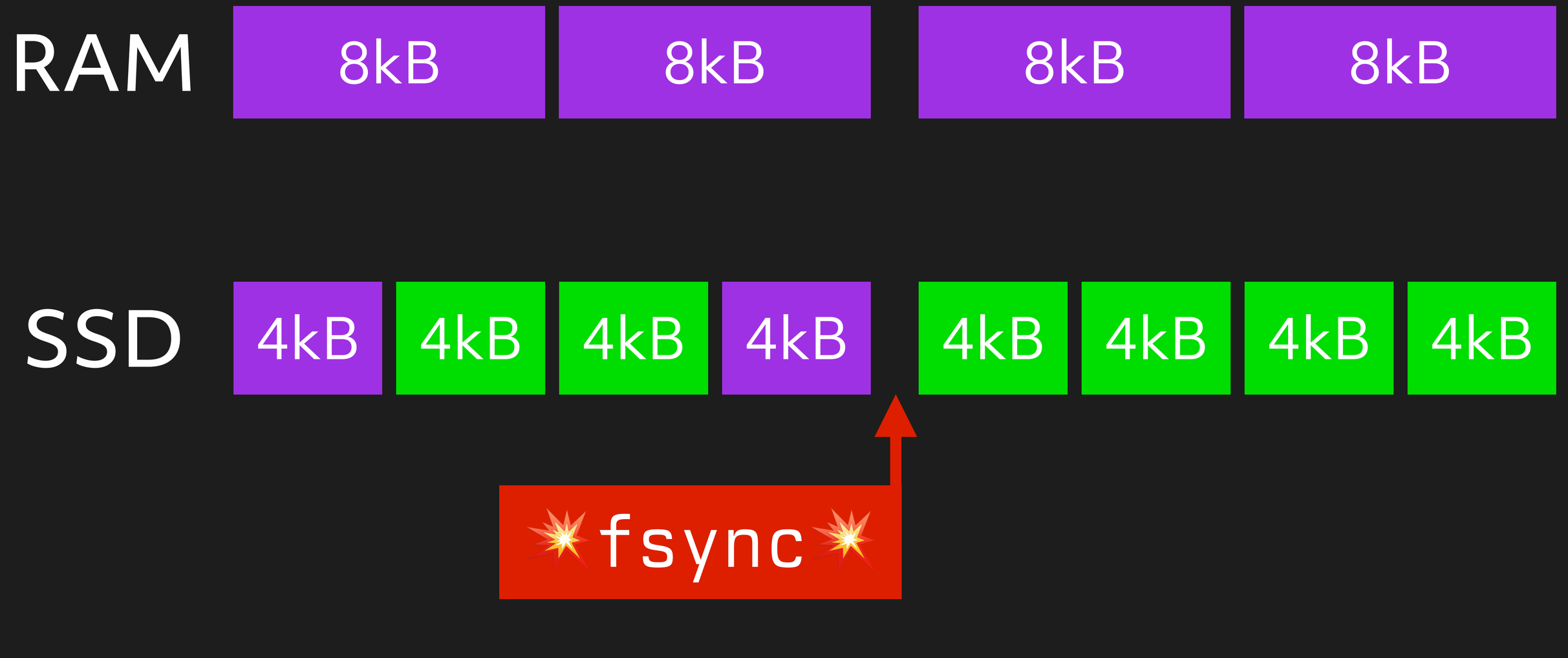


fsync pseudocode

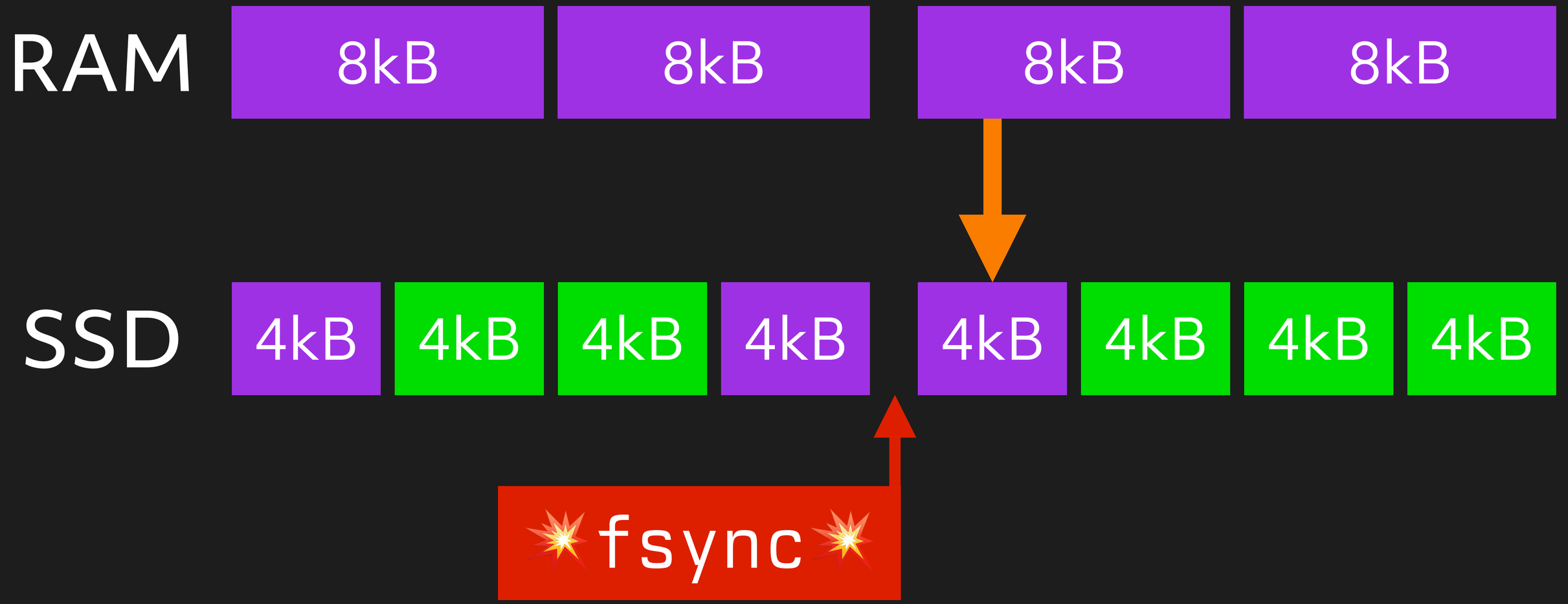
```
file = File.open("/data/base")
file.write("some data")
err = file.fsync

# Mark the data for a retry
if !err.nil? && err.type == EIO
  file.mark_for_retry
end
```



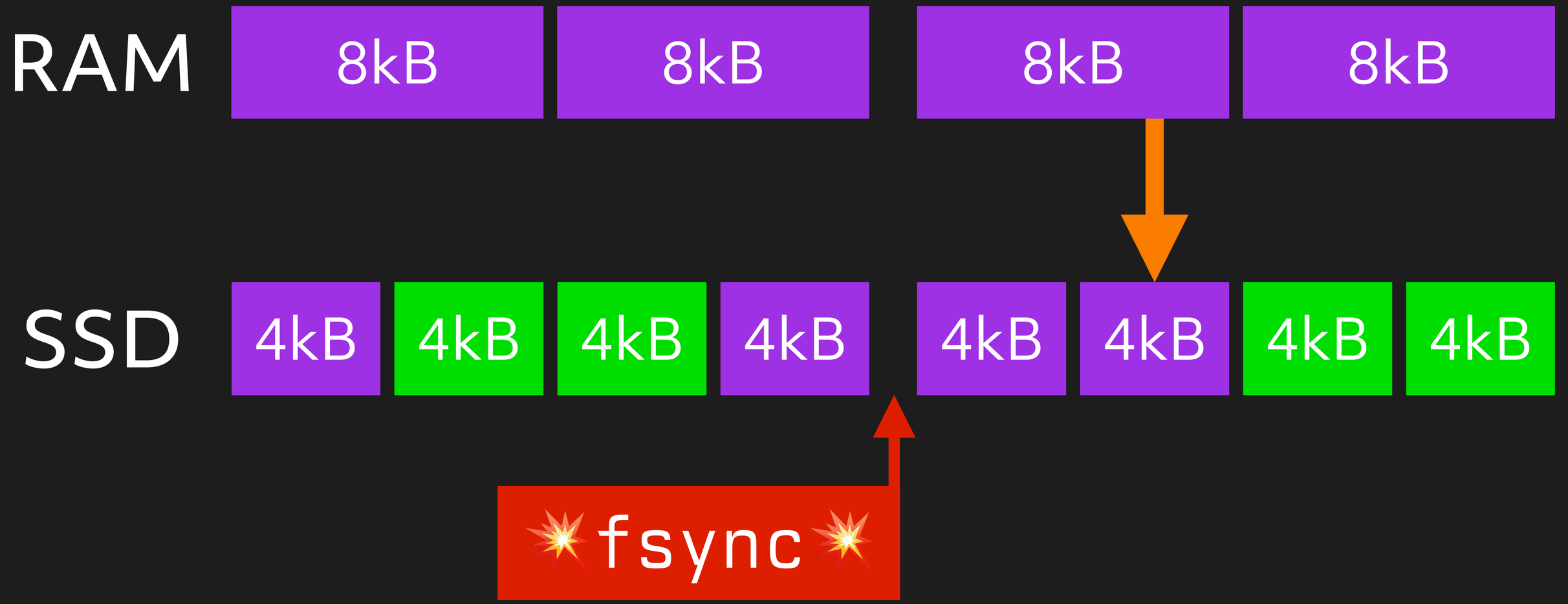
fsync



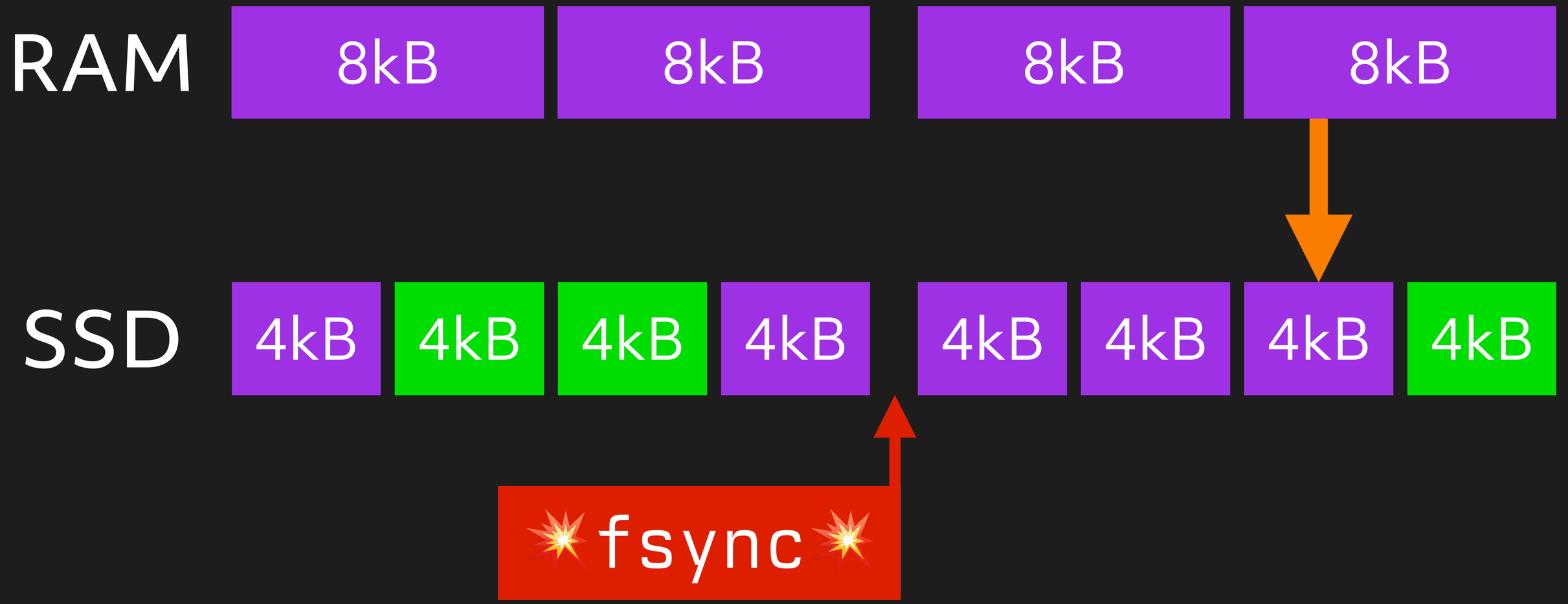
fsync



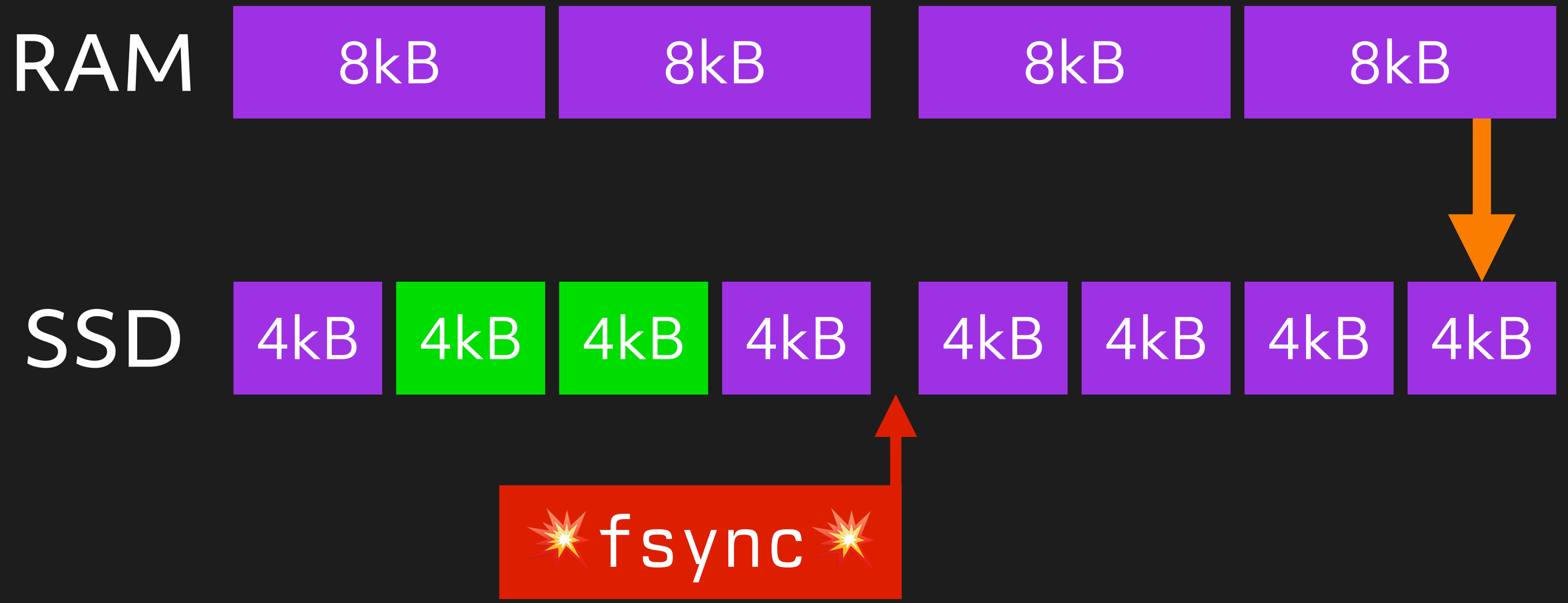
fsync



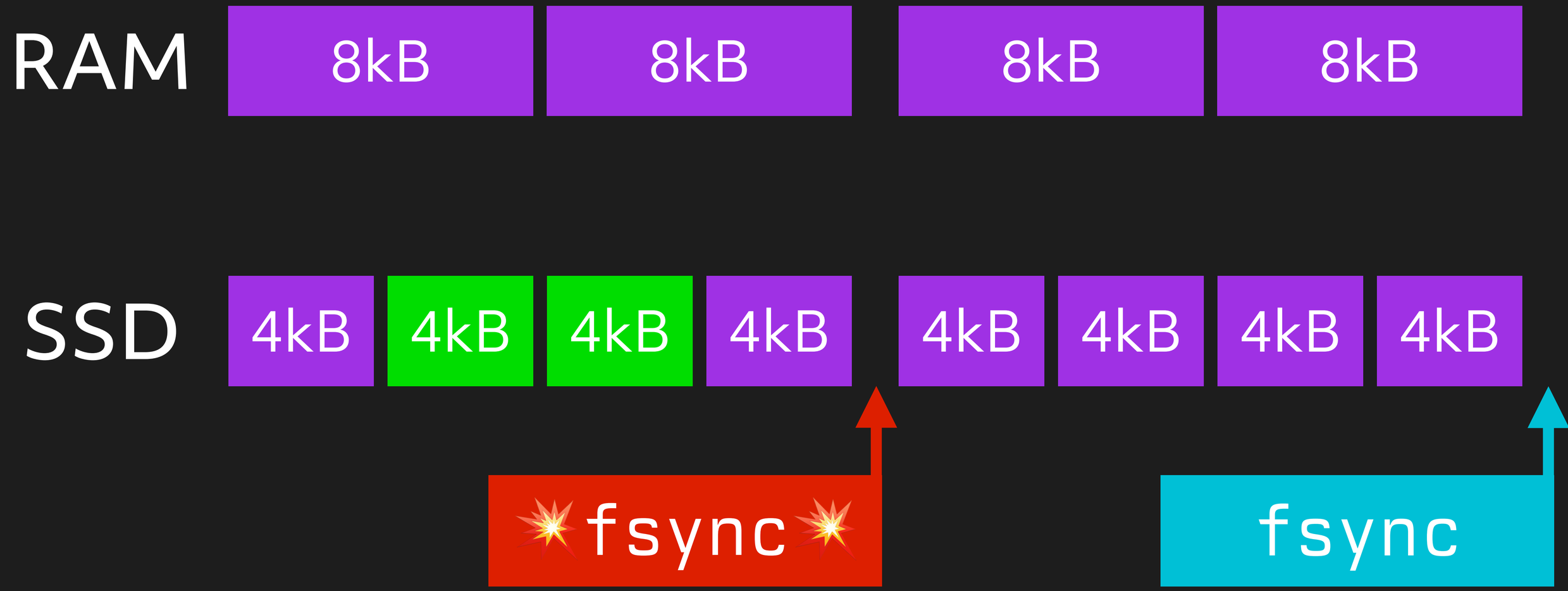
fsync



fsync



fsync


RAM
8kB
8kB
8kB
8kB
SSD
4kB
4kB
4kB
4kB
4kB
4kB
4kB
4kB
fsync
fsync

Postgres mailing list

From: Craig Ringer <craig(at)2ndquadrant(dot)com>
To: PostgreSQL Hackers <pgsql-hackers(at)postgresql(dot)org>
Subject: PostgreSQL's handling of fsync() errors is unsafe and risks data loss at least on XFS
Date: 2018-03-28 02:23:46
Message-ID: CAMsr+YHh+5Oq4xziwwoEfhoTZgr07vdGG+hu=1adXx59aTeaoQ@mail.gmail.com
Lists: pgsql-hackers

Hi all

Some time ago I ran into an issue where a user encountered data corruption after a storage error. PostgreSQL played a part in that corruption by allowing checkpoint what should've been a fatal error.

TL;DR: Pg should PANIC on fsync() EIO return. Retrying fsync() is not OK at least on Linux. When fsync() returns success it means "all writes since the last fsync have hit disk" but we assume it means "all writes since the last SUCCESSFUL fsync have hit disk".

...

2018-03-28 02:23:46 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-28 03:53:08 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
2018-03-29 02:30:59 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-03-29 02:48:27 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 05:00:31 from Justin Pryzby <pryzby(at)telsasoft(dot)com>
2018-03-29 05:06:22 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 05:25:51 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-21 19:21:39 from Gasper Zejn <zejn(at)owca(dot)info>
2018-03-29 05:32:43 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 05:35:47 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 05:58:45 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 12:07:56 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-29 13:15:10 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-29 16:20:00 from Catalin Iacob <iacobcatalin(at)gmail(dot)com>
2018-03-29 21:18:14 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-03-31 13:24:28 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-03-31 16:13:09 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-03-31 16:38:12 from Tom Lane <tgl(at)sss(dot)pgh(dot)pa(dot)us>
2018-04-01 00:20:38 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-01 01:14:46 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 18:13:46 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 18:53:20 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 19:32:45 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 20:38:06 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 20:58:08 from Stephen Frost <sfrost(at)snowman(dot)net>
2018-04-02 23:05:44 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-02 23:23:24 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-02 23:27:35 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-03 00:03:39 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-03 00:05:09 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-03 00:07:41 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-03 00:48:00 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-03 02:54:26 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-03 03:45:30 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-03 10:35:39 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 11:26:05 from Greg Stark <stark(at)mit(dot)edu>
2018-04-03 13:36:47 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 14:37:30 from Greg Stark <stark(at)mit(dot)edu>
2018-04-03 16:52:07 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-03 21:47:01 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-04 00:56:37 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 01:54:50 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 02:05:19 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 02:14:28 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 02:44:22 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 05:29:28 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 06:00:21 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 07:32:04 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 13:49:38 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 15:23:51 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 17:51:03 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-05 23:37:42 from Andrew Gierth <andrew(at)tao11(dot)riddles(dot)org(dot)uk>
2018-04-06 01:27:05 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-06 02:53:56 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-06 03:20:22 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 02:16:07 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-08 02:33:37 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 02:37:47 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 03:27:45 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 03:37:06 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-08 03:46:17 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 10:30:31 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 16:38:03 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 22:29:16 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 23:10:24 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-08 23:16:25 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-08 23:27:57 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-09 01:55:10 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 02:00:41 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 02:06:12 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 03:15:01 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 13:54:19 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 01:35:06 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 13:42:35 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 13:47:03 from Abhijit Menon-Sen <ams(at)2ndQuadrant(dot)com>
2018-04-09 18:02:21 from Gasper Zejn <zejn(at)owca(dot)info>
2018-04-17 21:29:17 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:34:53 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-18 09:52:22 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:32:45 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-17 21:41:42 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-18 11:56:57 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-08 09:41:06 from Andreas Karlsson <andreas(at)proxel(dot)se>
2018-04-08 10:31:24 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 21:23:21 from Greg Stark <stark(at)mit(dot)edu>
2018-04-08 21:28:43 from Christophe Pettus <xof(at)thebuild(dot)com>
2018-04-09 01:31:56 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-08 21:47:04 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 08:45:40 from Greg Stark <stark(at)mit(dot)edu>
2018-04-09 10:50:41 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 12:03:28 from Geoff Winkless <pgsqladmin(at)geoff(dot)dj>
2018-04-09 12:31:27 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 13:33:18 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 14:22:06 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 15:29:36 from Greg Stark <stark(at)mit(dot)edu>
2018-04-09 19:26:21 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:29:16 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:44:31 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:37:03 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:51:12 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:54:05 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:04:20 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 20:30:00 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:37:31 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 01:59:03 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 02:00:59 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 16:54:40 from Greg Stark <stark(at)mit(dot)edu>
2018-04-10 18:58:37 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-10 19:51:01 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-10 20:57:34 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-11 12:23:49 from Greg Stark <stark(at)mit(dot)edu>
2018-04-17 21:19:53 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-18 10:04:30 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 11:46:15 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-18 12:45:53 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 23:31:50 from Mark Kirkwood <mark(dot)kirkwood(at)catalyst(dot)net(dot)nz>
2018-04-19 00:44:33 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-20 20:49:08 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-09 19:47:44 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 22:33:16 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-10 00:32:20 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-09 12:16:38 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-09 12:54:16 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-09 19:41:19 from Justin Pryzby <pryzby(at)telsasoft(dot)com>
2018-04-09 19:59:34 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-10 01:44:59 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 01:52:21 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-09 16:45:00 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-09 17:26:24 from "Joshua D(dot) Drake" <jd(at)commandprompt(dot)com>
2018-04-09 18:29:42 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-09 19:22:58 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 19:02:11 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-09 19:13:14 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 19:25:33 from Peter Geoghegan <pg(at)bowt(dot)ie>
2018-04-17 21:49:42 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-09 20:25:54 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-09 20:34:15 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 20:43:03 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 20:55:29 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-09 21:08:29 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-09 21:25:52 from Tomas Vondra <tomas(dot)vondra(at)2ndquadrant(dot)com>
2018-04-09 21:33:29 from Mark Dilger <hornschnorter(at)gmail(dot)com>
2018-04-10 01:54:30 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 15:15:46 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-10 15:40:05 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-10 16:38:27 from Greg Stark <stark(at)mit(dot)edu>
2018-04-11 12:05:27 from Jonathan Corbet <corbet(at)lwn(dot)net>
2018-04-11 14:29:09 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-11 14:40:31 from Jonathan Corbet <corbet(at)lwn(dot)net>
2018-04-10 05:04:13 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-10 05:37:19 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 06:10:21 from Michael Paquier <michael(at)paquier(dot)xyz>
2018-04-10 12:15:15 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-18 10:19:28 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-10 15:28:07 from Robert Haas <robertmhaas(at)gmail(dot)com>
2018-04-10 00:41:10 from Andreas Karlsson <andreas(at)proxel(dot)se>
2018-04-10 02:02:48 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 17:23:58 from Gasper Zejn <zejn(at)owca(dot)info>
2018-04-04 07:51:53 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 14:00:15 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 14:09:09 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 14:25:47 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-04 14:42:18 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 15:23:31 from Antonis Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-04 21:28:09 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 22:14:24 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-04 02:40:16 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-04 13:53:01 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-03 14:29:10 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-01 00:58:22 from Anthony Iliopoulos <ailiop(at)altatus(dot)com>
2018-04-01 18:24:51 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-02 15:03:42 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-03 01:29:28 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-03 23:59:27 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-05 07:09:57 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-05 08:46:08 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-05 19:33:14 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-23 20:14:48 from Andres Freund <andres(at)anarazel(dot)de>
2018-04-24 00:09:23 from Bruce Momjian <bruce(at)momjian(dot)us>
2018-04-27 01:18:55 from Thomas Munro <thomas(dot)munro(at)enterprisedb(dot)com>
2018-04-26 02:16:52 from Craig Ringer <craig(at)2ndquadrant(dot)com>
2018-04-30 04:55:22 from Craig Ringer <craig(at)2ndquadrant(dot)com>

“The Linux kernel is wrong”

– Many messages on that thread

“It should retry”

or

“It should persist the errors”

“It should retry”

or

“It should persist the errors”

“It should retry”

or

“It should persist the errors”

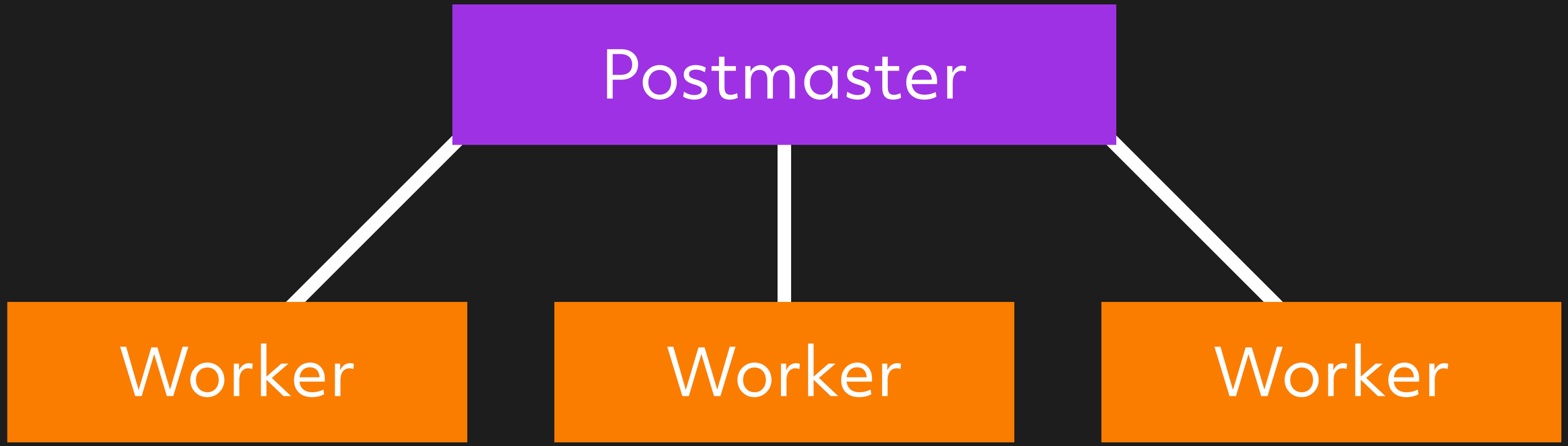

WD
My Passport
Ultra
USB 3.0
compatible with
USB 2.0

Postgres needs to run on the real kernel...

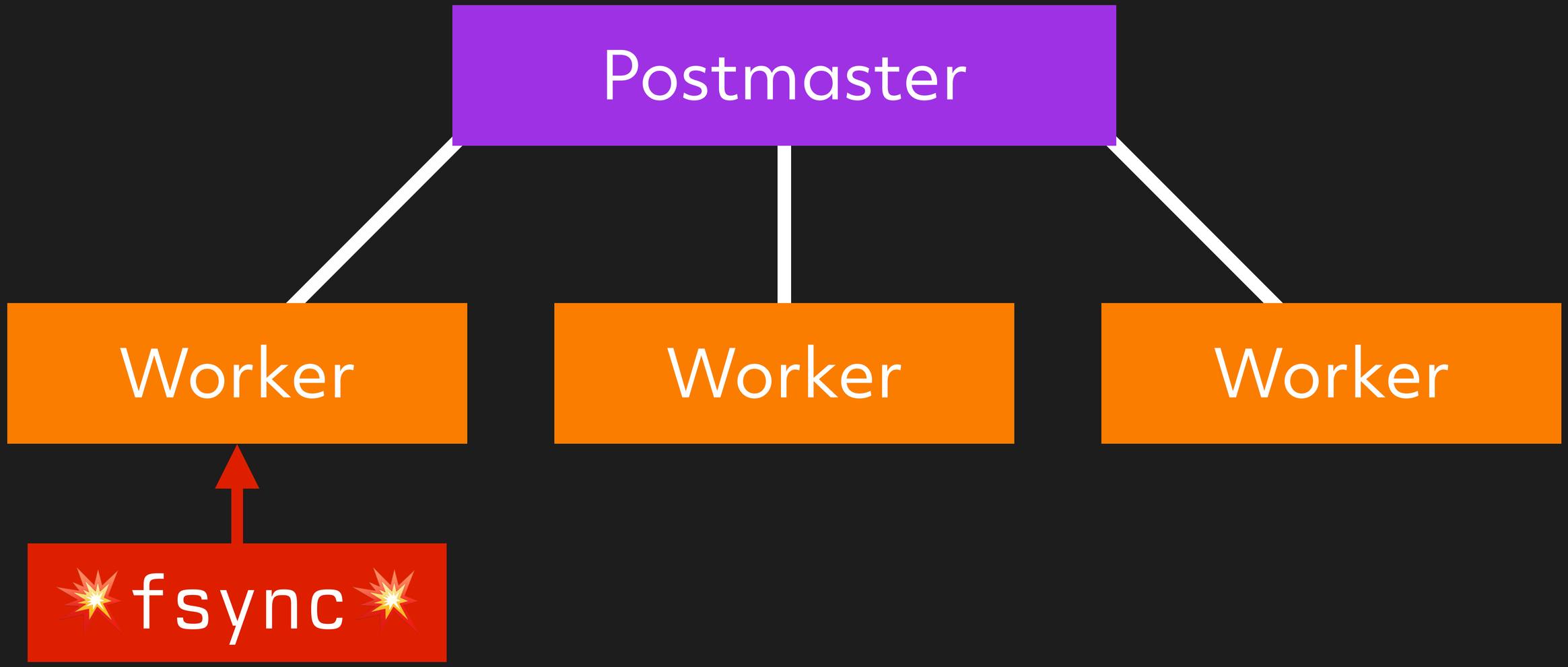
...not a hypothetical one

Sometimes
the right answer is
for software to
crash

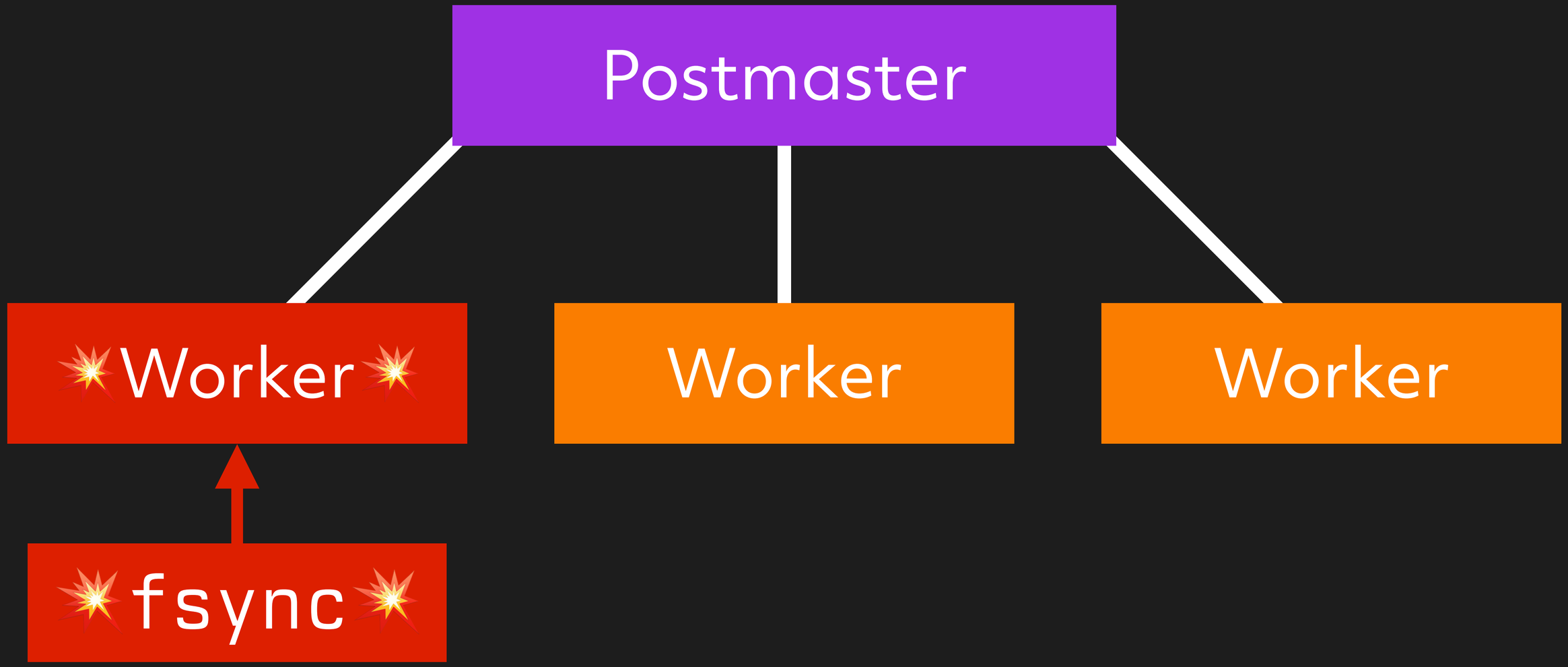
Postgres



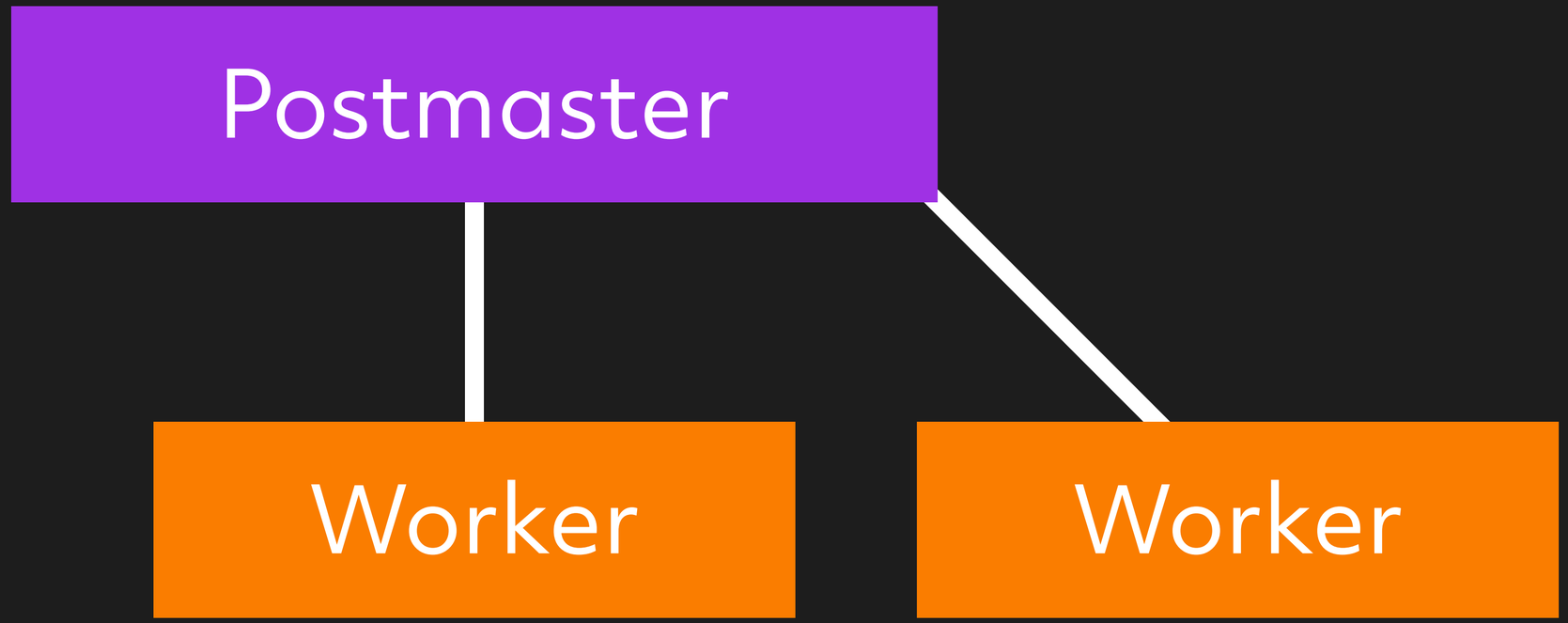
Postgres



Postgres



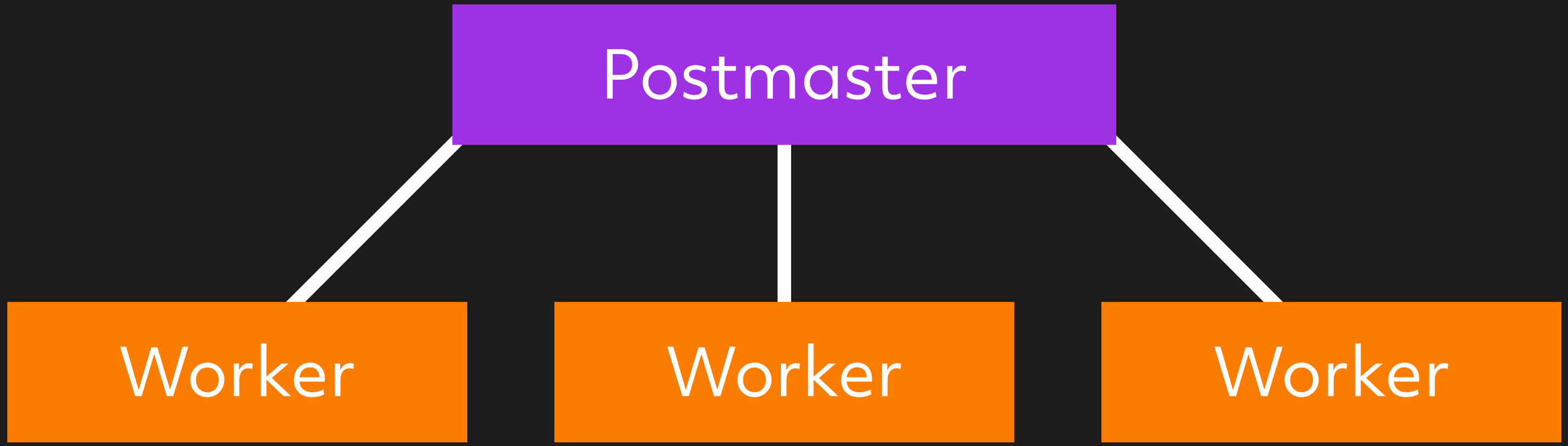
Postgres



Postgres

Postmaster

Postgres



SQL

```
BEGIN;
INSERT ("sunglasses", 27.99);
INSERT ("jorts", 10.99);
COMMIT;
```

WAL

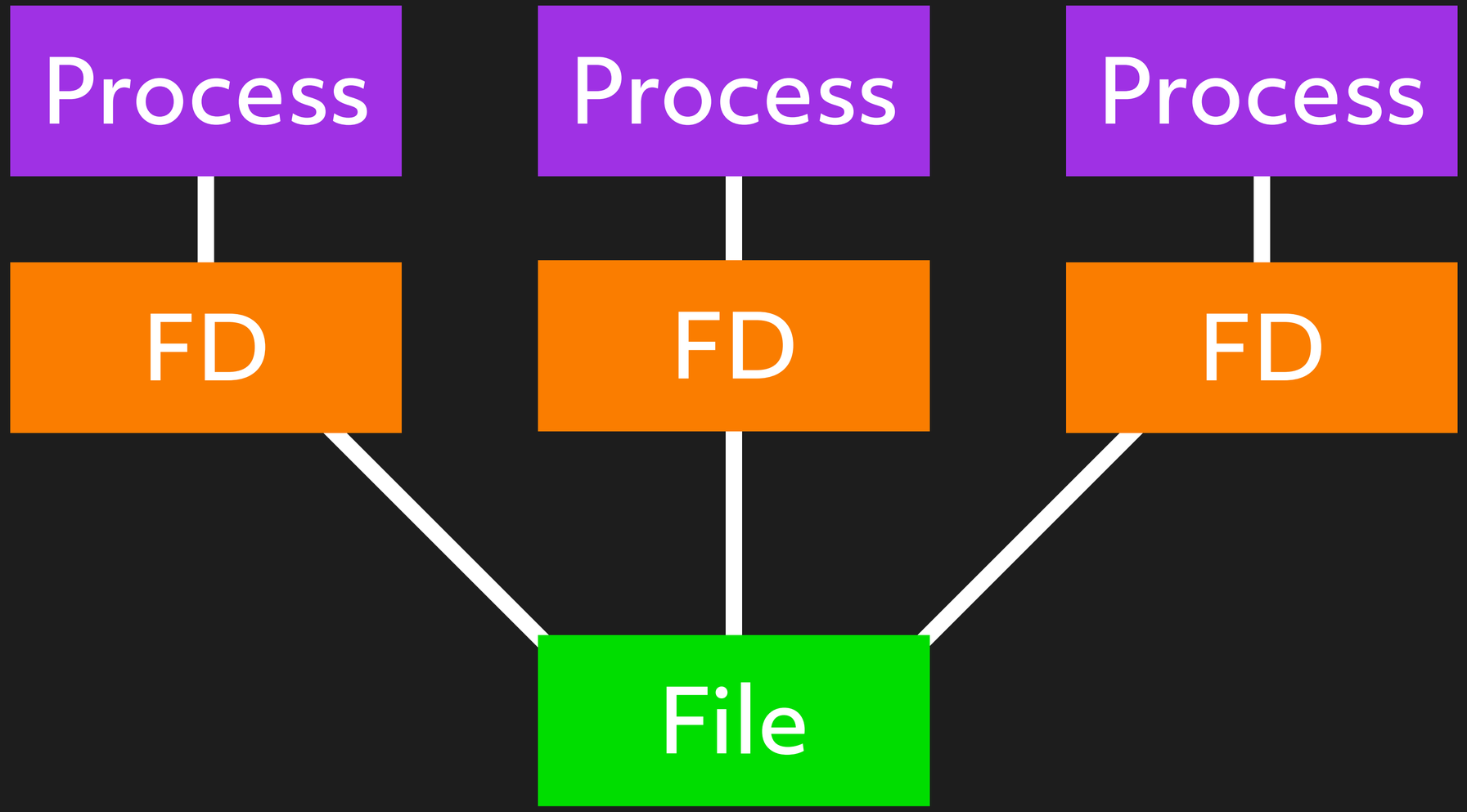
```
BEGIN TX 1
INS 1 (1, "sunglasses", 27.99)
INS 1 (2, "jorts", 10.99)
COMMIT TX 1
```

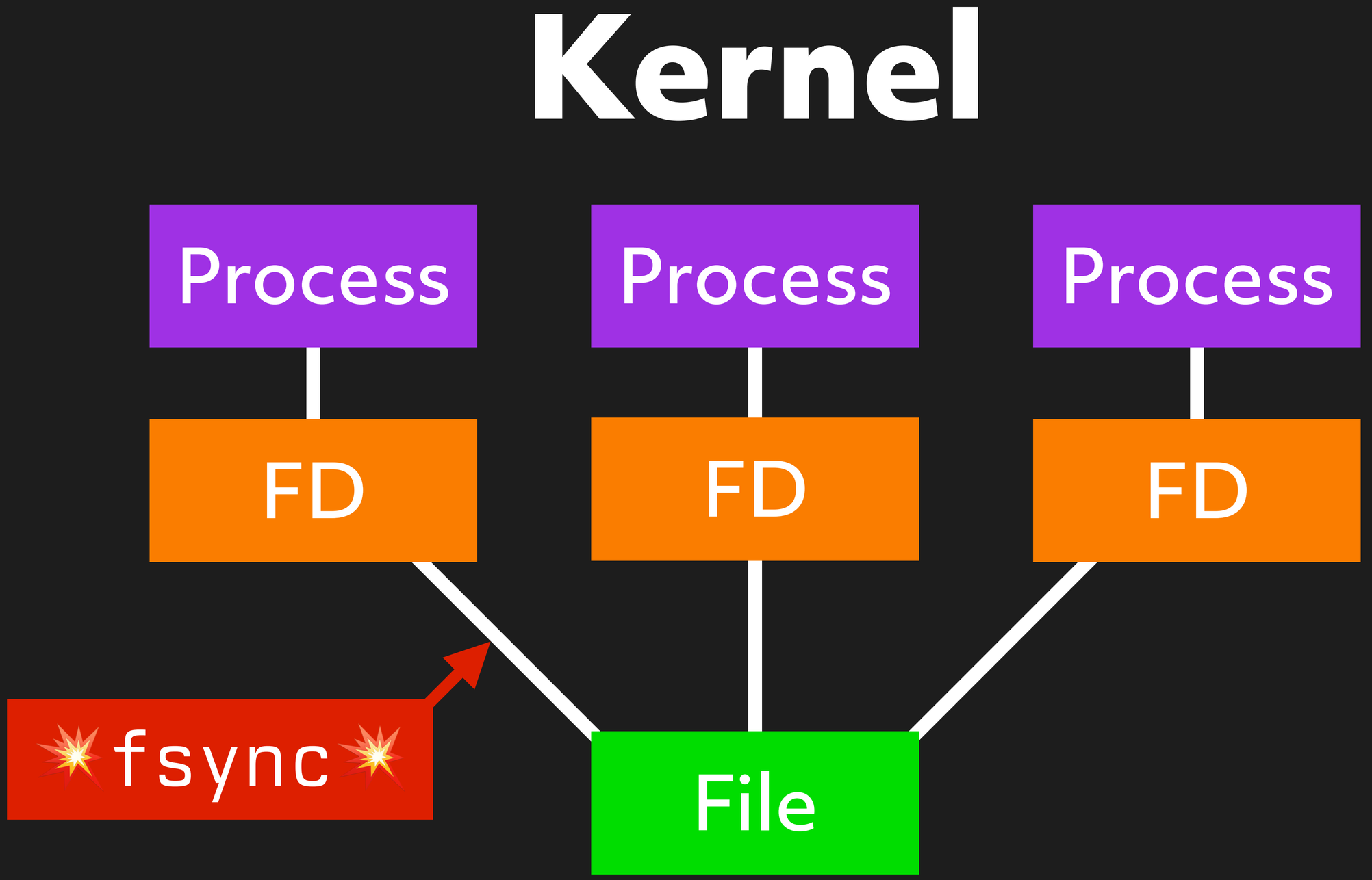
Table

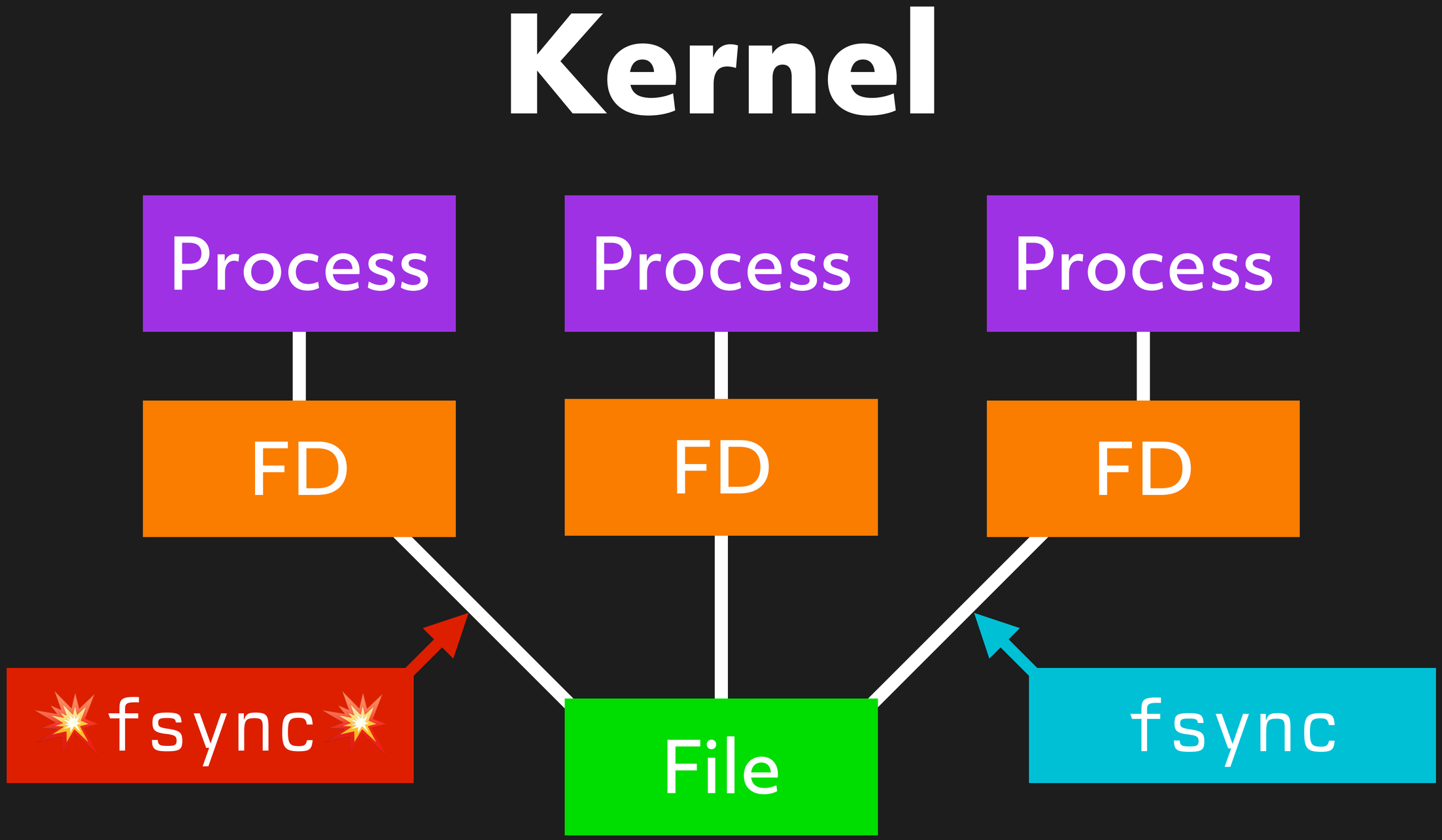
id	name	price
1	sunglasses	27.99
2	jorts	10.99

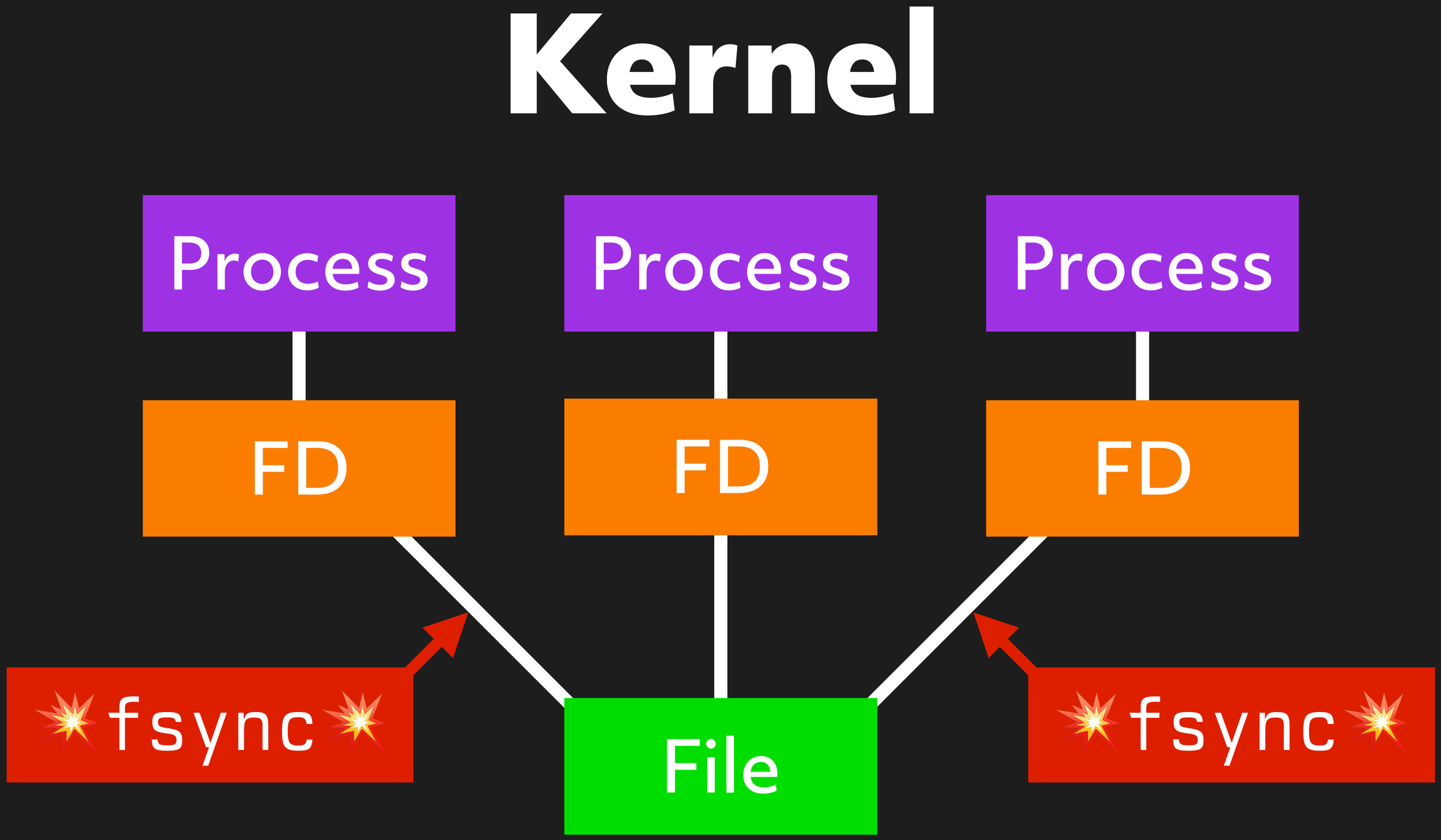
But the kernel did have a part to play...

Kernel

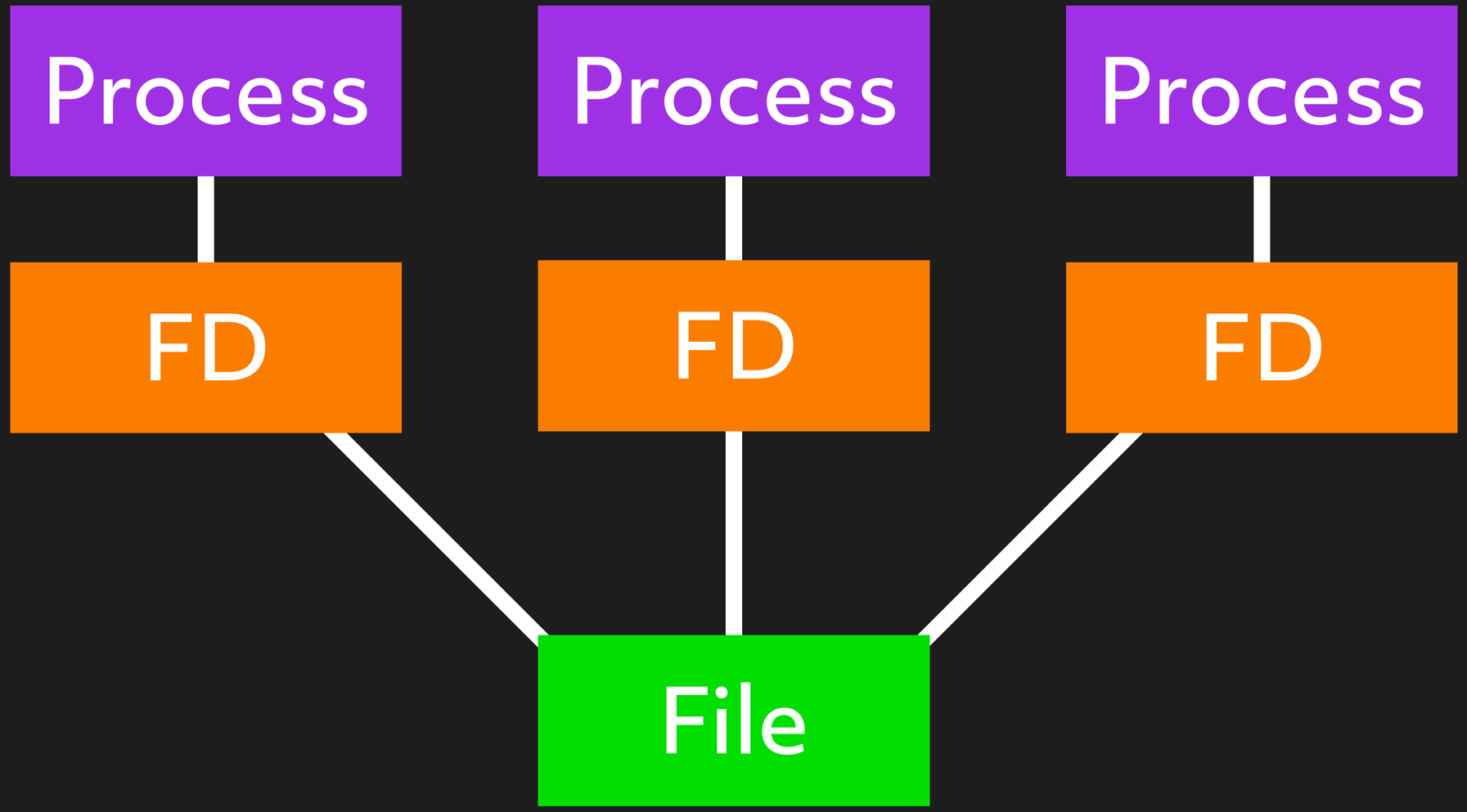


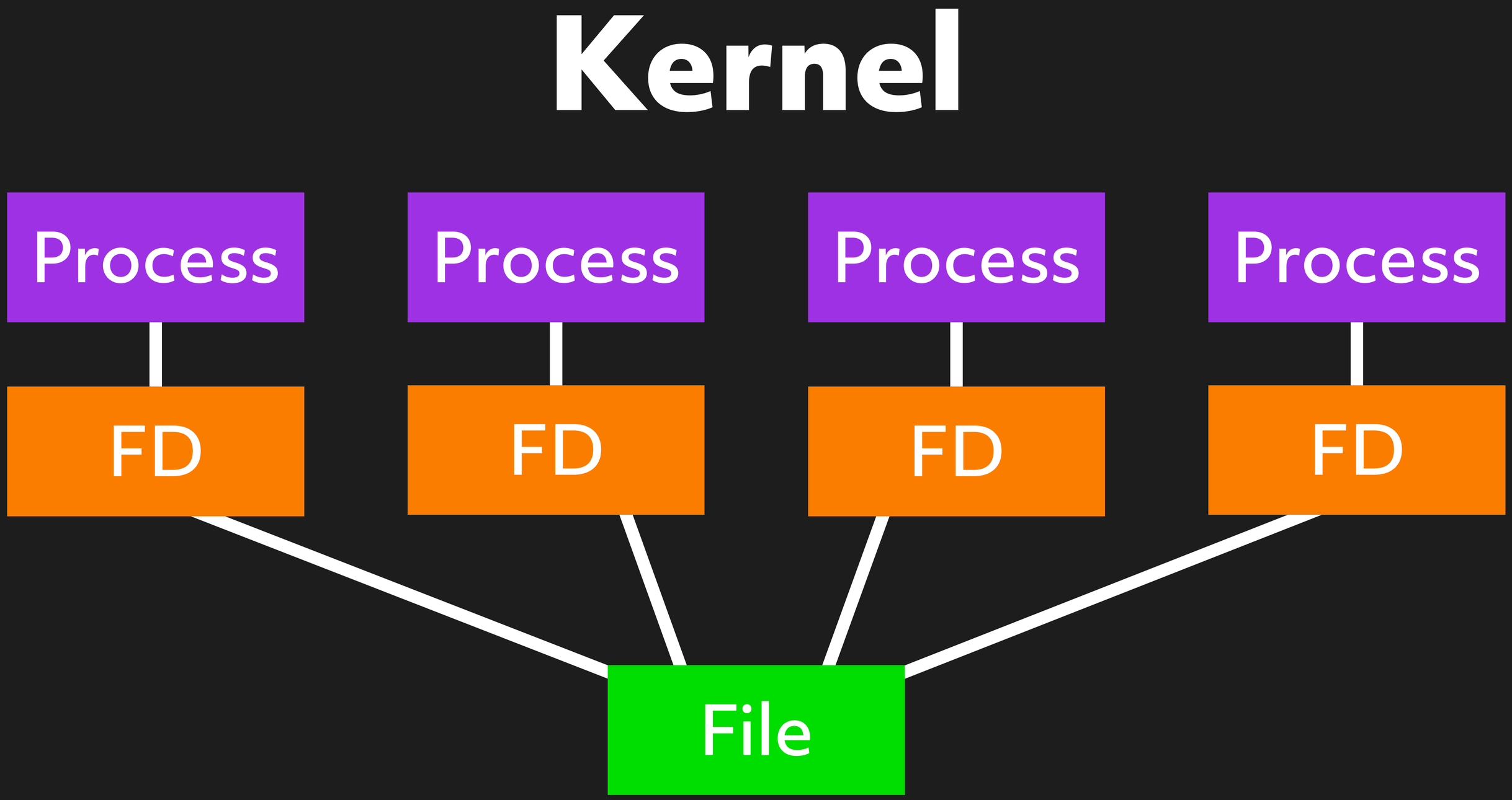

Kernel
Process
Process
Process
FD
FD
FD
fsync
File

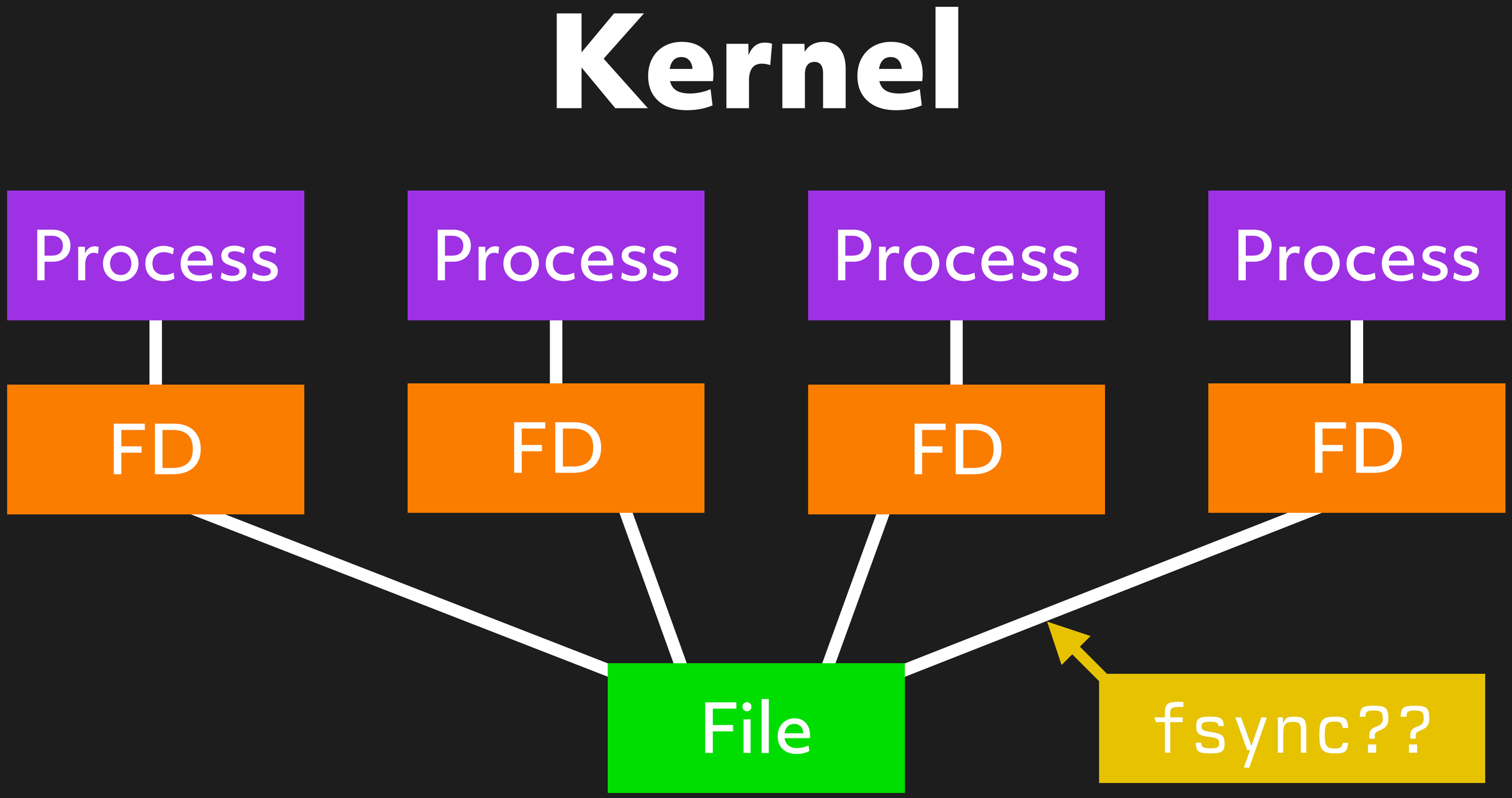

Kernel
Process
Process
Process
FD
FD
FD
fsync
File
fsync

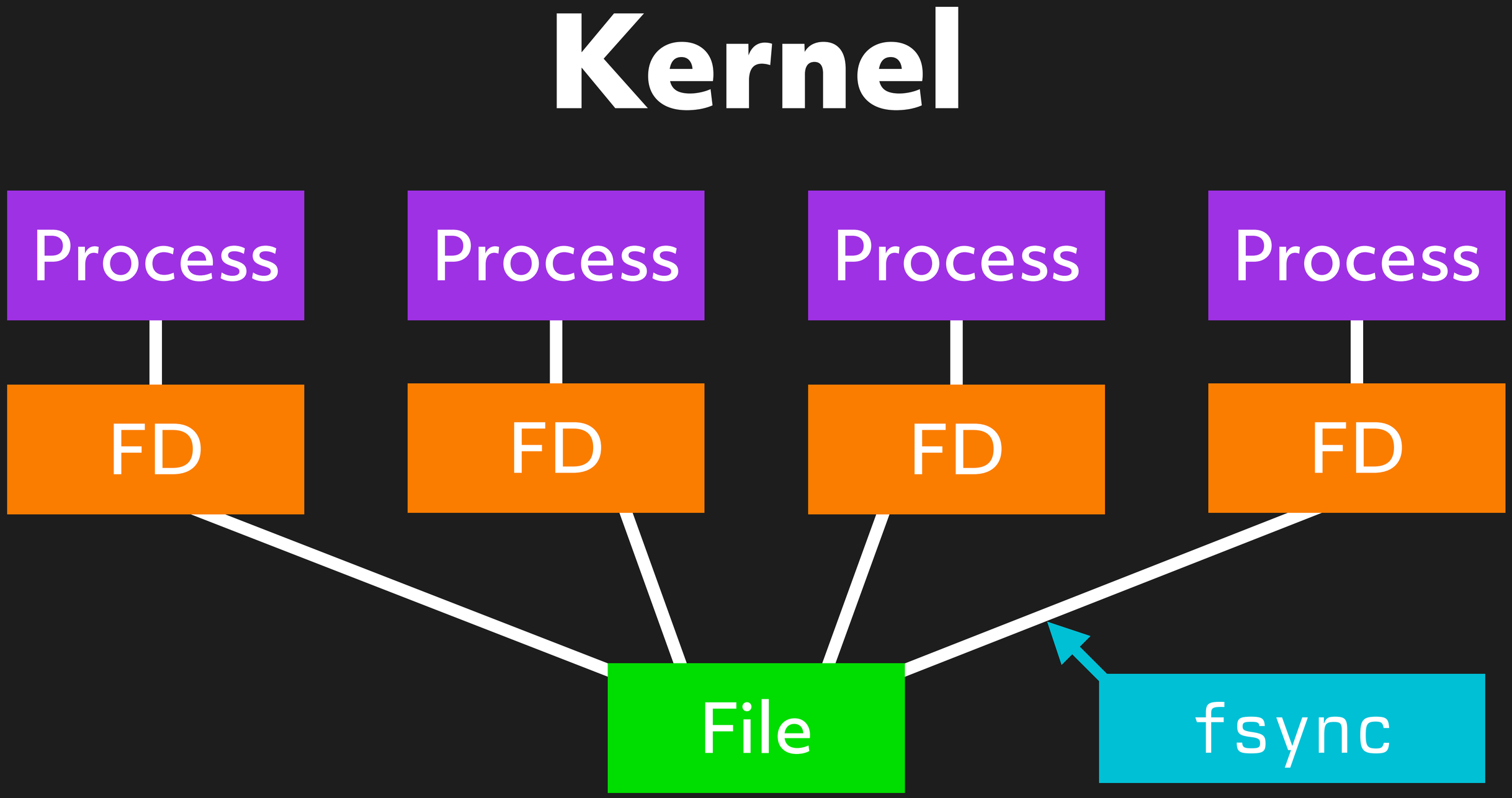

Kernel
Process
Process
Process
FD
FD
FD
fsync
File
fsync

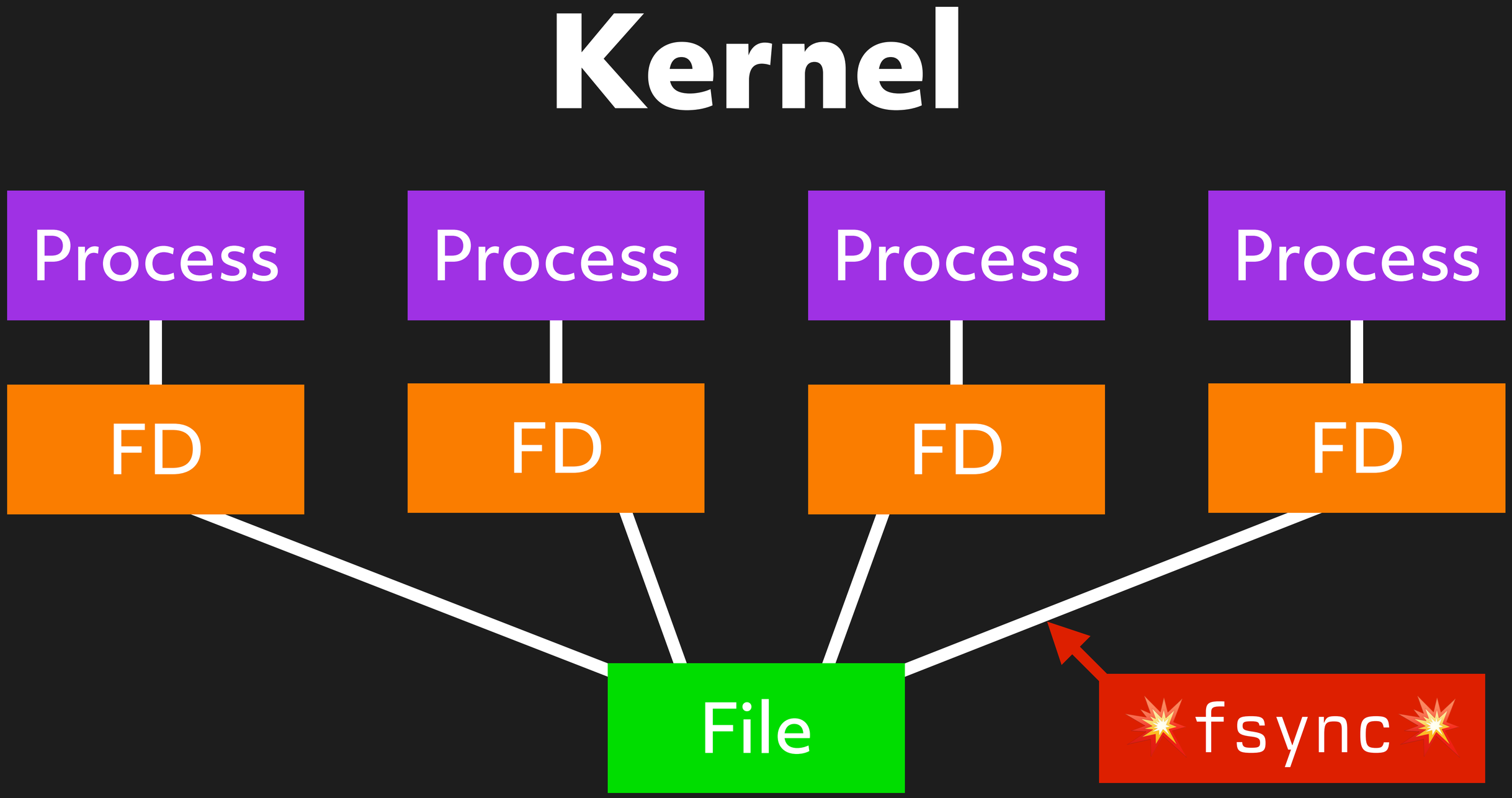
Kernel




Kernel
Process
Process
Process
Process
FD
FD
FD
FD
File


Kernel
Process
Process
Process
Process
FD
FD
FD
FD
File
fsync??

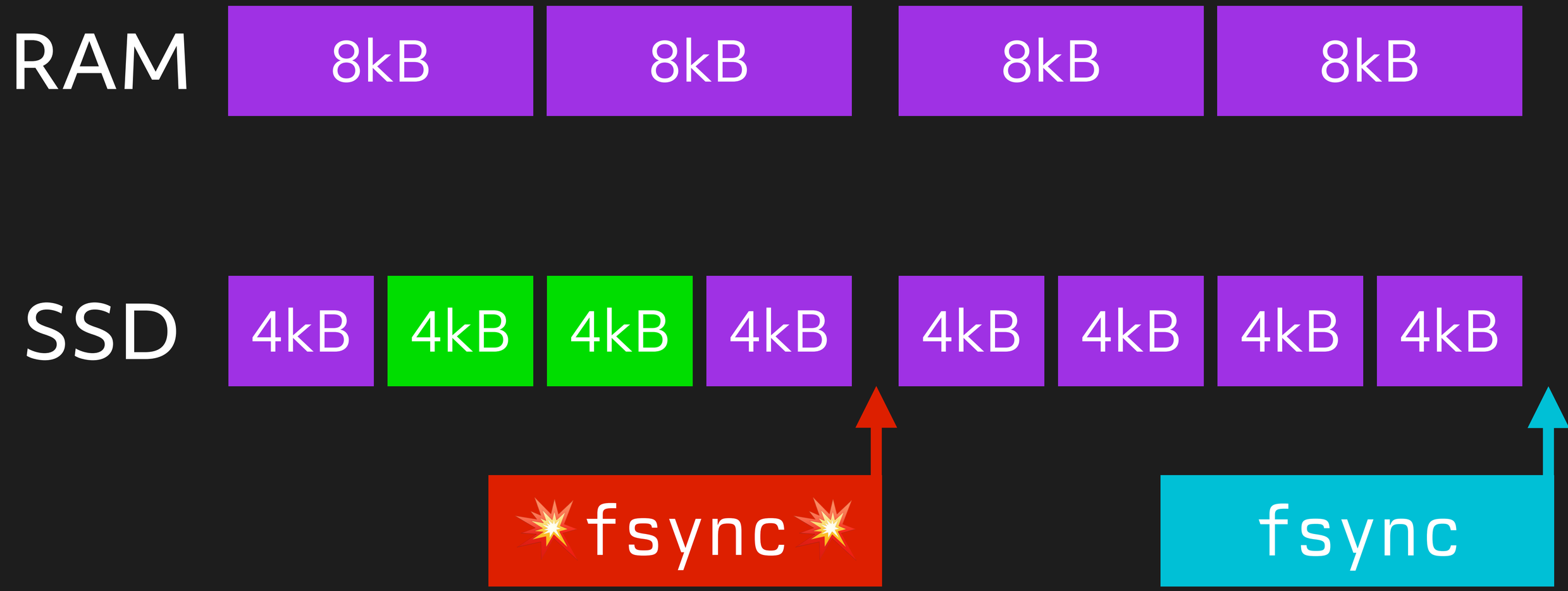

Kernel
Process
Process
Process
Process
FD
FD
FD
FD
File
fsync


Kernel
Process
Process
Process
Process
FD
FD
FD
FD
File
fsync

Kernel versions:

4.13 - 4.17

fsync


RAM
8kB
8kB
8kB
8kB
SSD
4kB
4kB
4kB
4kB
4kB
4kB
4kB
4kB
fsync
fsync

Other databases

- MySQL
- MongoDB
- ...?

Commit 8590c8e

Browse files

Allen Lai committed on Apr 8, 2018

```
Bug#27805553 HARD ERROR SHOULD BE REPORTED WHEN FSYNC() RETURN EIO.

fsync() will just return EIO only once when the IO error happens, so, it's
wrong to keep trying to call it till it return success.
When fsync() returns EIO it should be treated as a hard error and InnoDB must
abort immediately.
```

trunk · mysql-cluster-9.4.0 ··· mysql-5.7.23

1 parent 331797e commit 8590c8e

Filter files...

- storage/innobase/os
  - os0file.cc

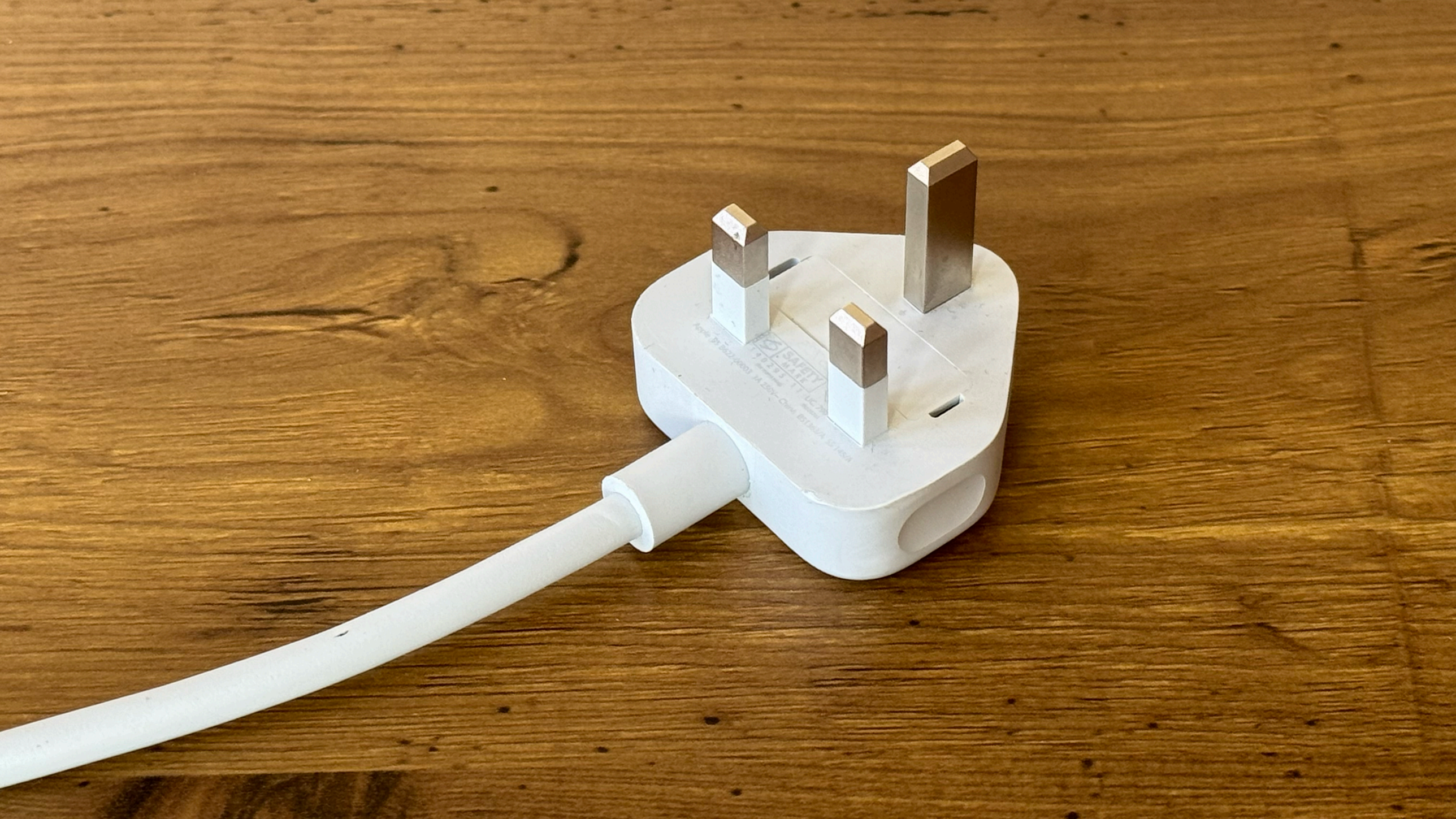
1 file changed +2 -15 lines changed

Search within code

storage/innobase/os/os0file.cc +2 -15

```
@@ -3079,21 +3079,8 @@ os_file_fsync_posix(
3079
3080            case EIO:
3081
3082  -             ++failures;
3083  -             ut_a(failures < 1000);
3084  -
3085  -             if (!(failures % 100)) {
3086  -
3087  -                 ib::warn()
3088  -                     << "fsync(): "
3089  -                     << "An error occurred during "
3090  -                     << "synchronization,"
3091  -                     << " retrying";
3092  -             }
3093  -
3094  -             /* 0.2 sec */
3095  -             os_thread_sleep(200000);
3096  -             break;
```

```
3079
3080            case EIO:
3081
3082  +                 ib::fatal()
3083  +                     << "fsync() returned EIO, aborting.";
```



Doing extra work to save your data...

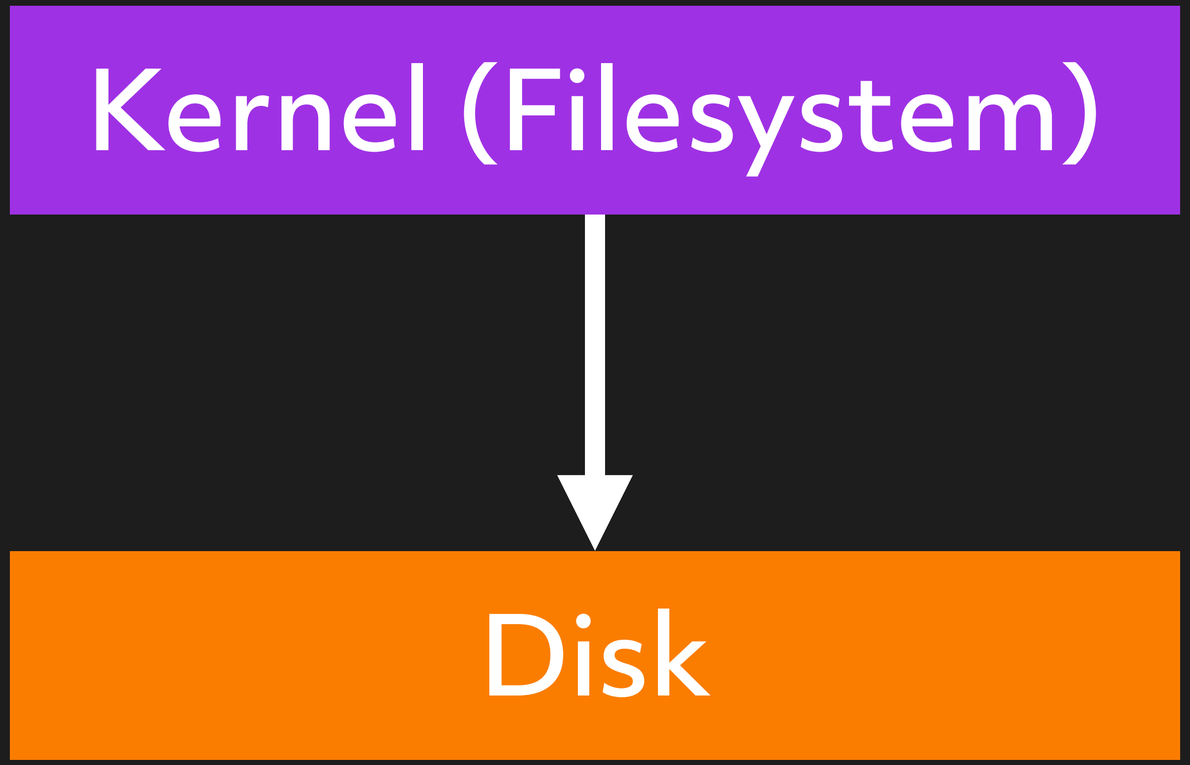
Doing extra work to save your data...

...can make the computer lose your data

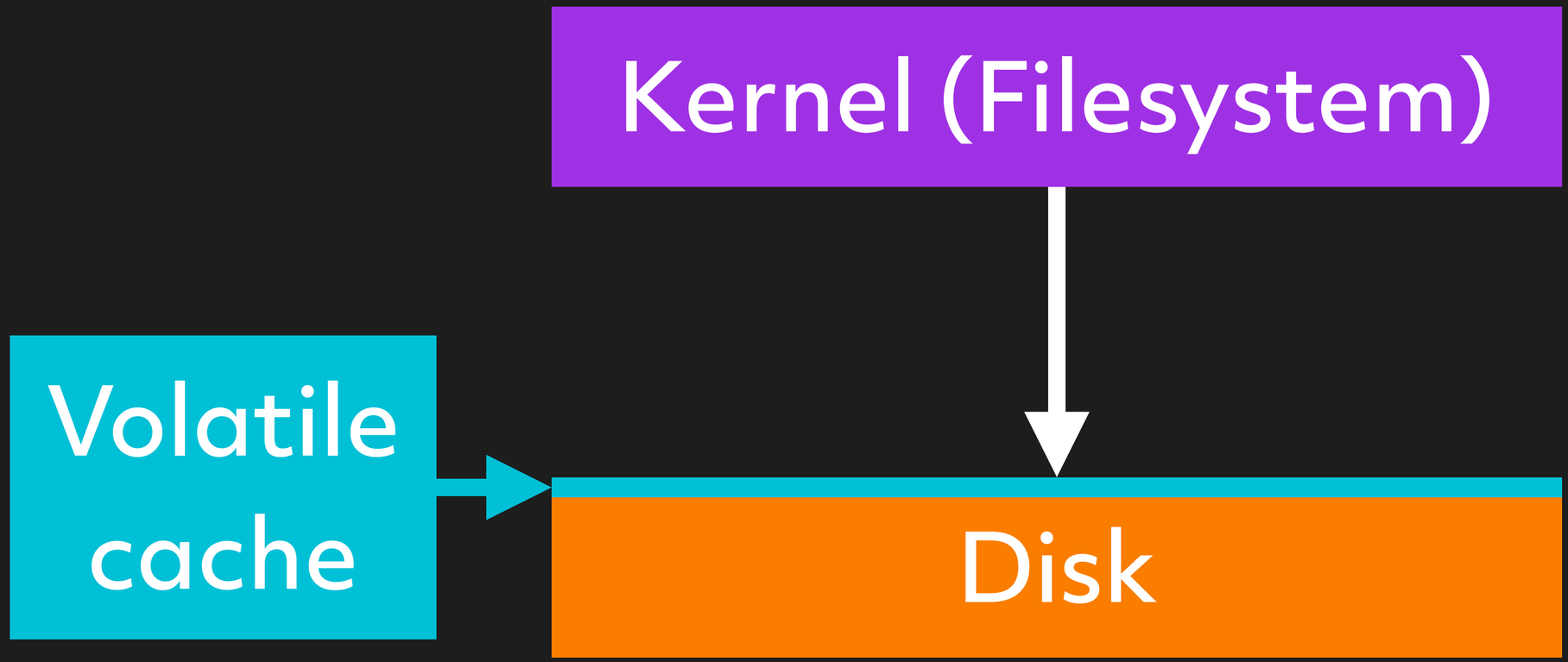
Case 3

Hardware disk caches

Hardware disk caches



Hardware disk caches



Hardware disk caches

Hardware disk caches

- Reduce latency

Disk operations

```
write("some data")
write("some more data")
write("yet more data")
```

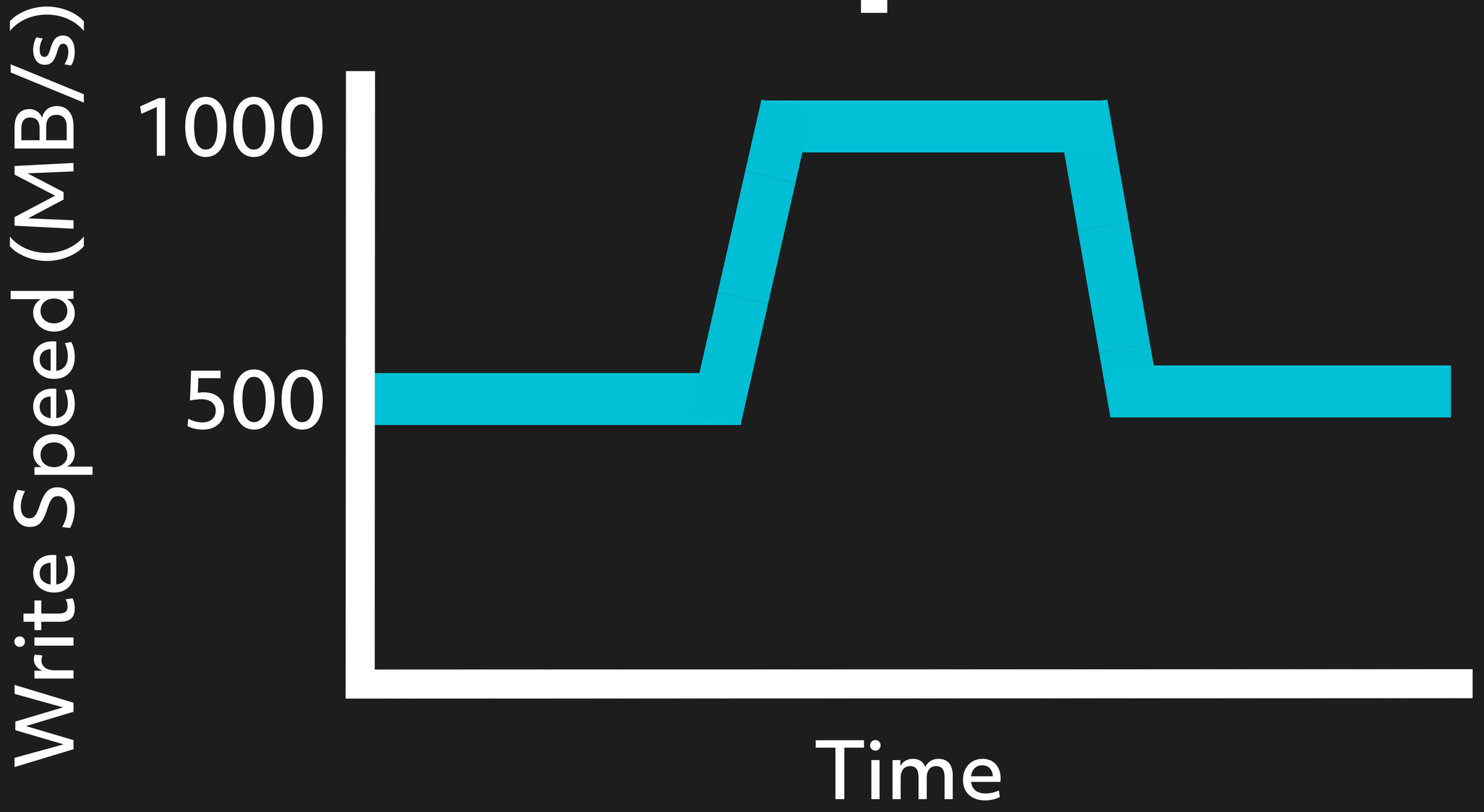
Disk operations

```
write("some data")
write("some more data")
write("yet more data")

flush()
```

Hardware disk caches

- Reduce latency
- Handle spikes

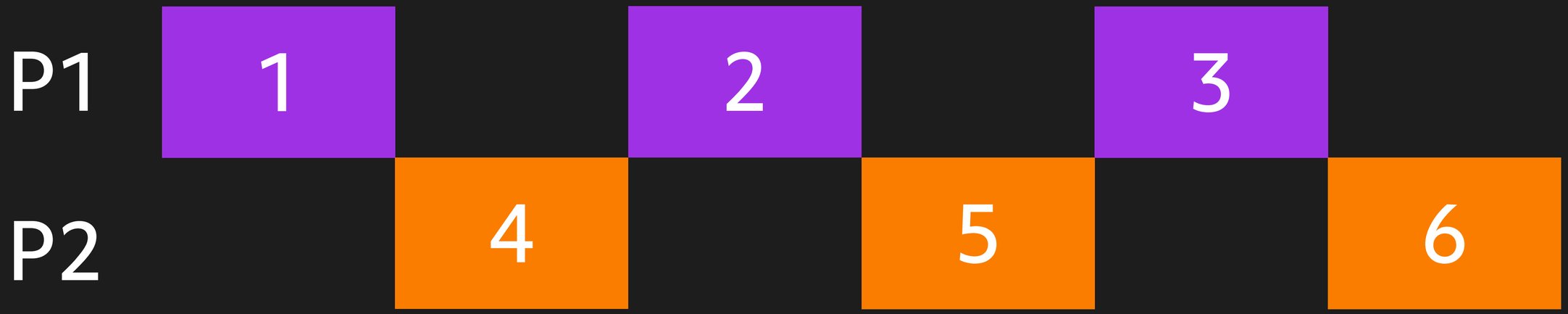
Write spikes



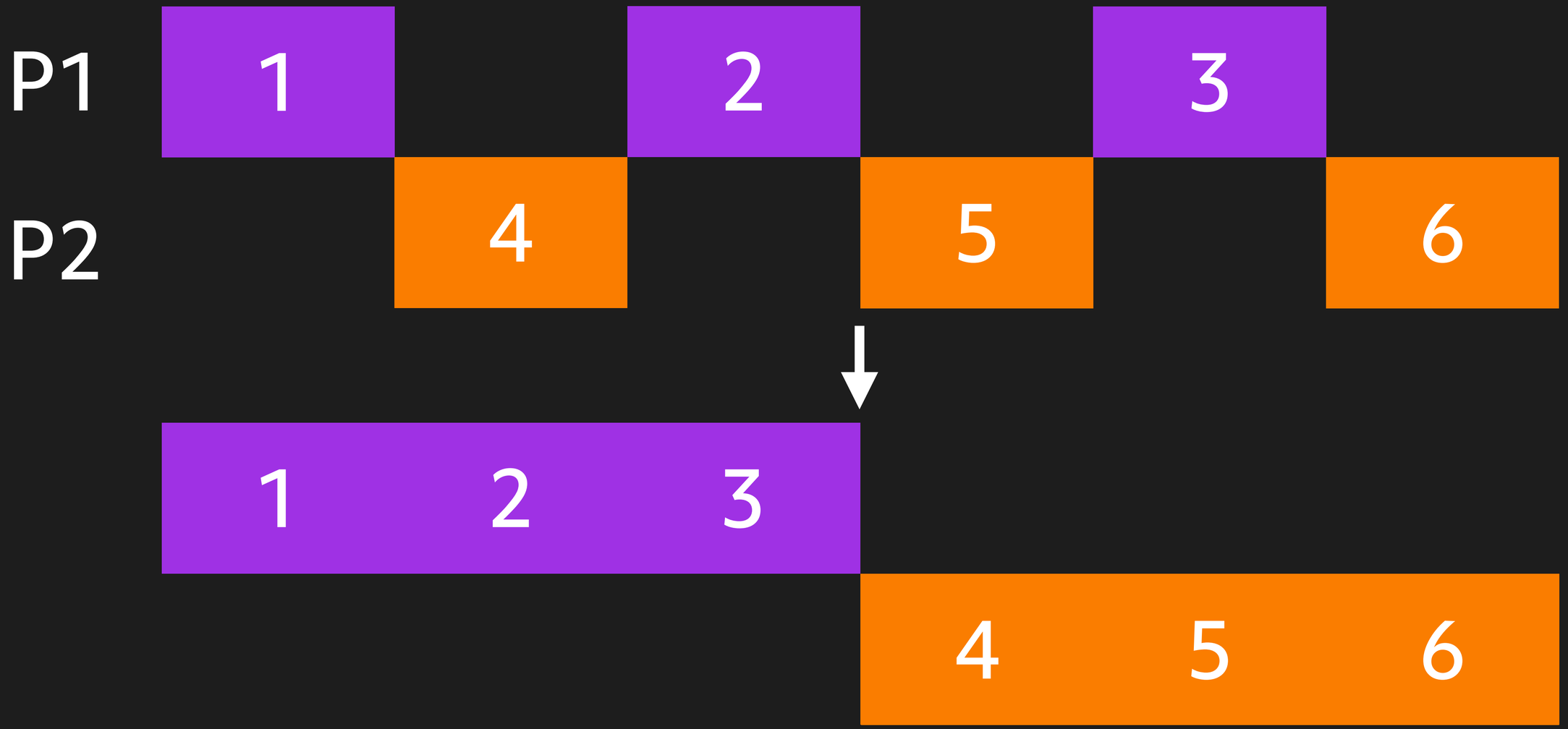
Hardware disk caches

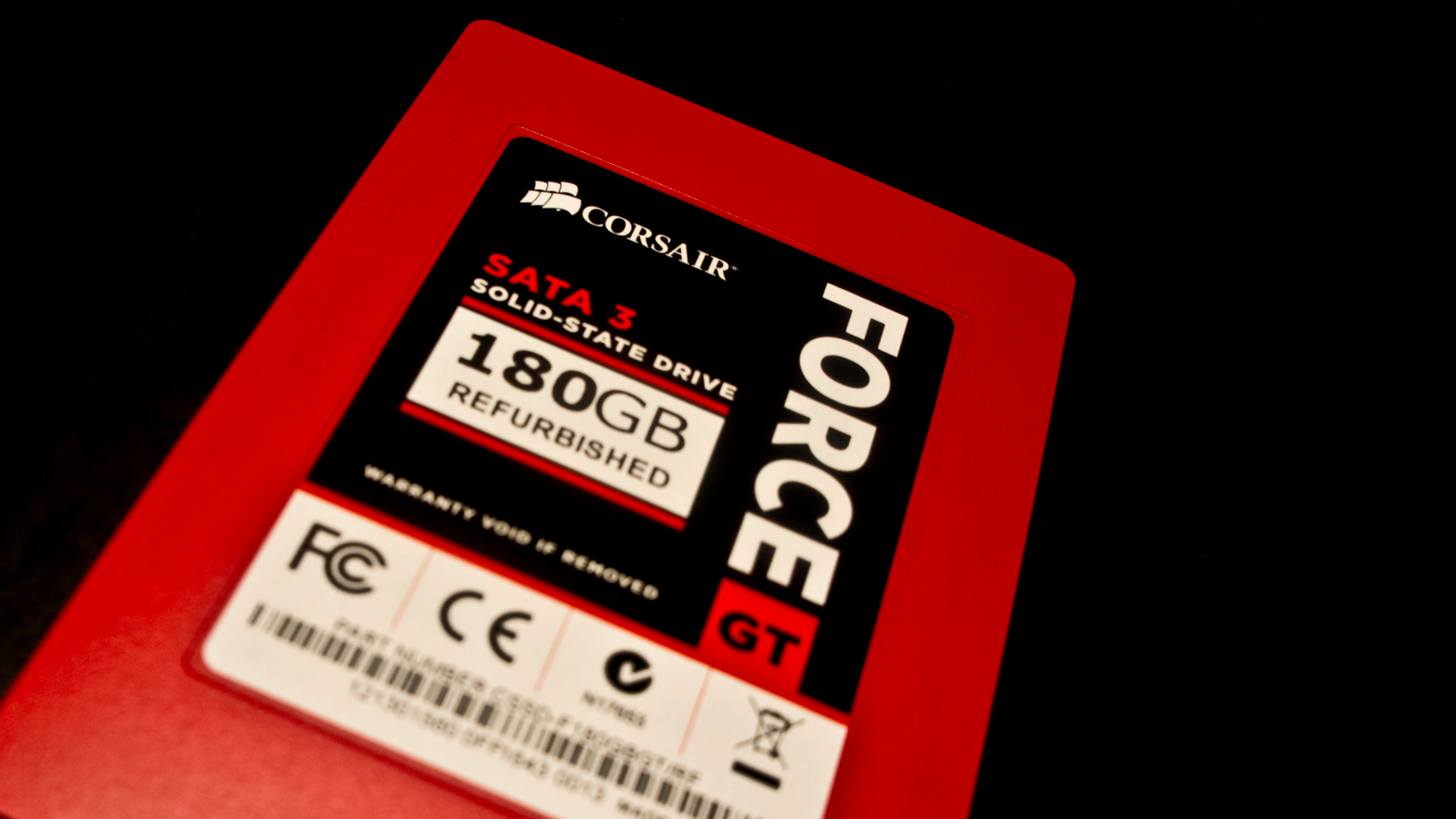
- Reduce latency
- Handle spikes
- Write reordering

Write reordering



Write reordering




CORSAIR
SATA 3
SOLID-STATE DRIVE
180GB
REFURBISHED
WARRANTY VOID IF REMOVED
FORCE
GT


FIRECUDA®
SEAGATE
1TB
FireCuda® 520 SSD
PCIe G4 x4 NVMe
SN: 7QG002F1
PN: 2TN301-300
ZP1000GM30002
FW: STNSC011
CAUTION:
HOT SURFACE
Model: STA020
CAN ICES-3(B) / NMB-3(B)

A cynical reason:

A cynical reason:

to look good

on bad

benchmarks

What does this mean for **fsync**?

fsync pseudocode

```
file = File.open("/data/base")

file.write("some data")
file.write("some more data")
file.write("yet more data")

file.fsync
```

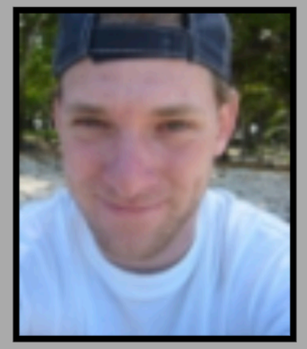
fsync pseudocode

```
file = File.open("/data/base")

file.write("some data")
file.write("some more data")
file.write("yet more data")

file.fsync
```

Flush commands

- **(S)ATA:** `FLUSH CACHE EXT`
- **SCSI:** `SYNCHRONIZE CACHE`
- **NVMe:** `FLUSH`

brad's life [entries|archive|friends|userinfo]



Brad Fitzpatrick

[**website**|bradfitz.com]
[**userinfo**|livejournal userinfo]
[**archive**|journal archive]

diskchecker.pl [May. 9th, 2005|**08:49 pm**]

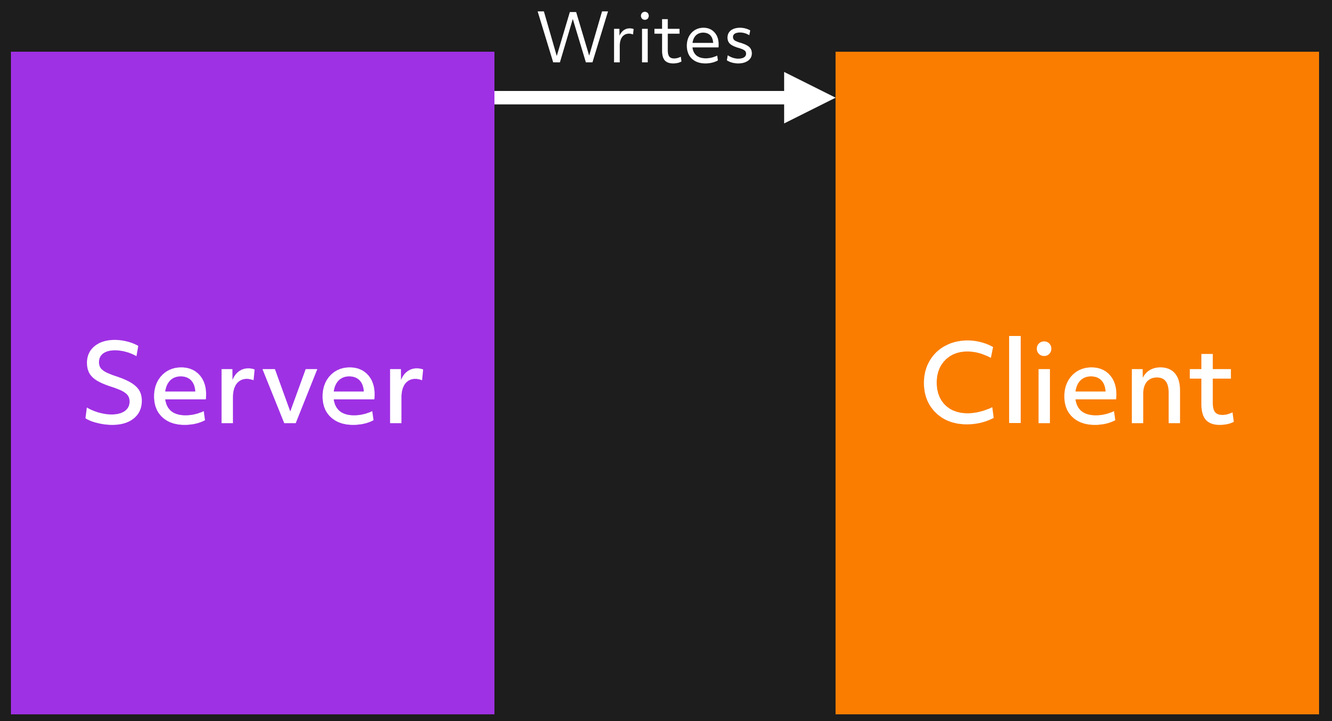
Remember my disk-checker program I wrote about before? I'd never released it because it was too hard to use, but now it's dead simple, so here it is:

~~http://code.sixapart.com/svn/tools/trunk/diskchecker.pl~~ Edit: now https://gist.github.com/3172656

Run it and be amazed how much your disks/raid/OS lie. ("lie" = an fsync doesn't work)

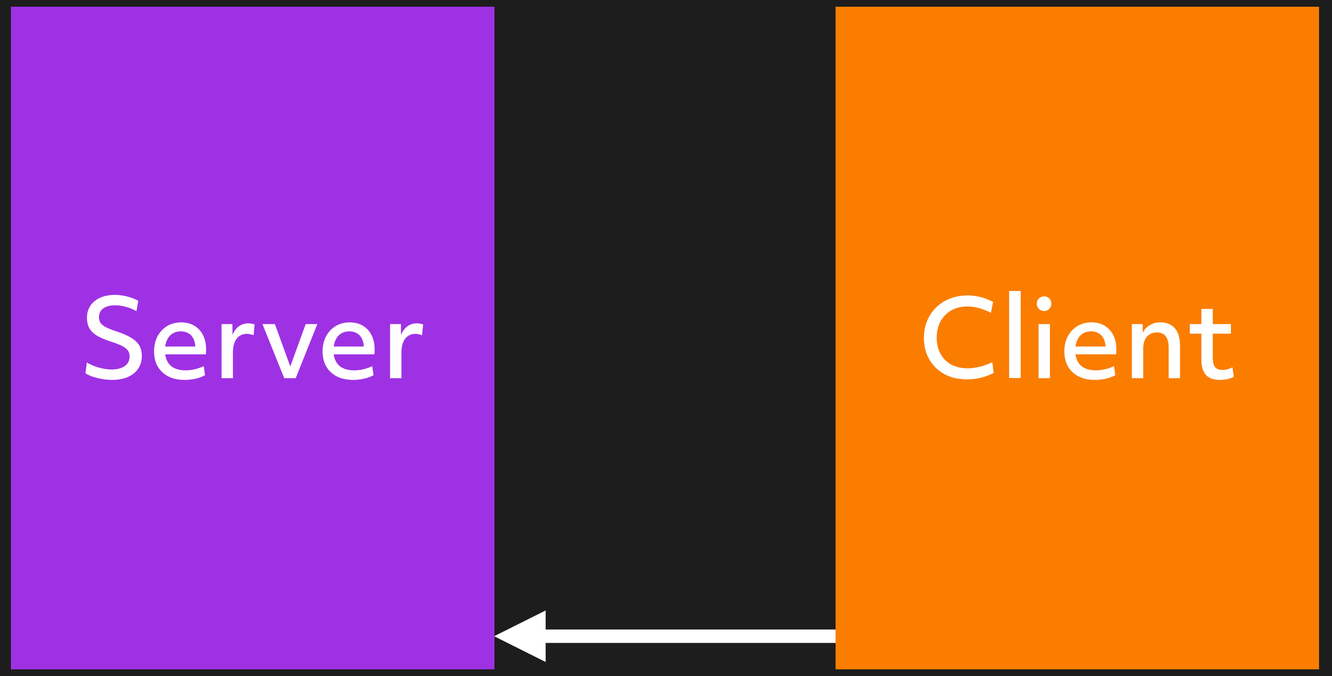
https://brad.livejournal.com/2116715.html

Tl;dr: `diskchecker.pl`

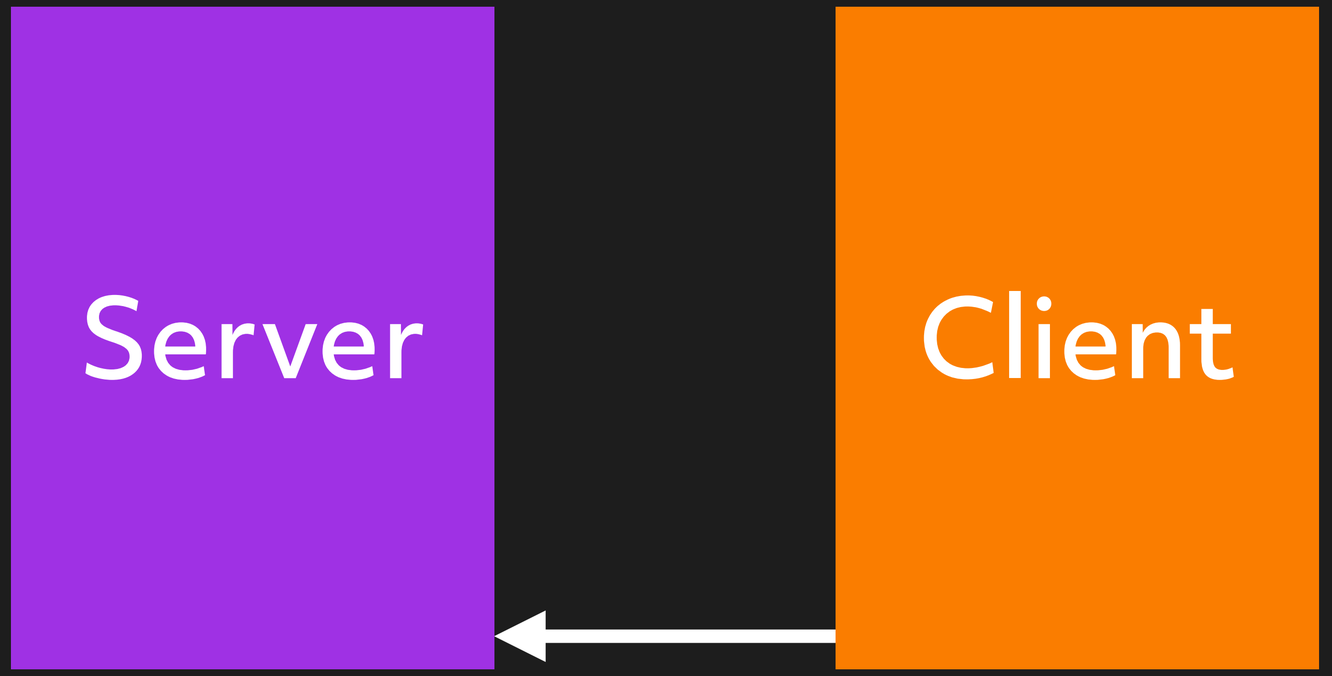


Tl;dr: diskchecker.pl

Tl;dr: `diskchecker.pl`



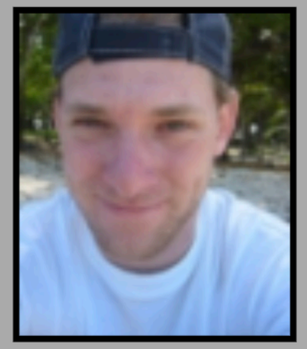
Tl;dr: `diskchecker.pl`



Fix for bad drives

```
# Disable write-back caching
#
# Might fix flush on a bad drive,
# but ruin performance

hdparm -W 0 /dev/sda
```

brad's life [entries|archive|friends|userinfo]



Brad Fitzpatrick

[**website**|bradfitz.com]
[**userinfo**|livejournal userinfo]
[**archive**|journal archive]

diskchecker.pl [May. 9th, 2005|**08:49 pm**]

Remember my disk-checker program I wrote about before? I'd never released it because it was too hard to use, but now it's dead simple, so here it is:

~~http://code.sixapart.com/svn/tools/trunk/diskchecker.pl~~ Edit: now https://gist.github.com/3172656

Run it and be amazed how much your disks/raid/OS lie. ("lie" = an fsync doesn't work)

https://brad.livejournal.com/2116715.html

Russ Bishop @xenadu02 · Feb 21

Fun story: I tested a random selection of four NVMe SSDs from four vendors. Half lose FLUSH'd data on power loss. That is the flush went to the drive, confirmed, success reported all the way back to userspace. Then I manually yanked the cable. Boom, data gone.

52 330 1.5K

Russ Bishop @xenadu02 · Feb 21

The other half never lost data confirmed after a flush (F_FULLFSYNC on macOS) no matter how much I abused them. All four had perf hit from flushing so they are doing some work.

Top two performers on flush? One lost data 40% of the time. The other never lost any.

4 12 267

https://web.archive.org/web/20220221094231/https://twitter.com/xenadu02/status/1495694090796941314

Russ Bishop @xenadu02 · Feb 21

I guess review sites don’t test this stuff. Everyone just assumes data disappearing on crash/power loss is just how computers work?

I feel bad for the other two vendors who must have test suites and spent engineering hours making sure FLUSH works, only to find out no one cares

14 35 610

https://web.archive.org/web/20220221094654/https://twitter.com/xenadu02/status/1495695209958895618

Russ Bishop @xenadu02 · Feb 21

I guess review sites don't test this stuff. Everyone just assumes data disappearing on crash/power loss is just how computers work?

I feel bad for the other two vendors who must have test suites and spent engineering hours making sure FLUSH works, only to find out no one cares

14 35 610

https://web.archive.org/web/20220221094654/https://twitter.com/xenadu02/status/1495695209958895618

Telling lies

without

breaking promises

RAID Controllers with BBU

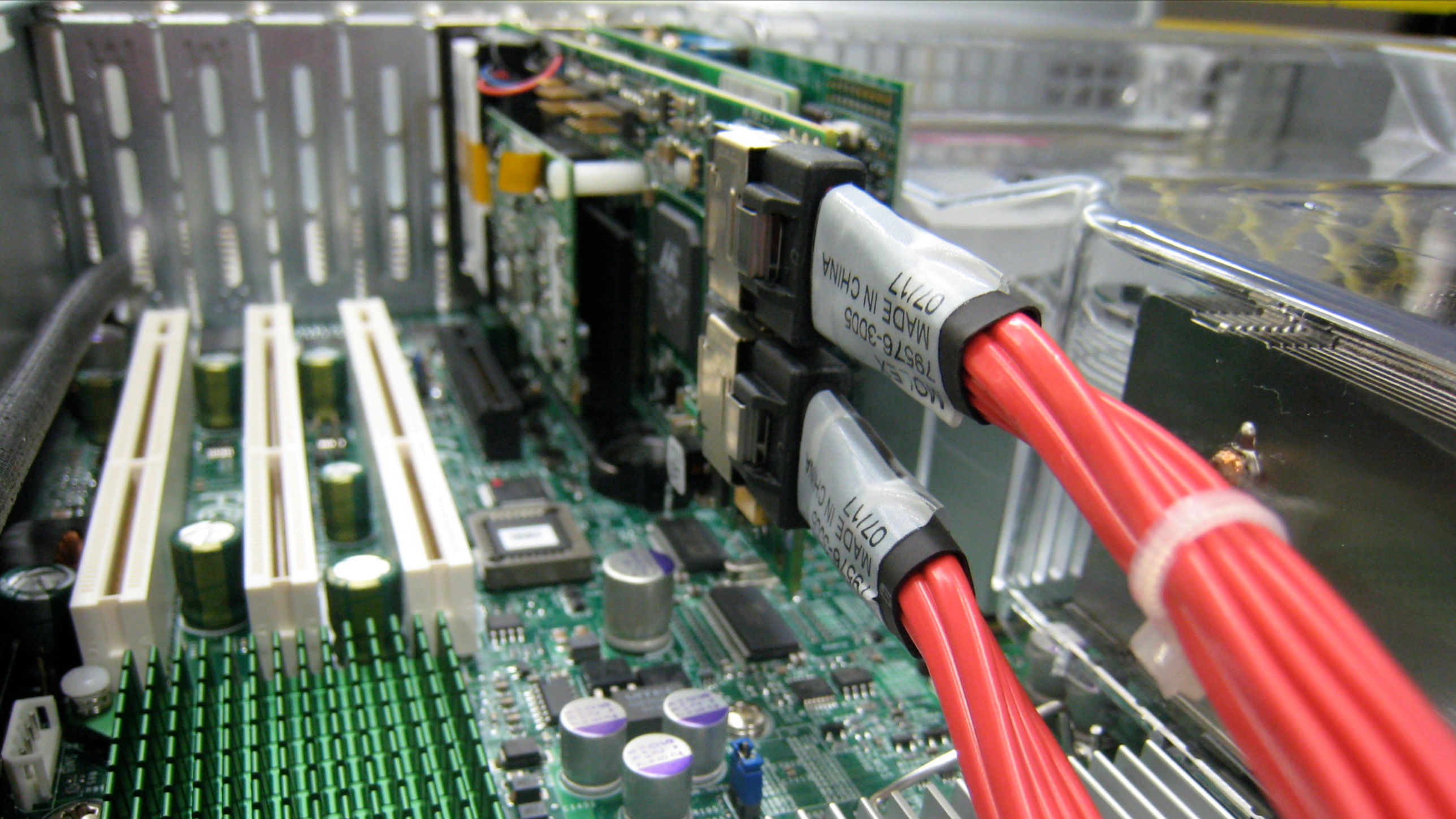
and

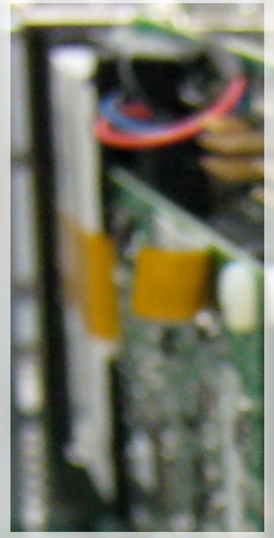
Drives with Capacitors

RAID Controllers with BBU

and

Drives with Capacitors


MADE IN CHINA
07/17



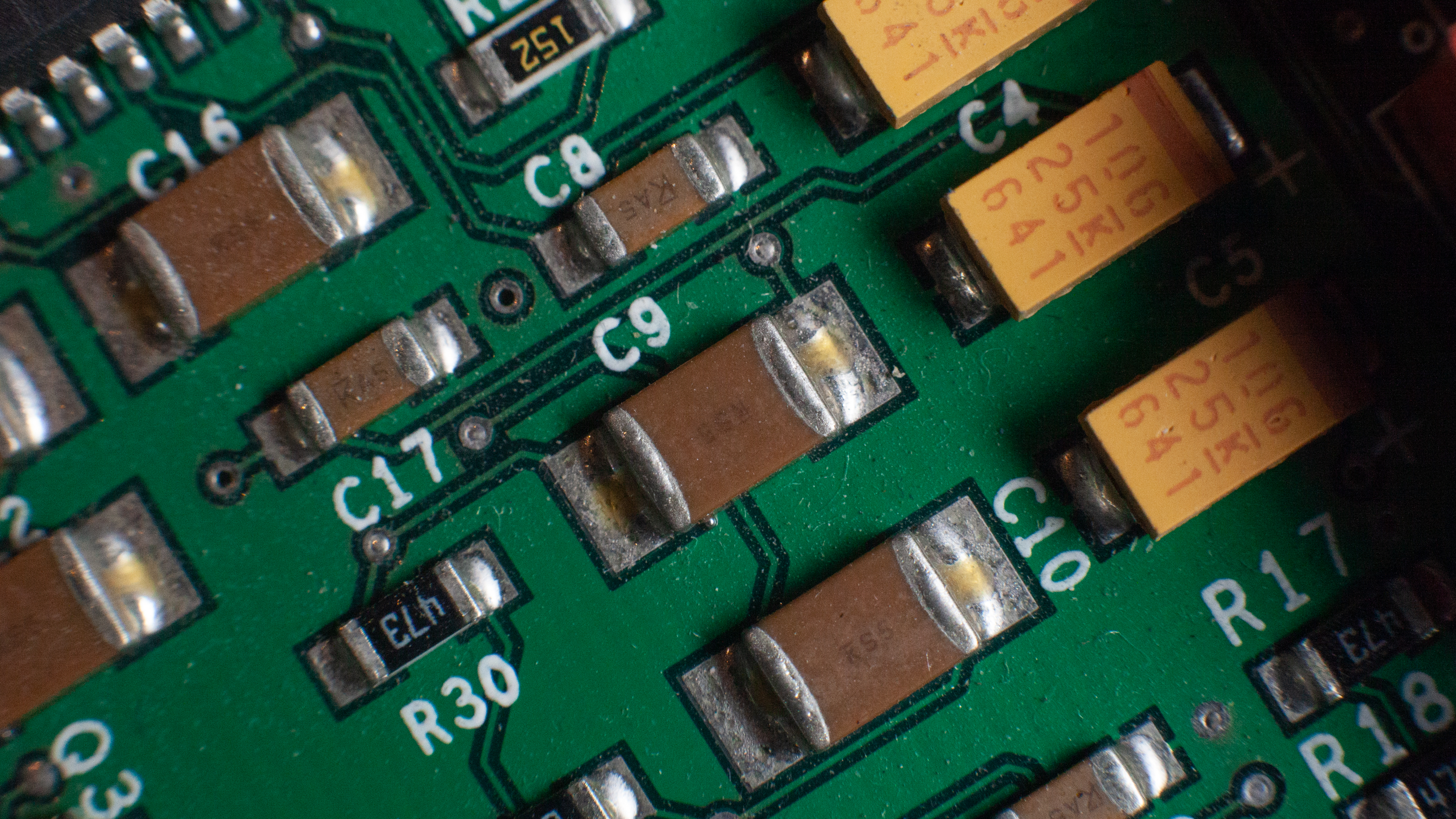
72 hours

1-10 hours

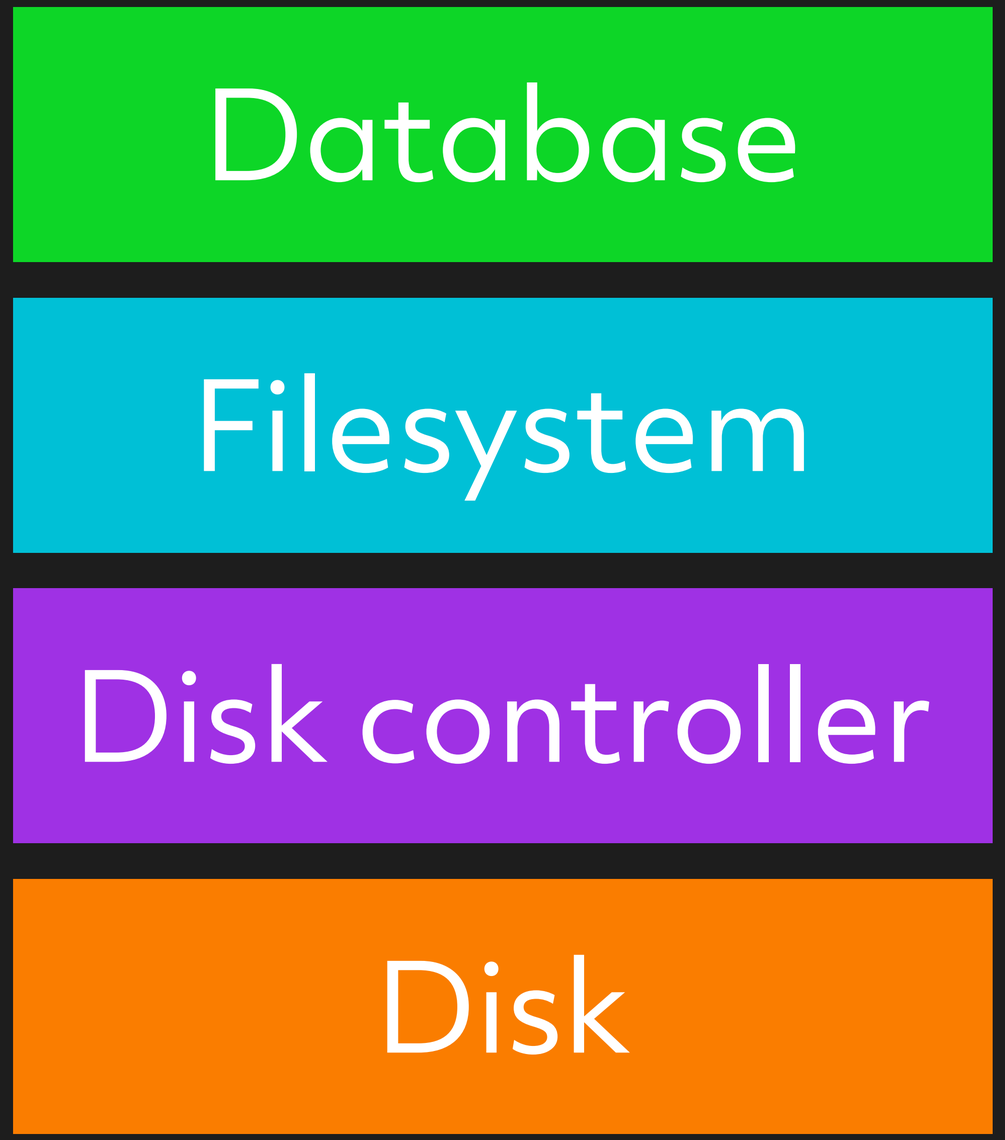
RAID Controllers with BBU

and

Drives with Capacitors


C16
C8
C4
C9
C3
C17
C10
R17
R30
R18
251

All
of this
is your
responsibility



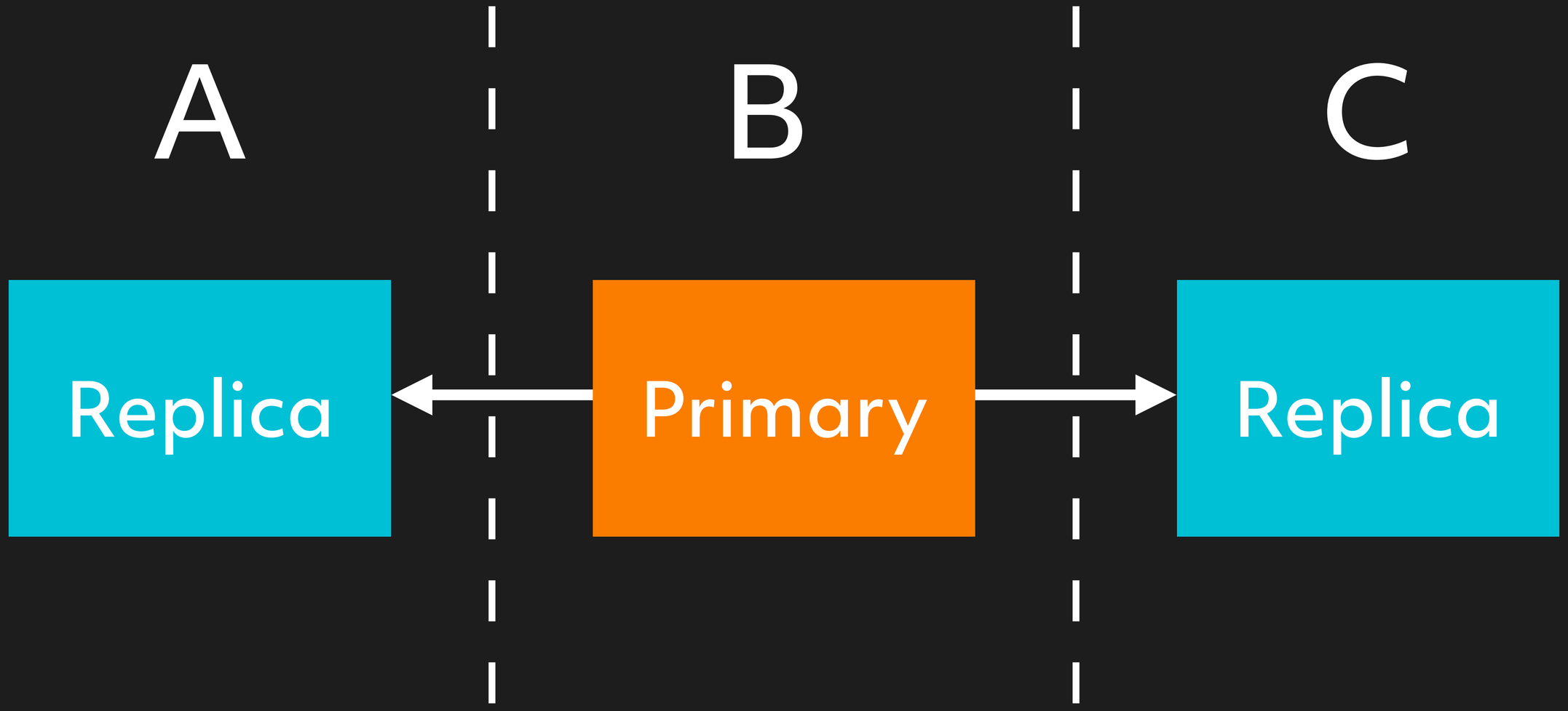
The computer might
reward you...

...by
losing your data

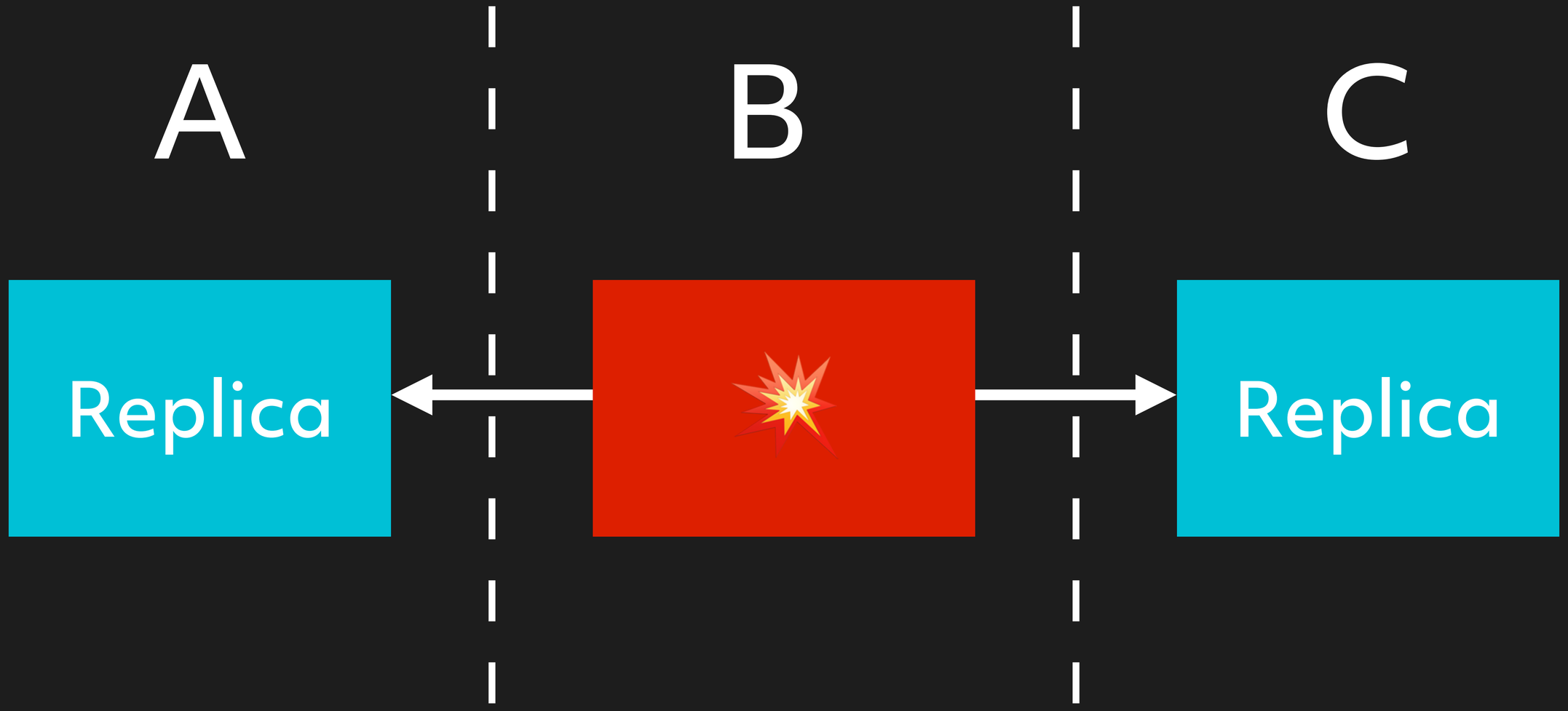
Before we wrap up

An aside

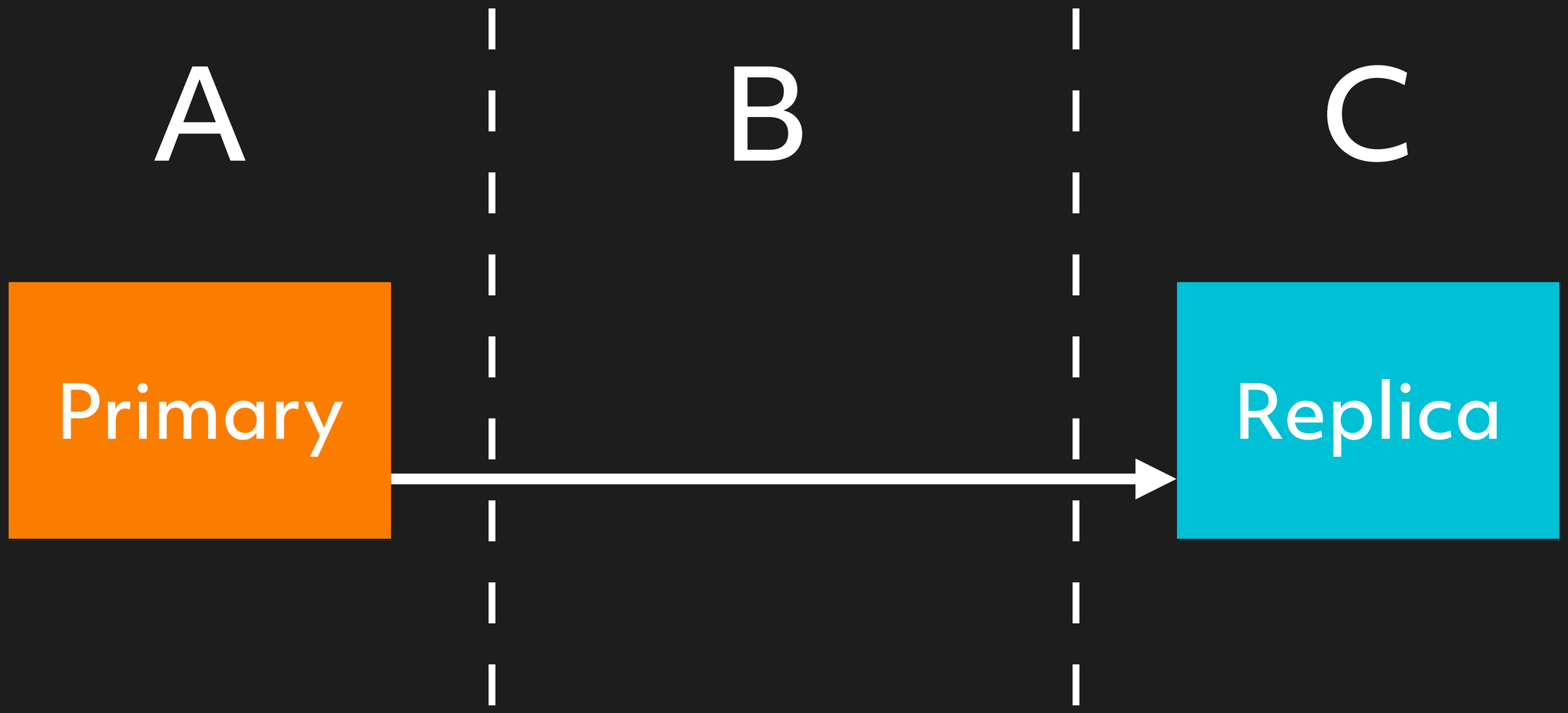
Datacentres/AZs



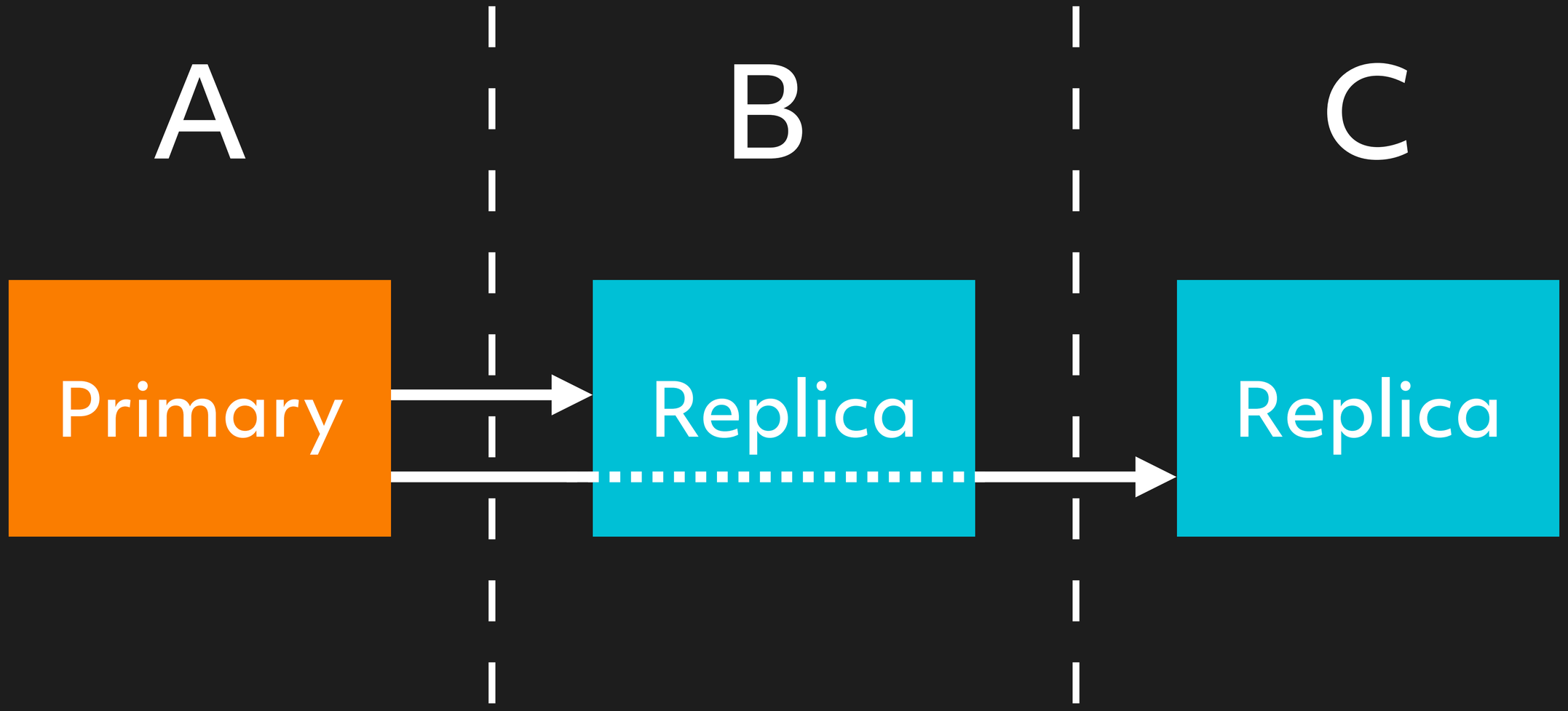
Datacentres/AZs



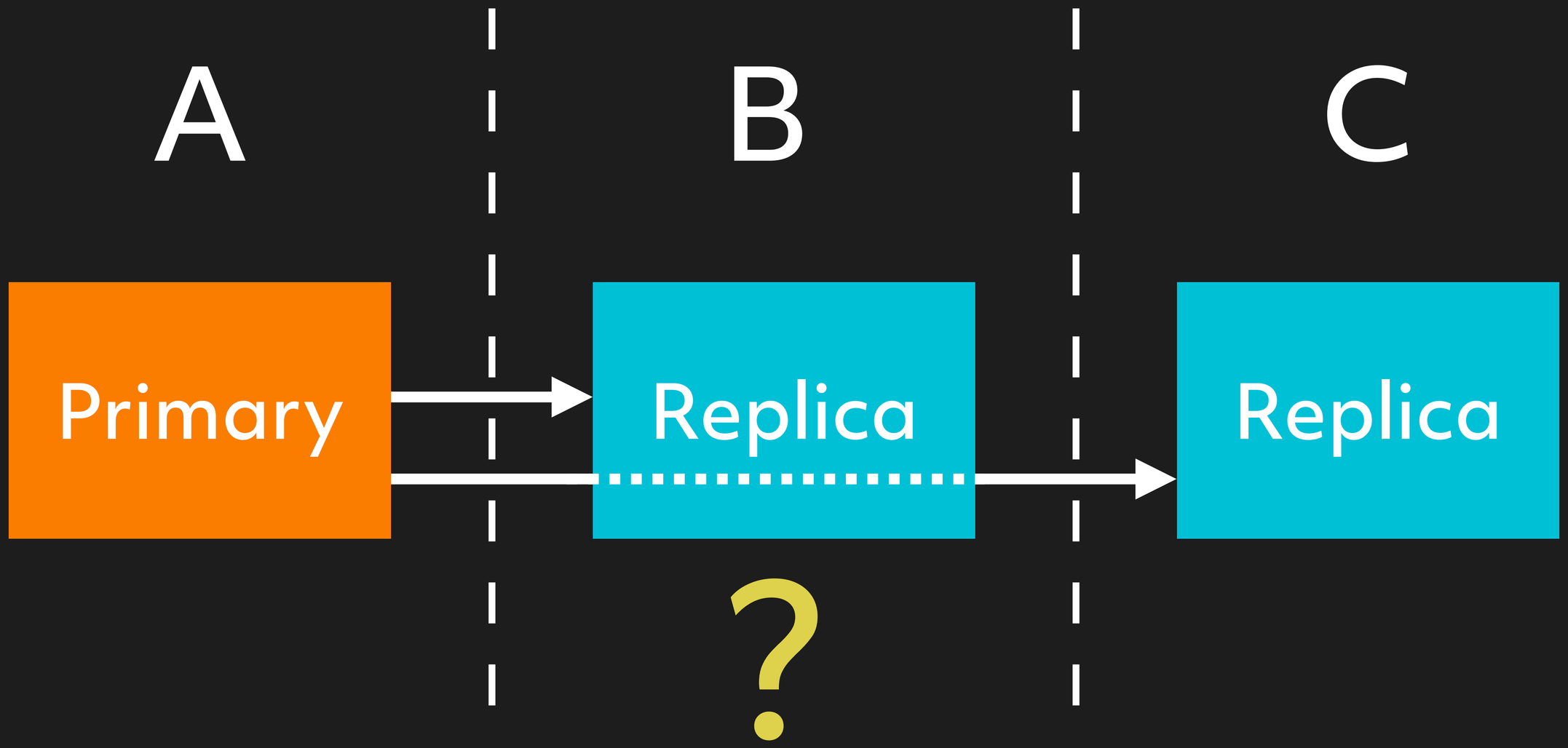
Datacentres/AZs



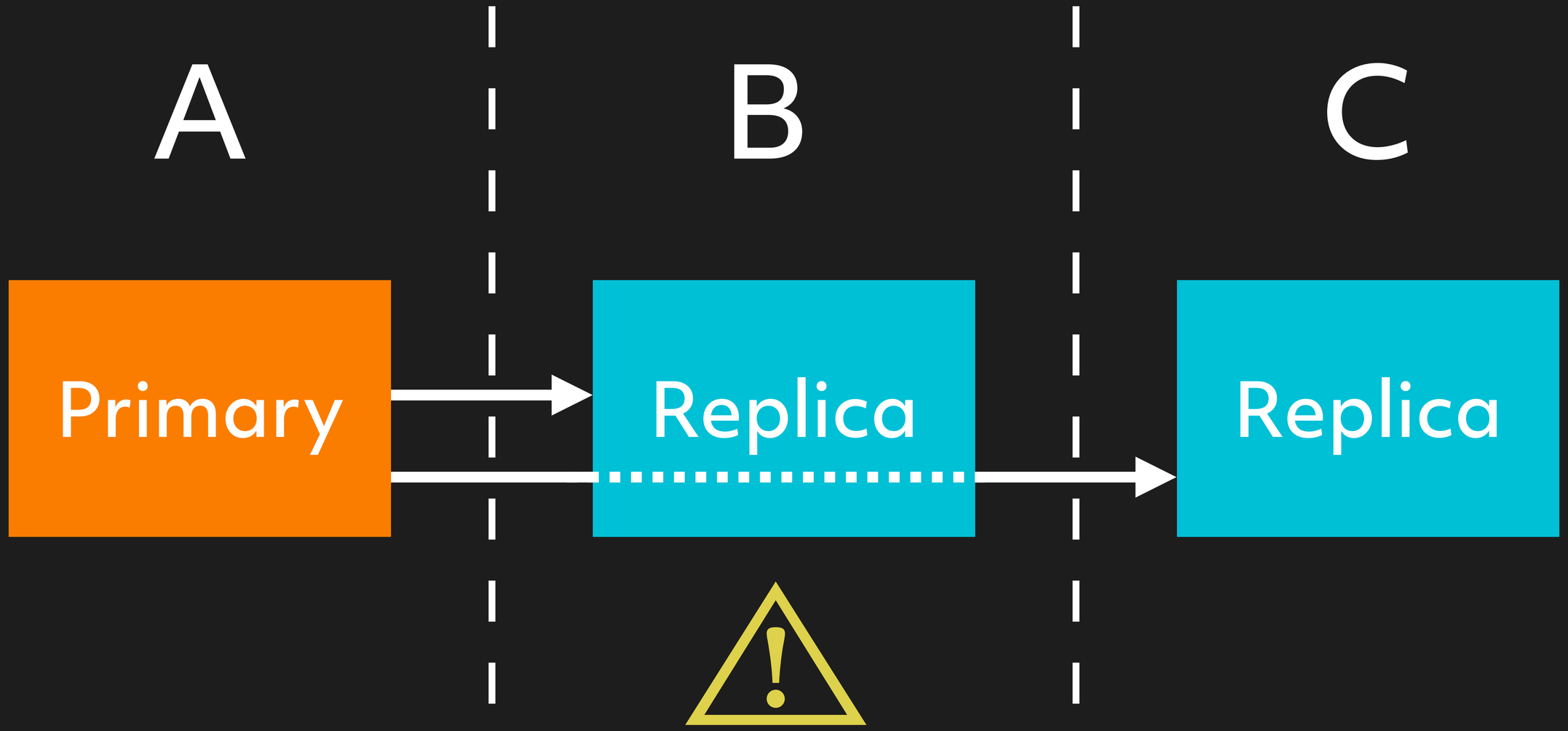
Datacentres/AZs



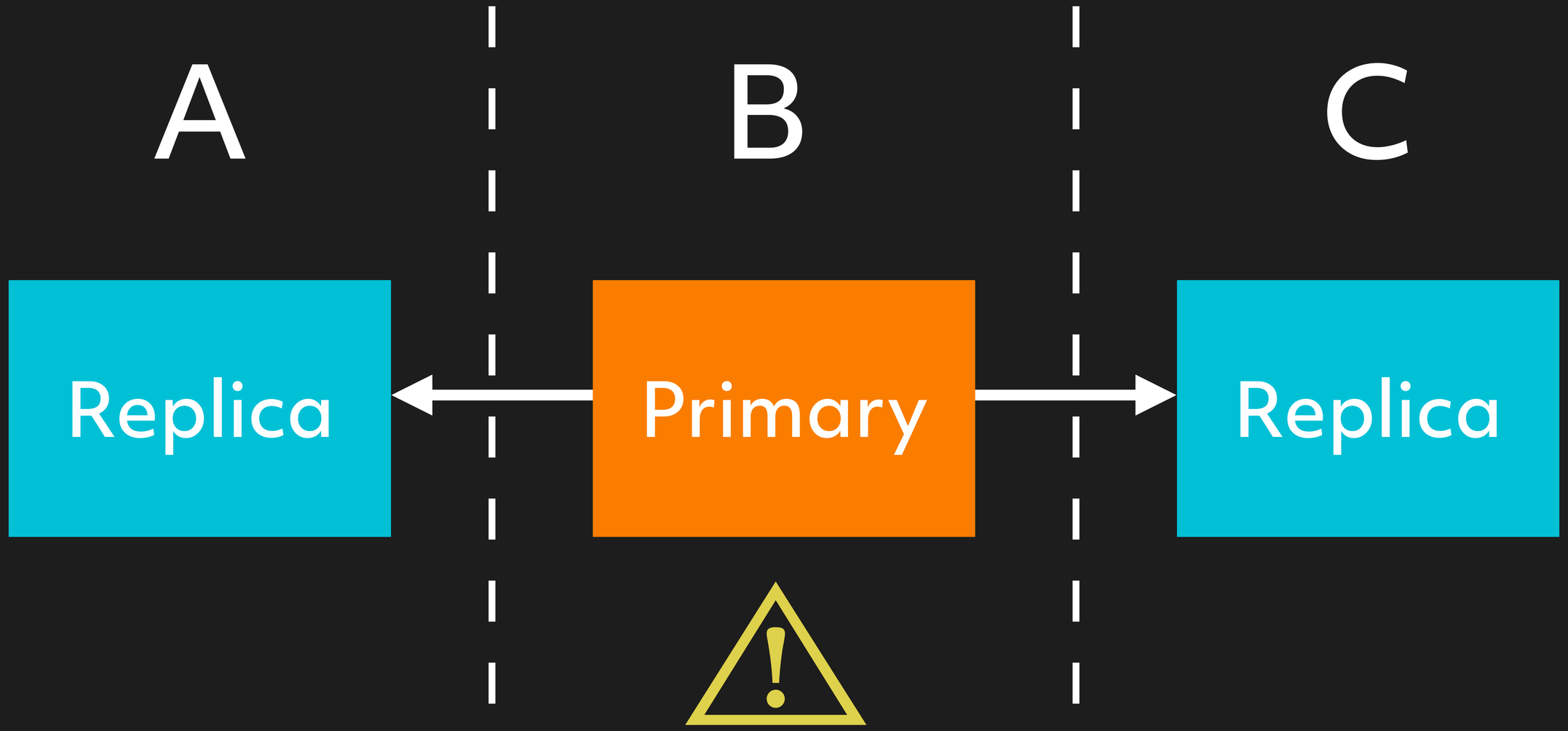
Datacentres/AZs



Datacentres/AZs



Datacentres/AZs



Replication
doesn't save us from
thinking about
crash safety

We need

both

Case Studies

Case 1

MySQL doublewrite buffer

Case 2

Postgres fsyncgate (2018)

Case 3

Hardware disk caches

Lessons

- Storage is unforgiving

The APIs are confusing

and

The stakes are high

Lessons

- Storage is unforgiving
- Higher layers can't fix lower ones

If these tell lies...

Database

Filesystem

Disk controller

Disk

...then these won't know

If these tell lies...

Database

Filesystem

Disk controller

Disk

Lessons

- Storage is unforgiving
- Higher layers can't fix lower ones
- Slower is easier

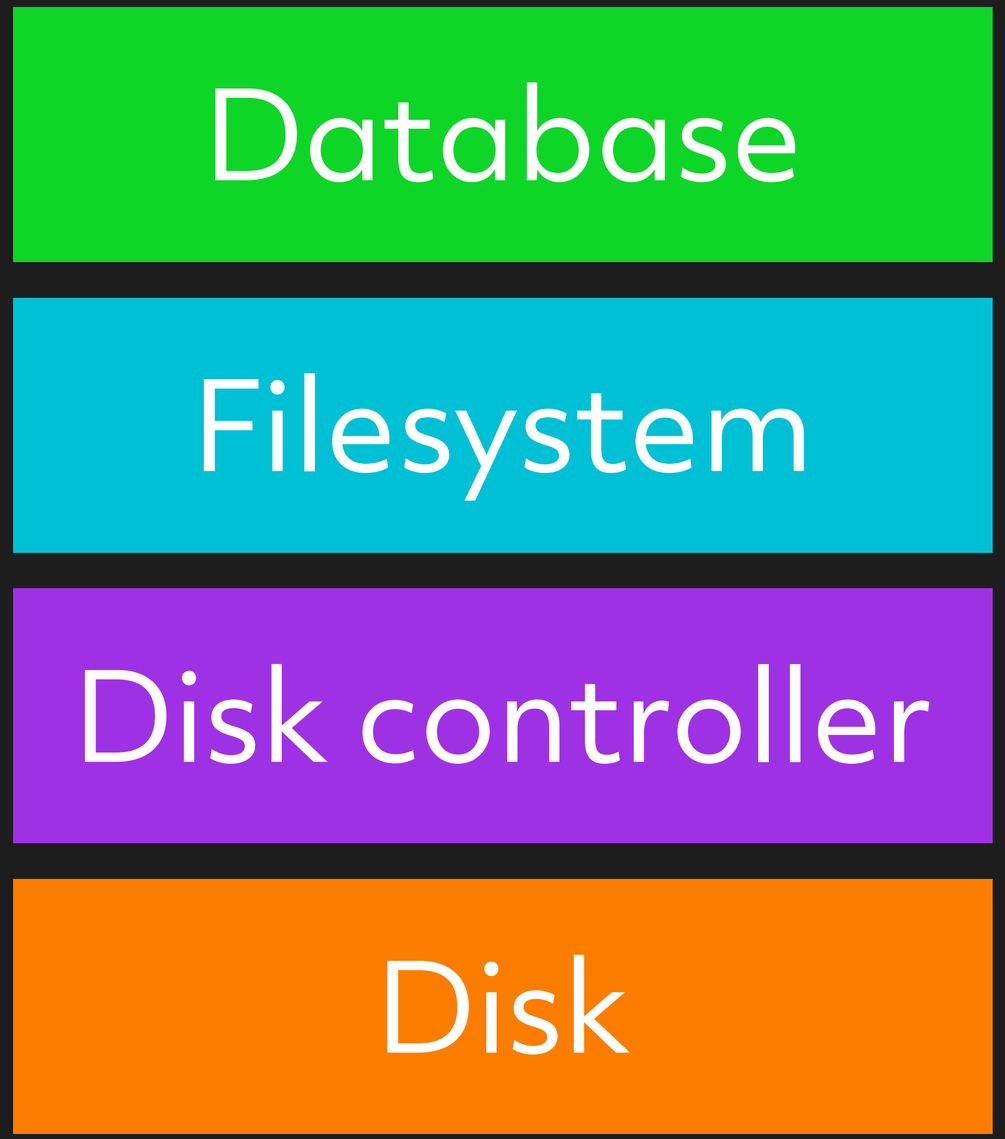
You can:

- Enable doublewrite
- Disable caches

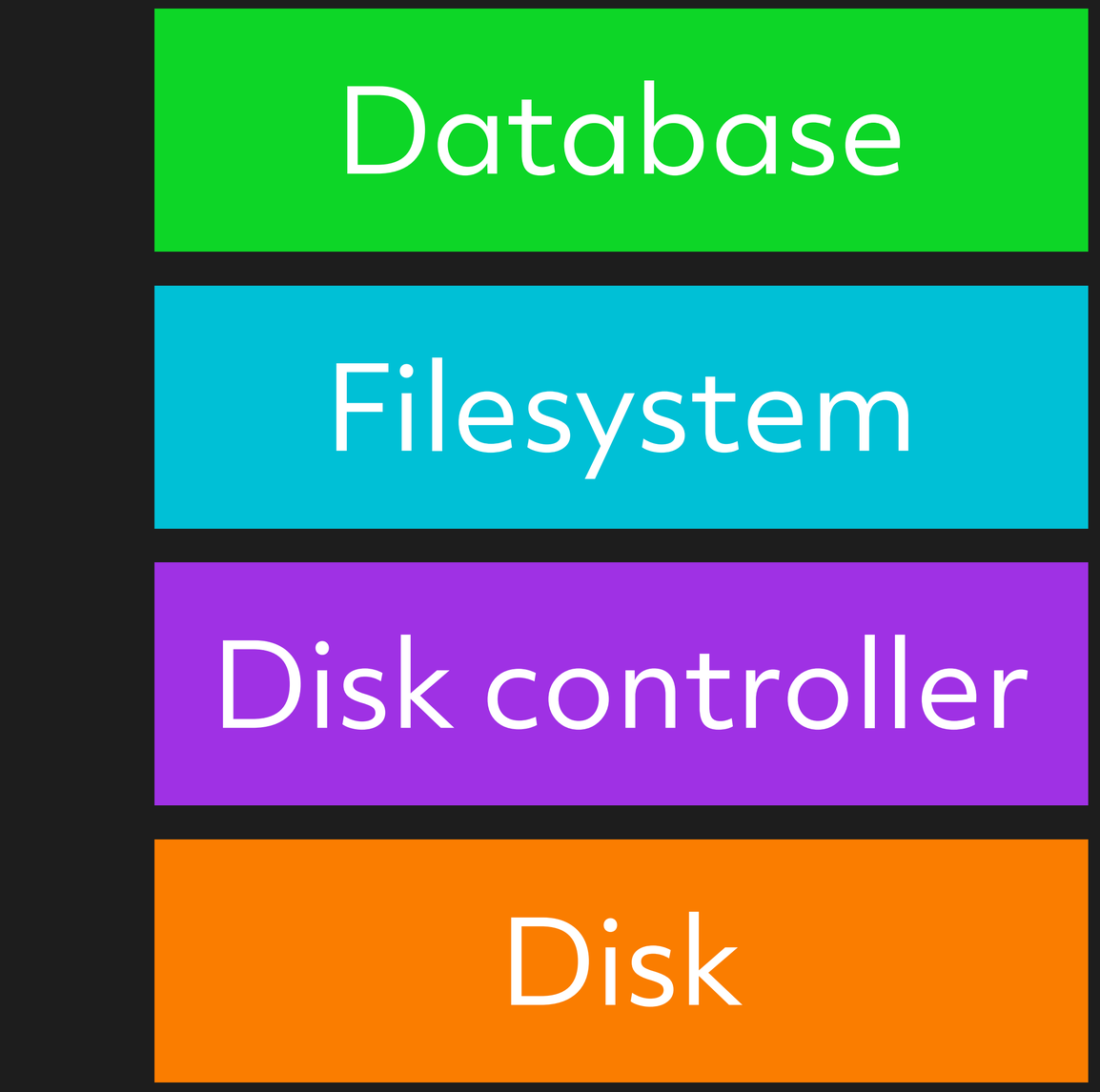
Fast & **Safe**

is hard(er)

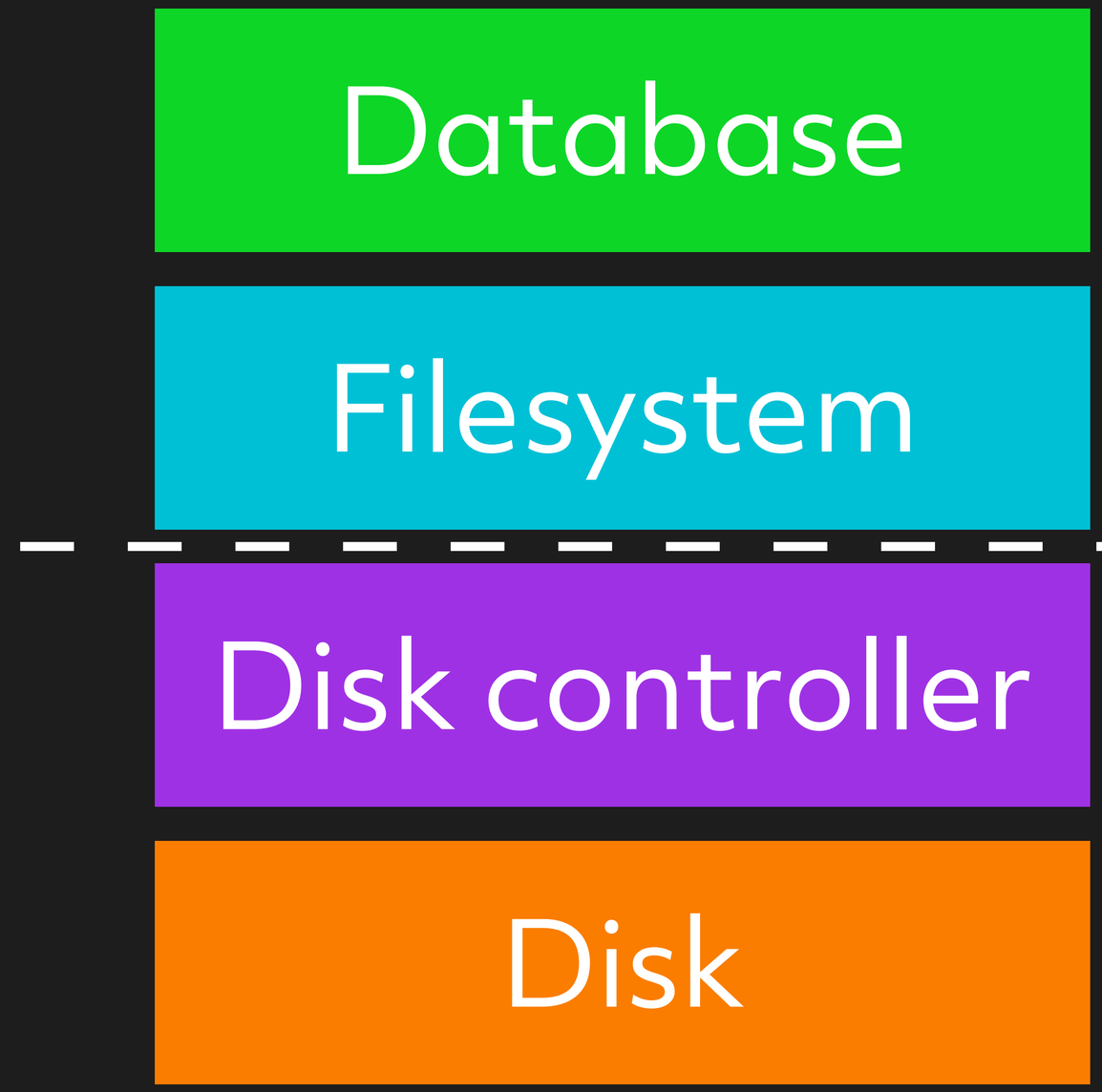
Choose
your own
adventure



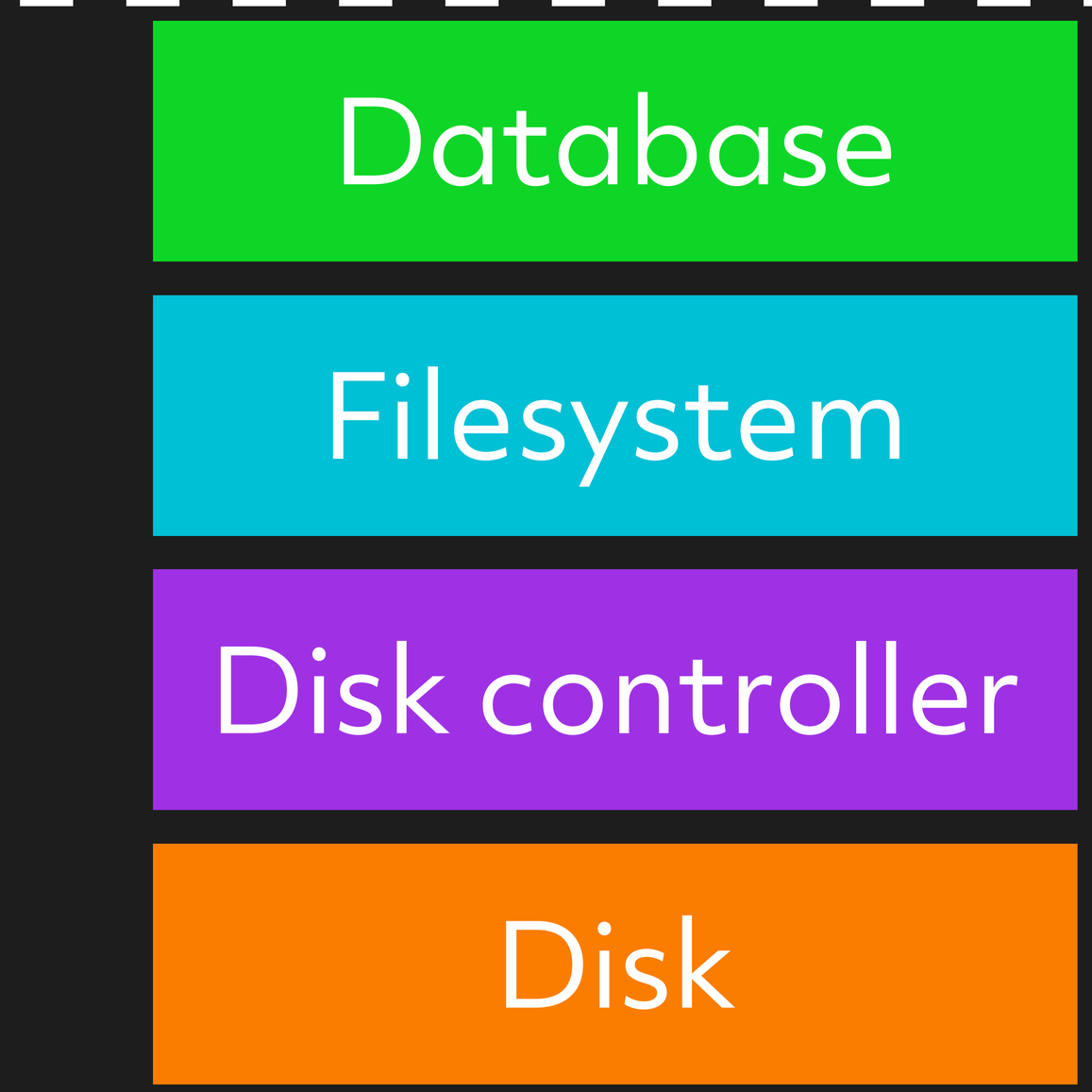
Choose
your own
adventure



Choose
your own
adventure



Choose
your own
adventure



PlanetScale

Thank you

✌️❤️

sinjo.dev

@PlanetScale

Image credits

- Twemoji Floppy Disk Emoji - CC-BY - https://github.com/twitter/twemoji/blob/d94f4cf793e6d5ca592aa00f58a88f6a4229ad43/assets/svg/1f4be.svg
- Hard Disk Guts - CC-BY - https://www.flickr.com/photos/mattandkim/97533589/
- Yellow Slippery Road Signage - CC0 - https://www.pexels.com/photo/sign-slippery-wet-caution-4341/
- Black and Green Circuit board - CC0 - https://www.pexels.com/photo/black-and-green-circuit-board-2644597/
- Corsair ForceGT 180GB - CC-BY - https://www.flickr.com/photos/ruocaled/8173124575/

Image credits

- High Performance NVMe SSD on Gray Surface - CC0 - https://www.pexels.com/photo/high-performance-nvme-ssd-on-gray-surface-28666524/
- Server Guts - CC-BY - https://www.flickr.com/photos/chrisdag/2142582850
- Ceramic capacitors mounted on a PCB - CC-BY - https://commons.wikimedia.org/w/index.php?curid=113868467
- NOIRLab HQ Server Racks - CC-BY - https://commons.wikimedia.org/wiki/File:NOIRLab_HQ_Server_Racks_(6V6A0402-CC).jpg

Questions?

✌️❤️

sinjo.dev

@PlanetScale