facebook

MD/RAID-456 Write Journal and Cache

Shaohua Li & **Song Liu**

Software Engineer, Facebook

MD/RAID-456 Write Journal and Cache

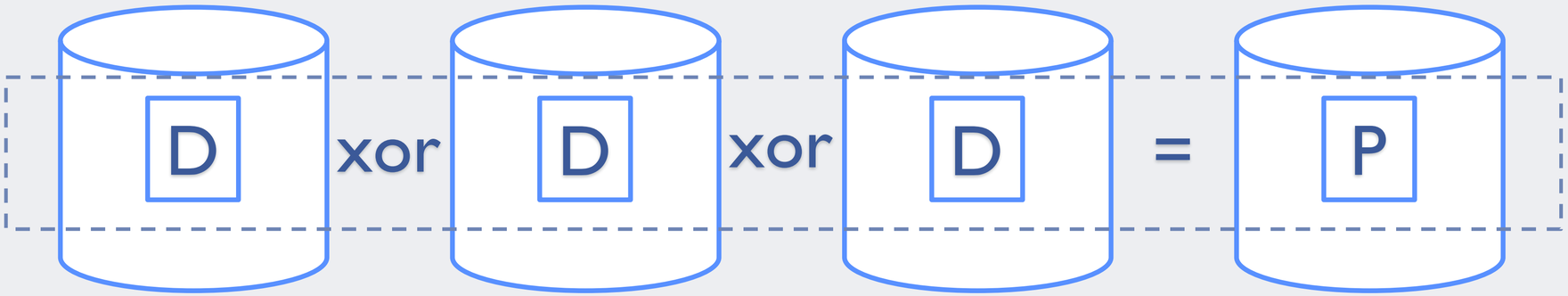
- Write holes of RAID-456
- Hardware RAID: benefits and challenges
- Write operation in MD/RAID-456
- RAID-456 write journal: plug the write hole
- RAID-456 write cache: fast fsync(), more full stripe writes
- Examples

Write Hole of RAID-456

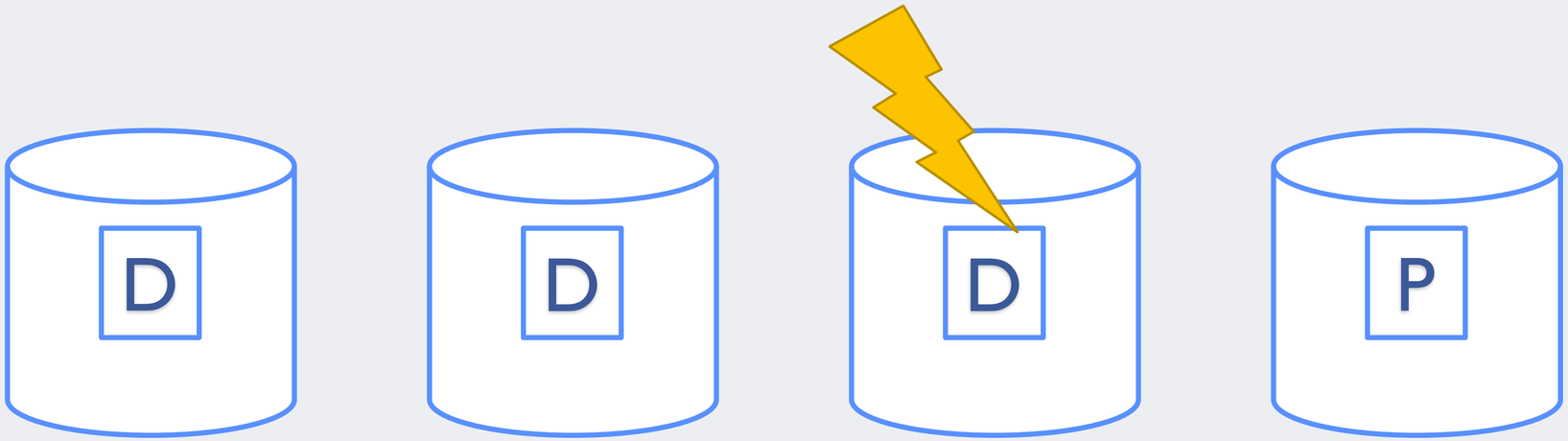
Failure Recovery of RAID-456

- Disk failure recovery: rebuild data from parity
- Power failure recovery: resync stripes with mis-matched data and parity

RAID-5: Data and Parity in Sync

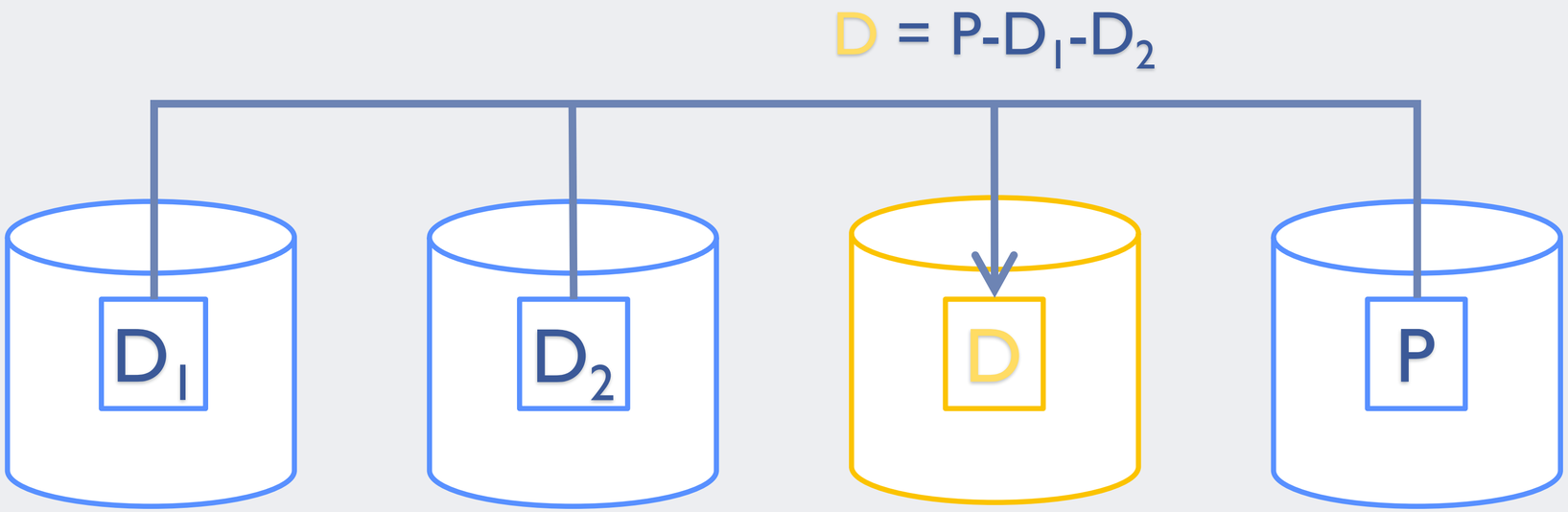


RAID-5 Disk Failure

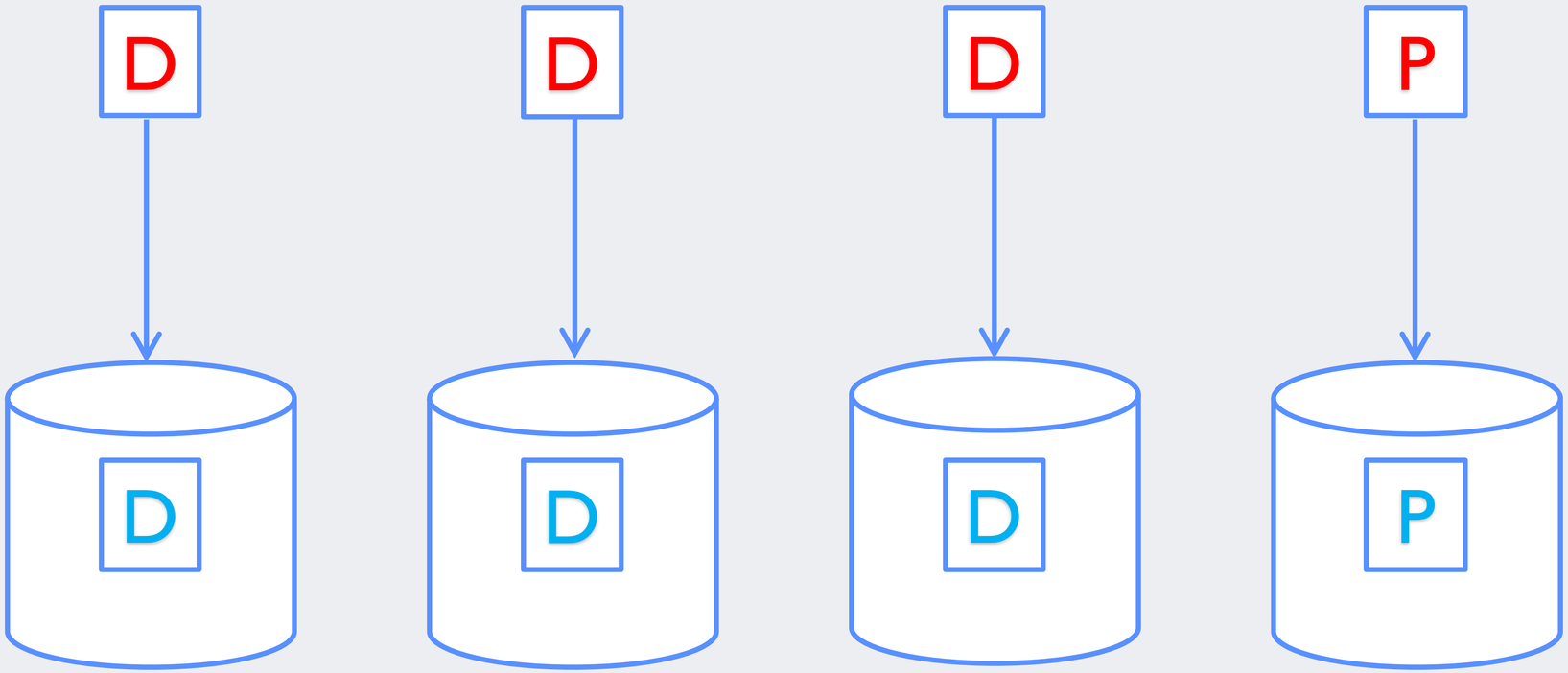


RAID-5 Degraded Mode

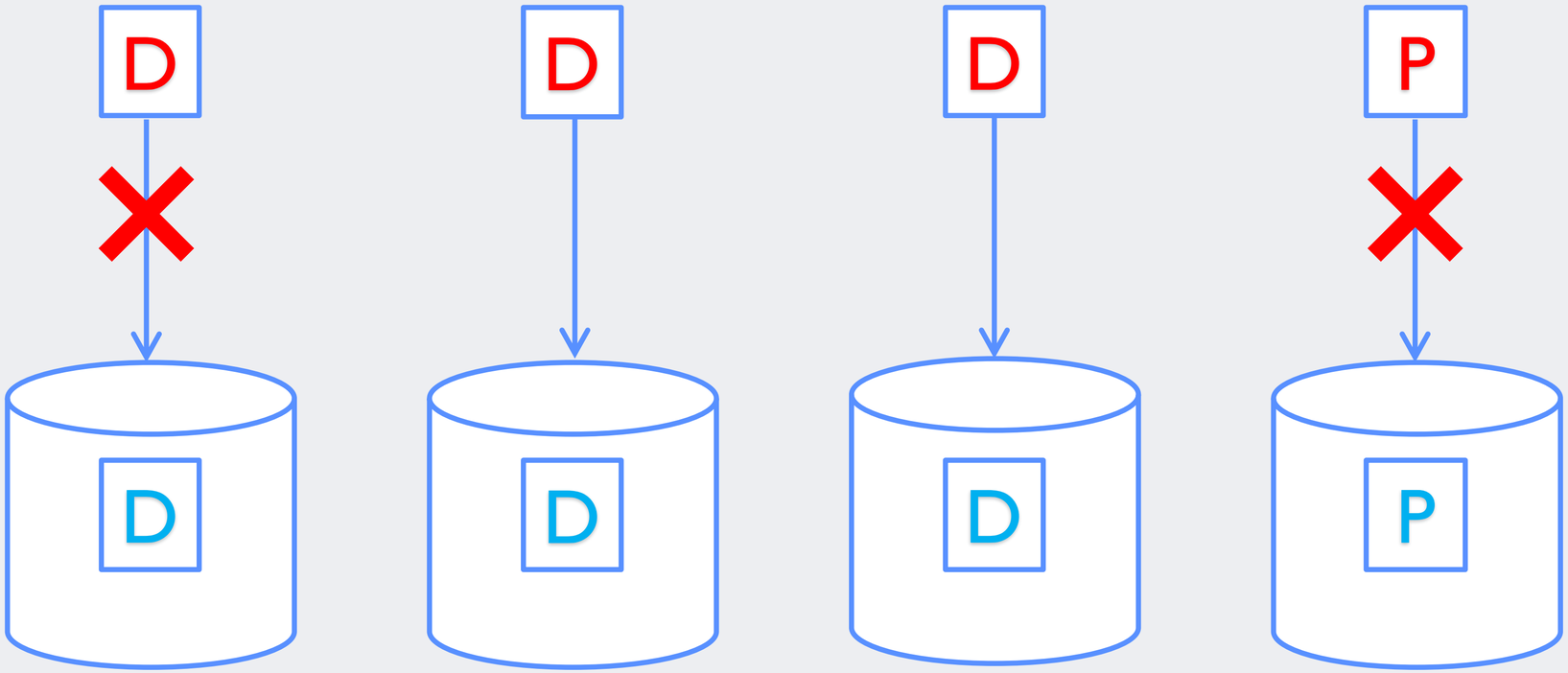
RAID-5 Rebuild Data for Disk Failure



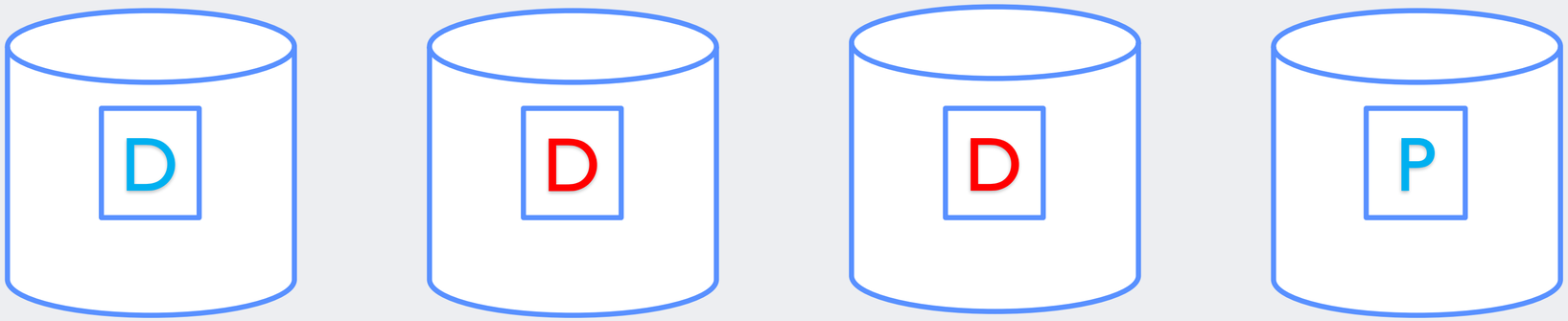
RAID-5 Power Failure



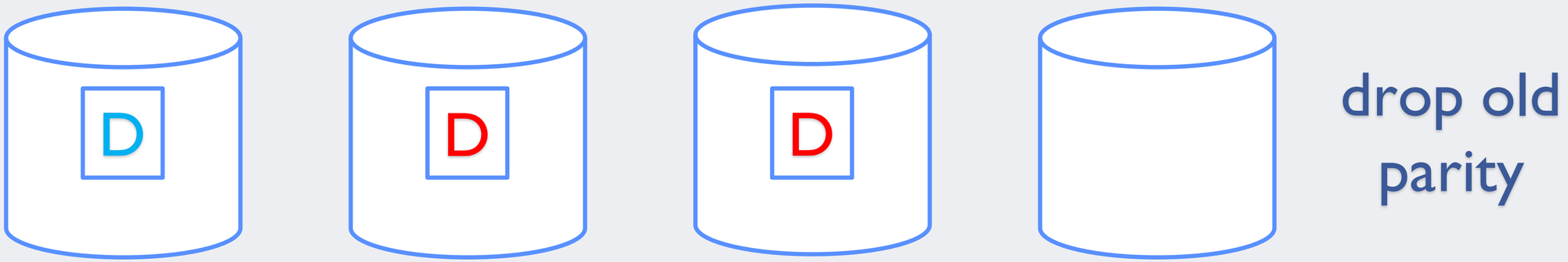
RAID-5 Power Failure



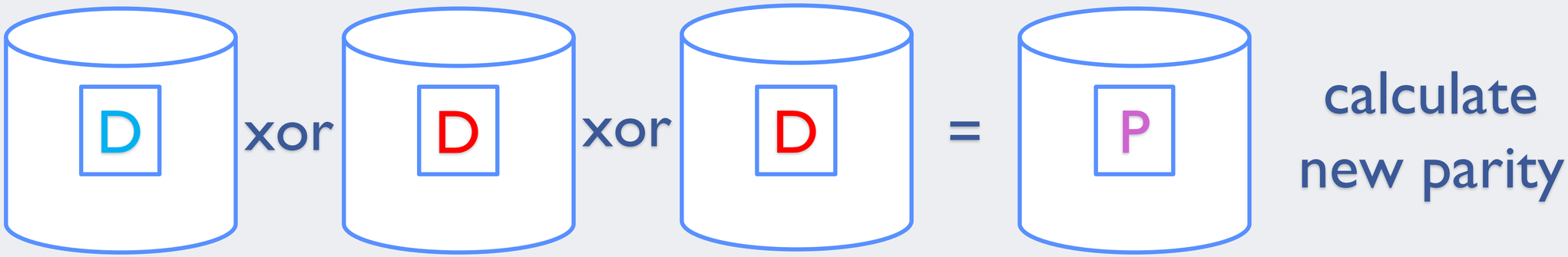
RAID-5 after Power Failure



RAID-5 Resync after Power Failure

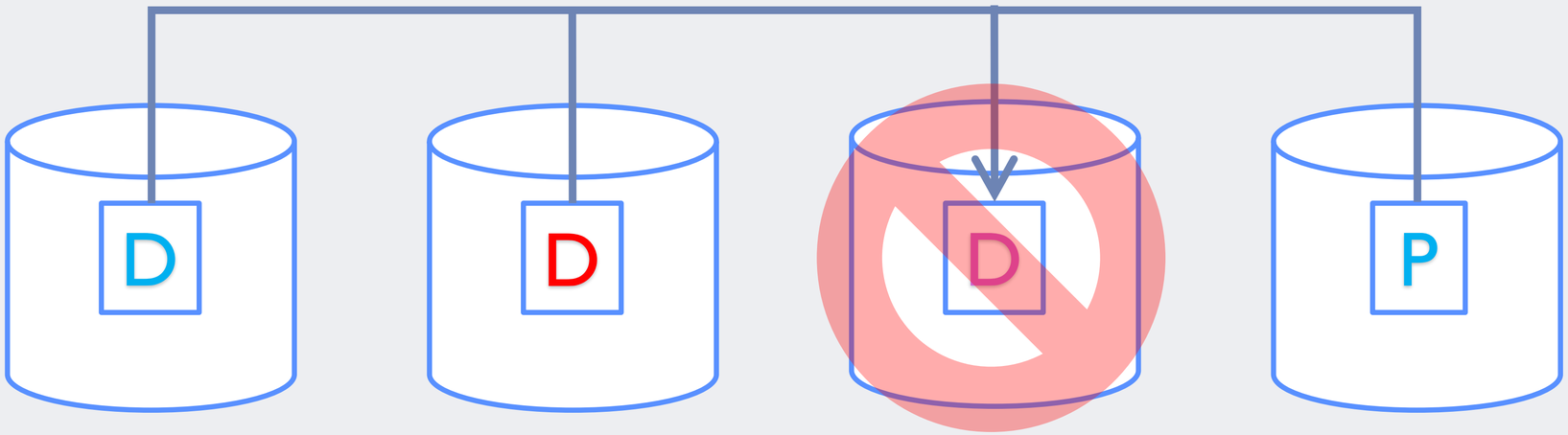


RAID-5 Resync after Power Failure



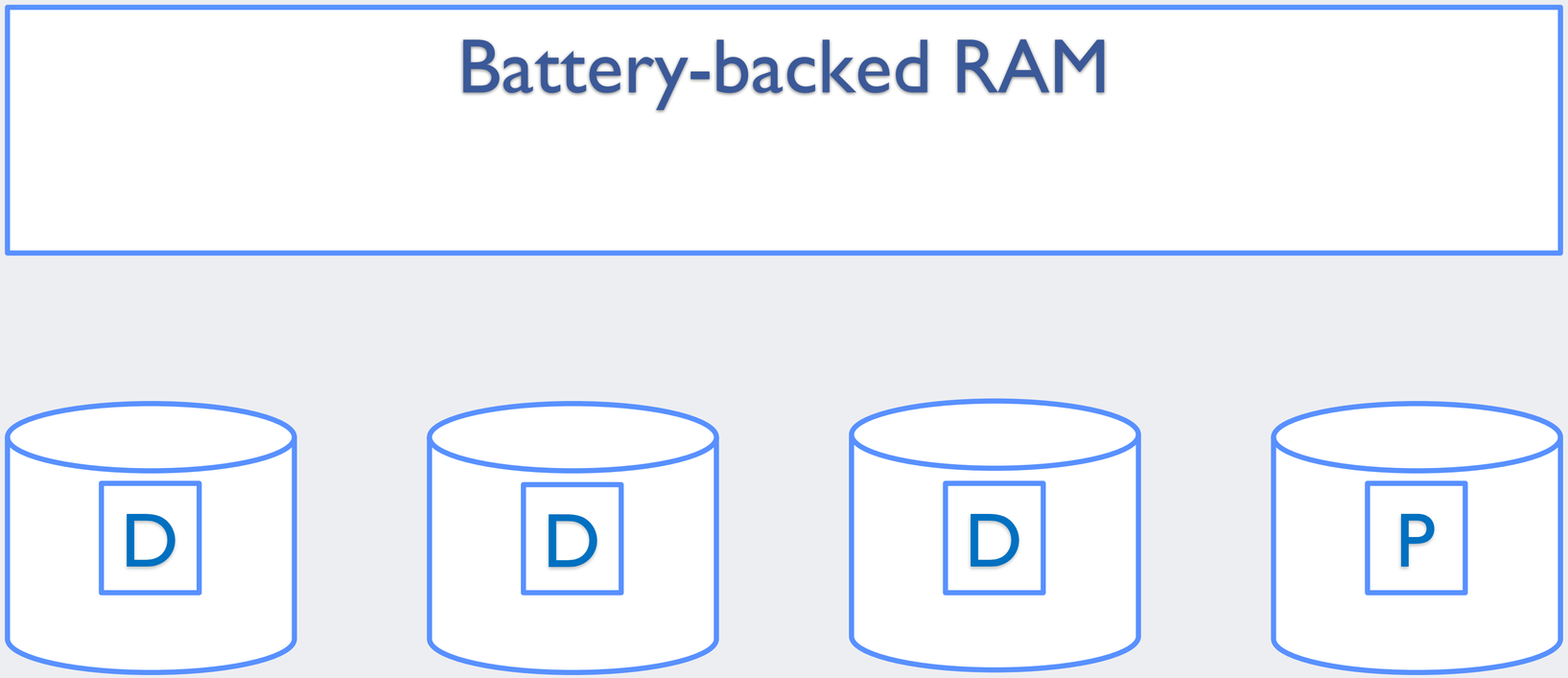
Write Hole: Disk Failure + Power Failure

Write Hole: Rebuild Wrong Data

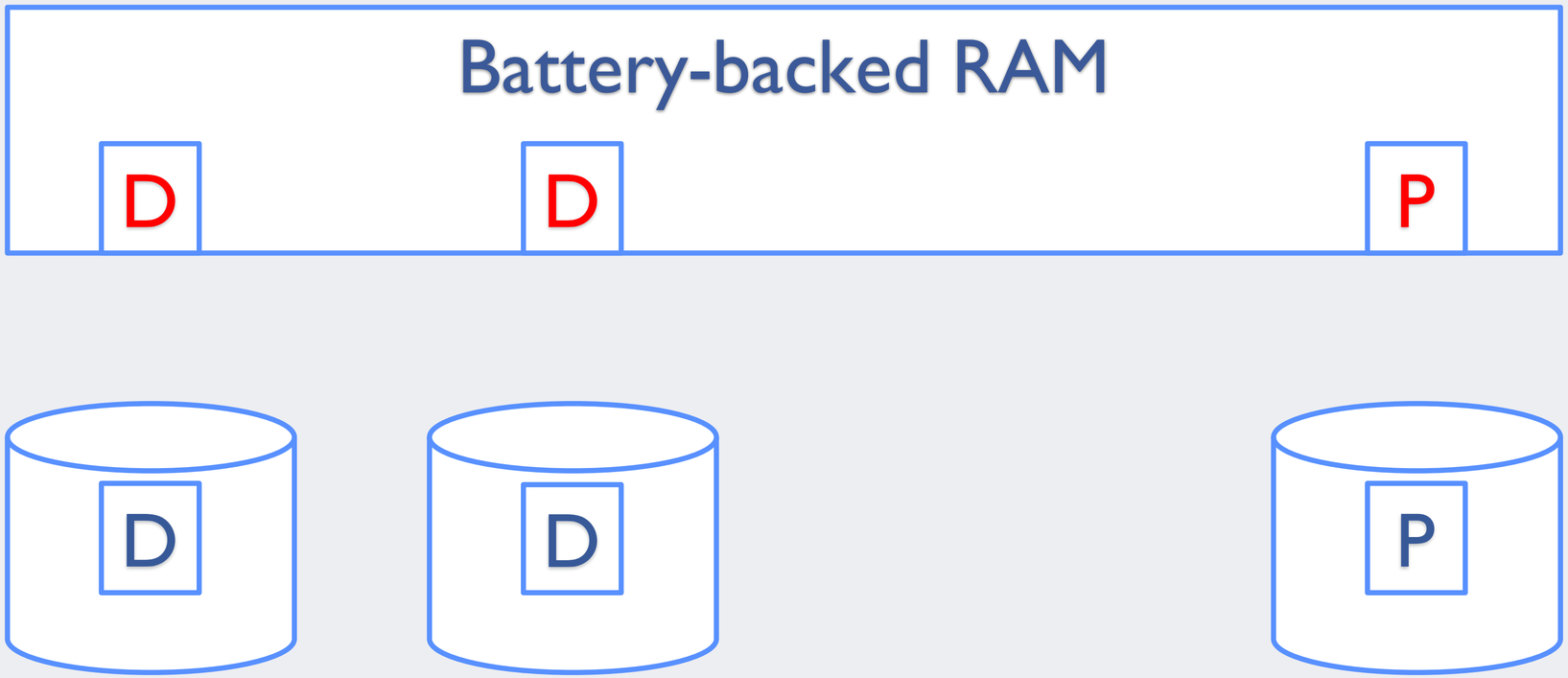


Hardware RAID

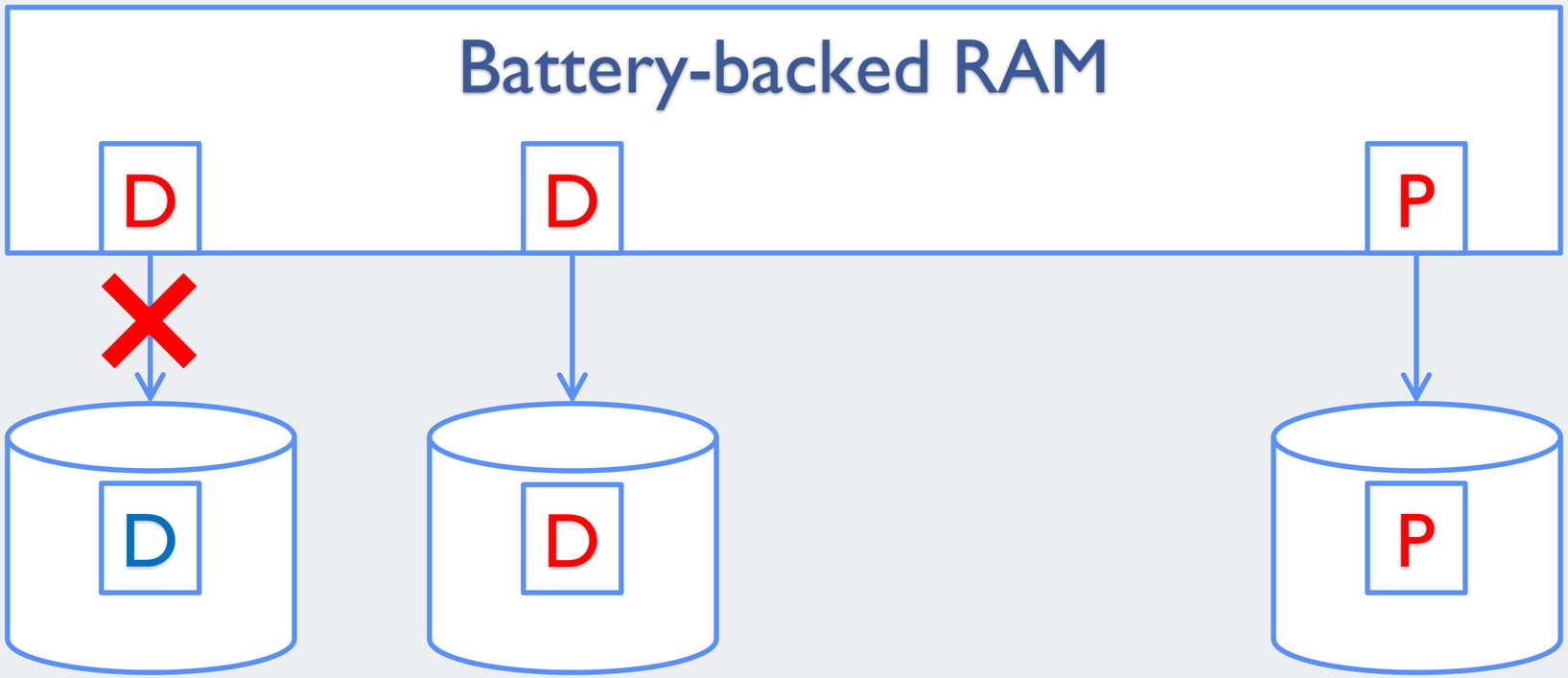
Hardware RAID



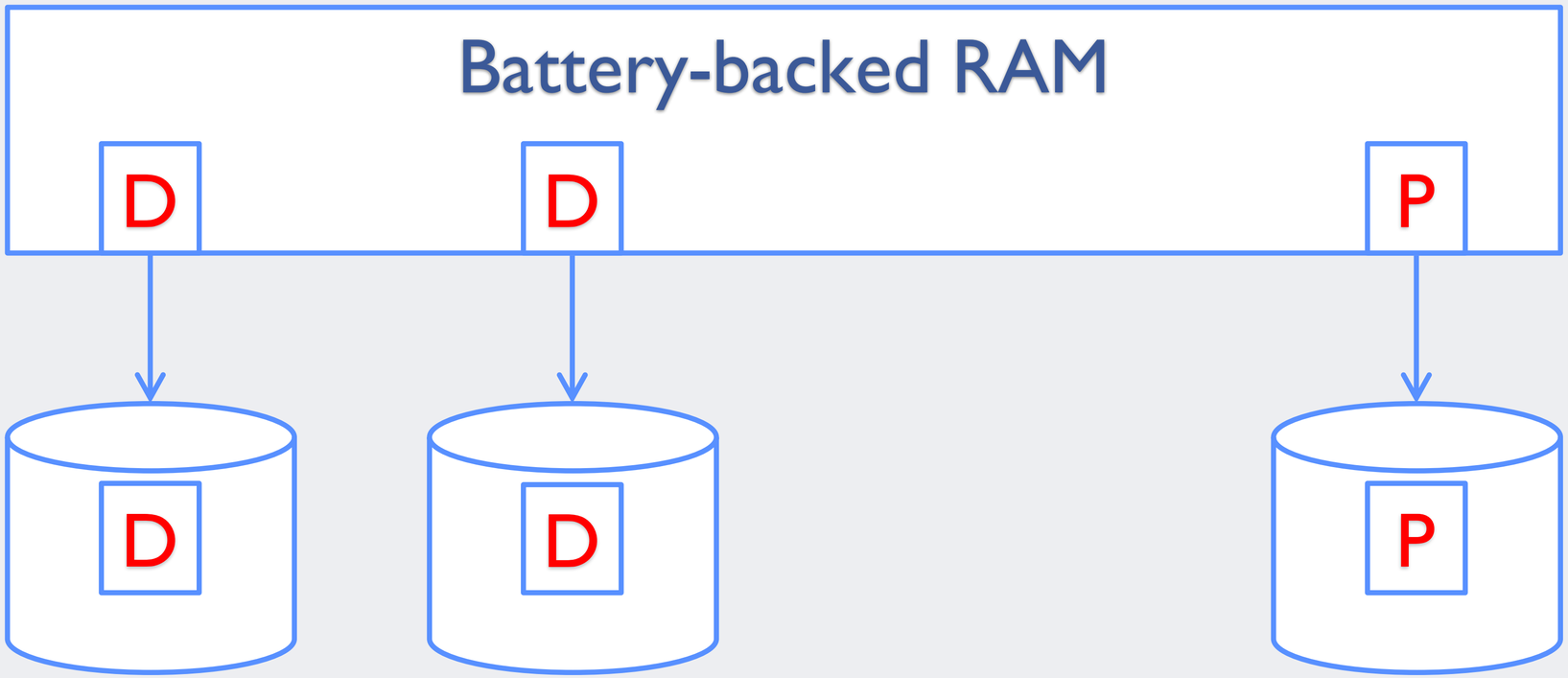
Hardware RAID: No Write Hole



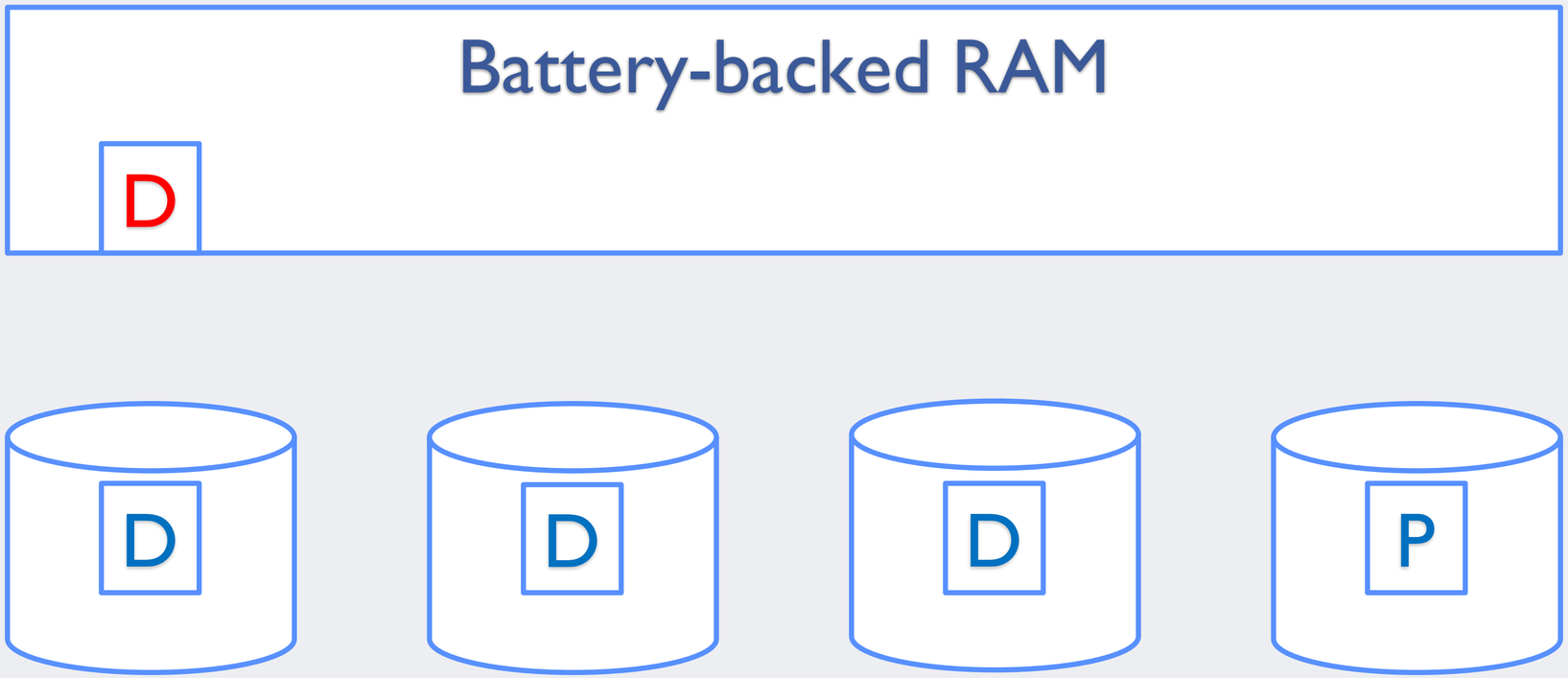
Hardware RAID: No Write Hole



Hardware RAID: No Write Hole

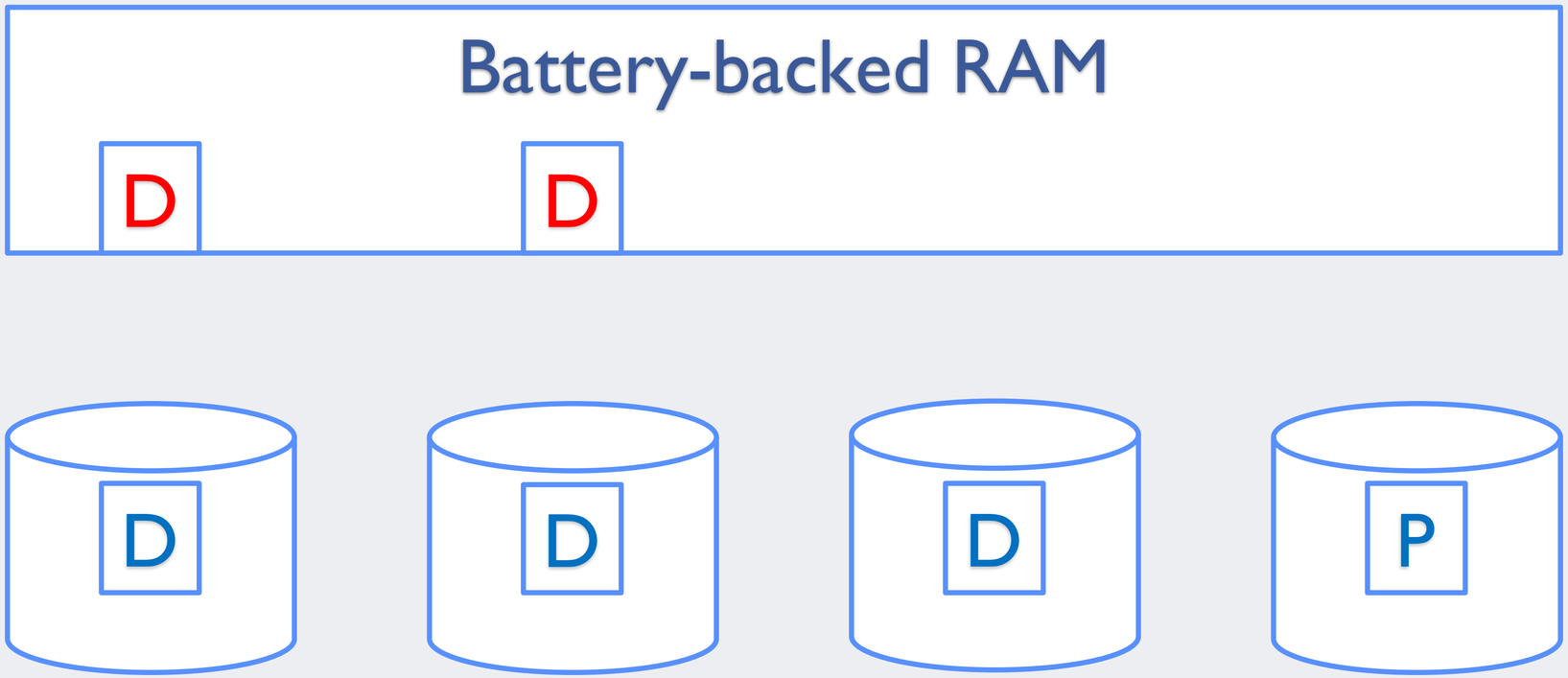


Hardware RAID: Fast fsync()



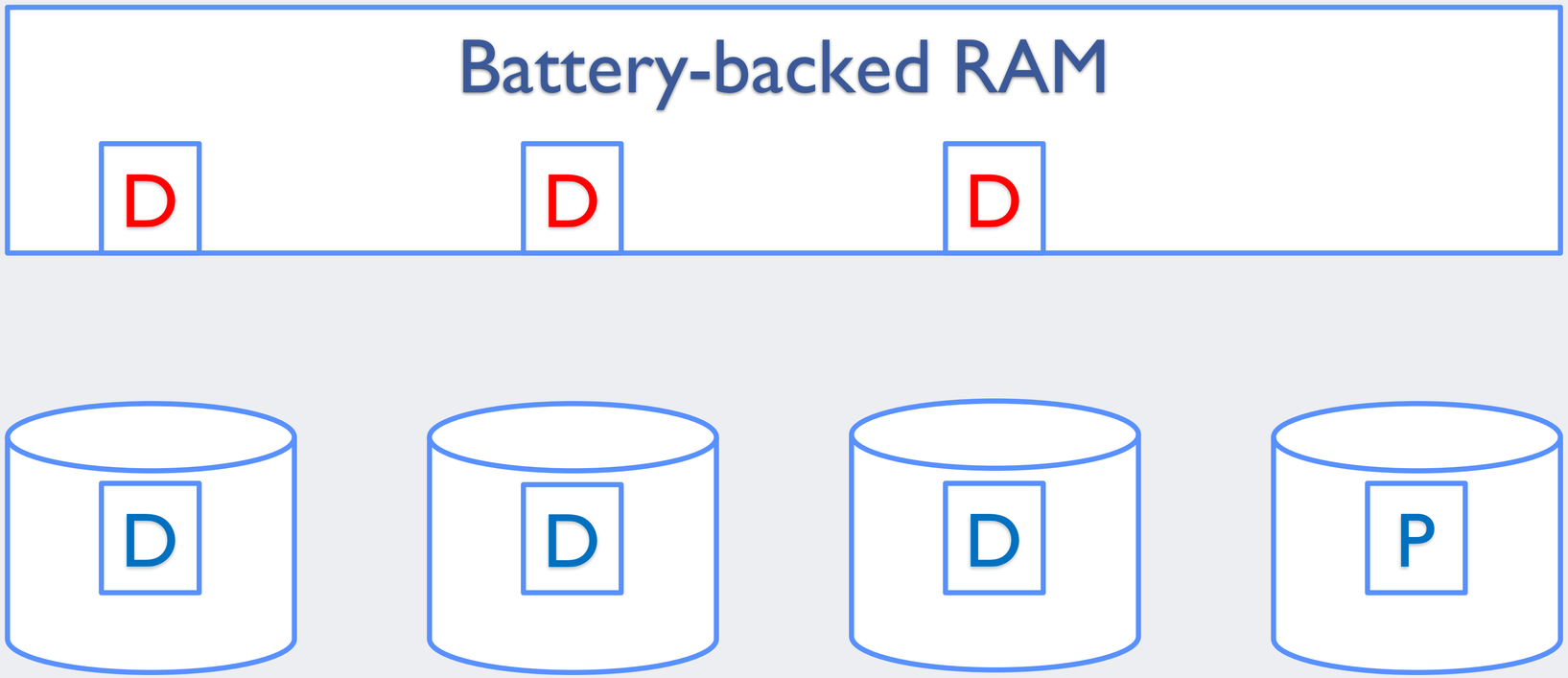
return writes from RAM

Hardware RAID: Fast fsync()

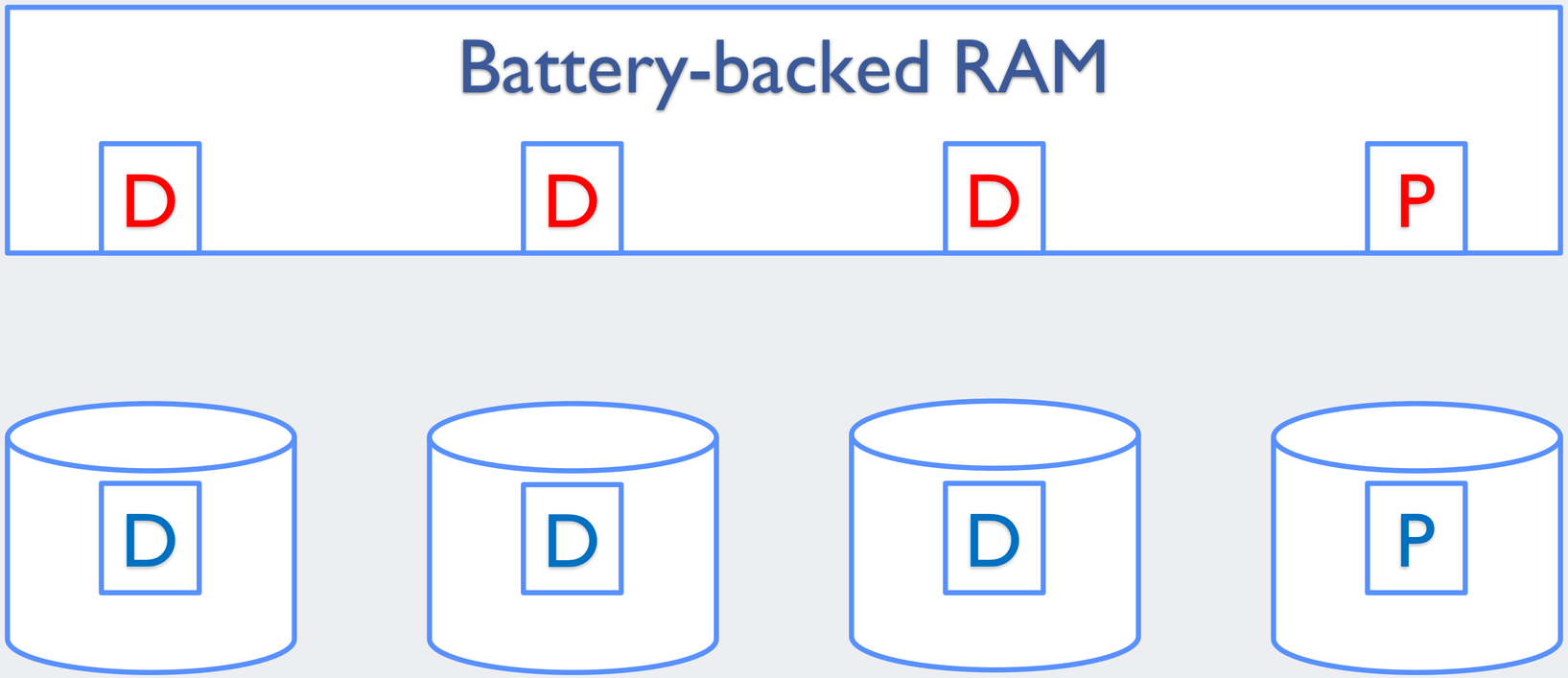


return writes from RAM

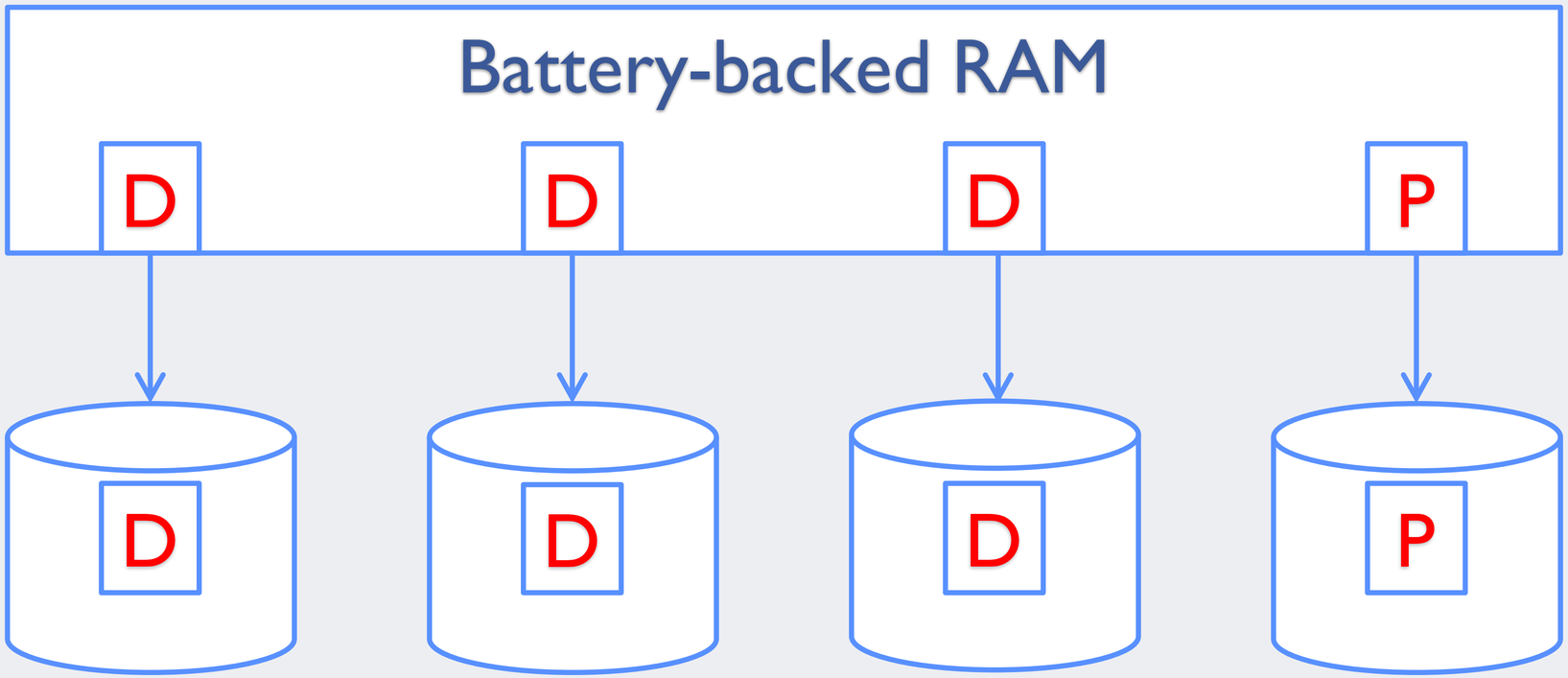
Hardware RAID: Full Stripe Write



Hardware RAID: Full Stripe Write



Hardware RAID: Full Stripe Write



Hardware RAID @ Facebook

- RAID-6
  - Haystack: photo storage
  - GlusterFS: scalable network filesystem
- RAID-0 of single HDD, for fast fsync()

Challenges with Hardware RAID

- Black box solution
  - Low transparency; low flexibility
- Vendor specific toolset

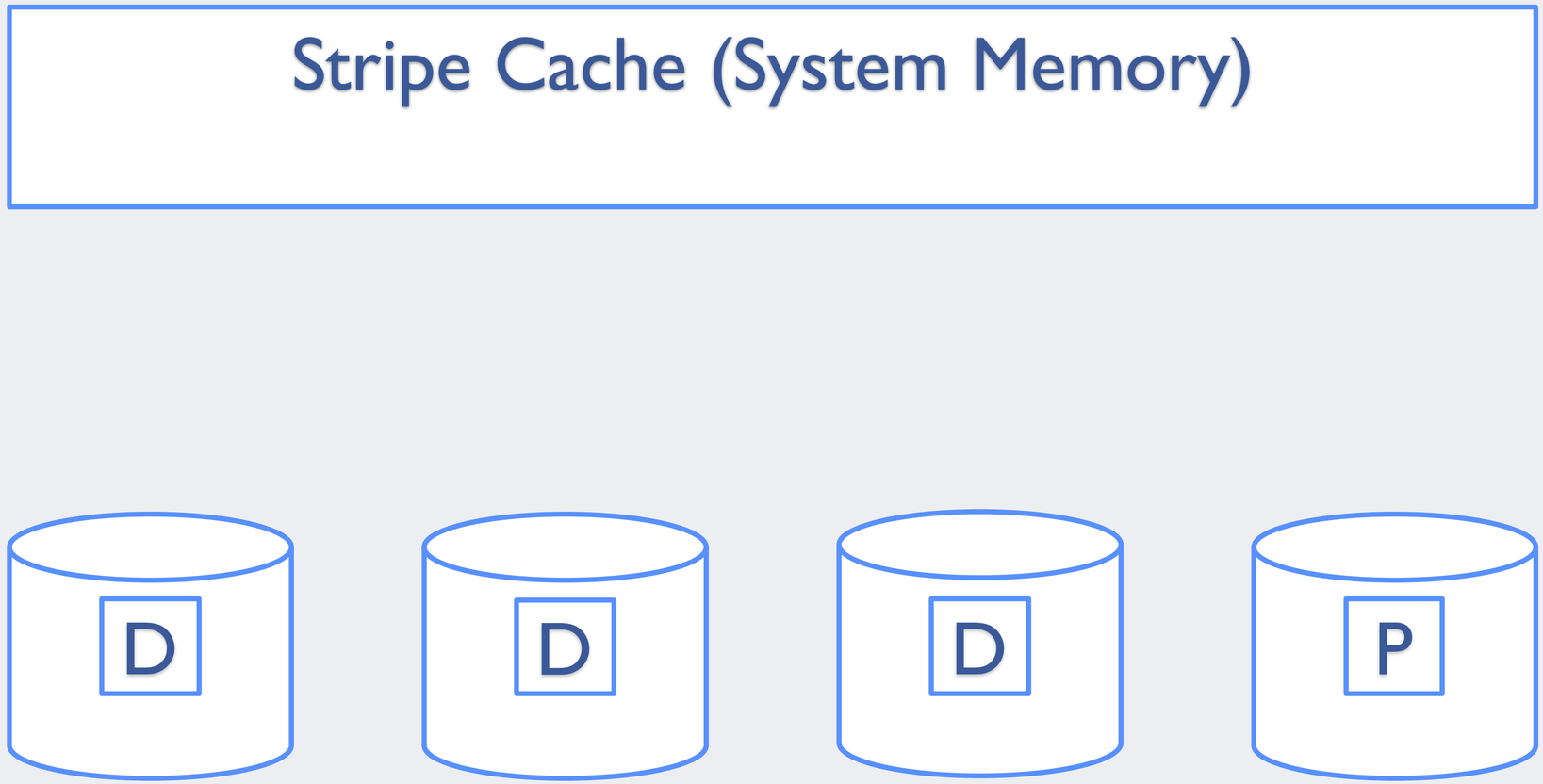
Make Software RAID Better

- Write journal: plug write hole

- Write cache: accelerate fsync(); more full stripe writes

MD/RAID-456 Write

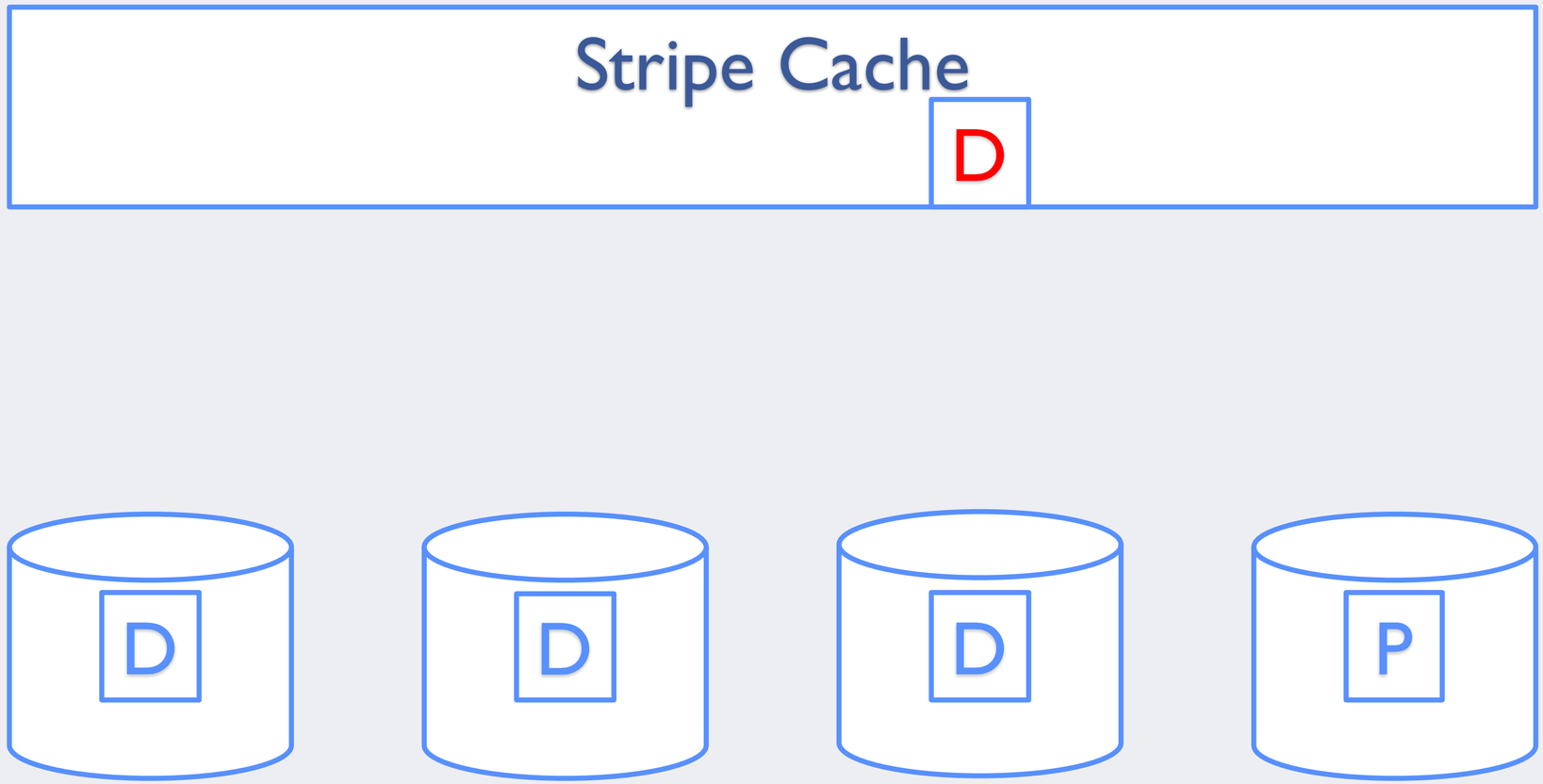
RAID-456: Stripe Cache



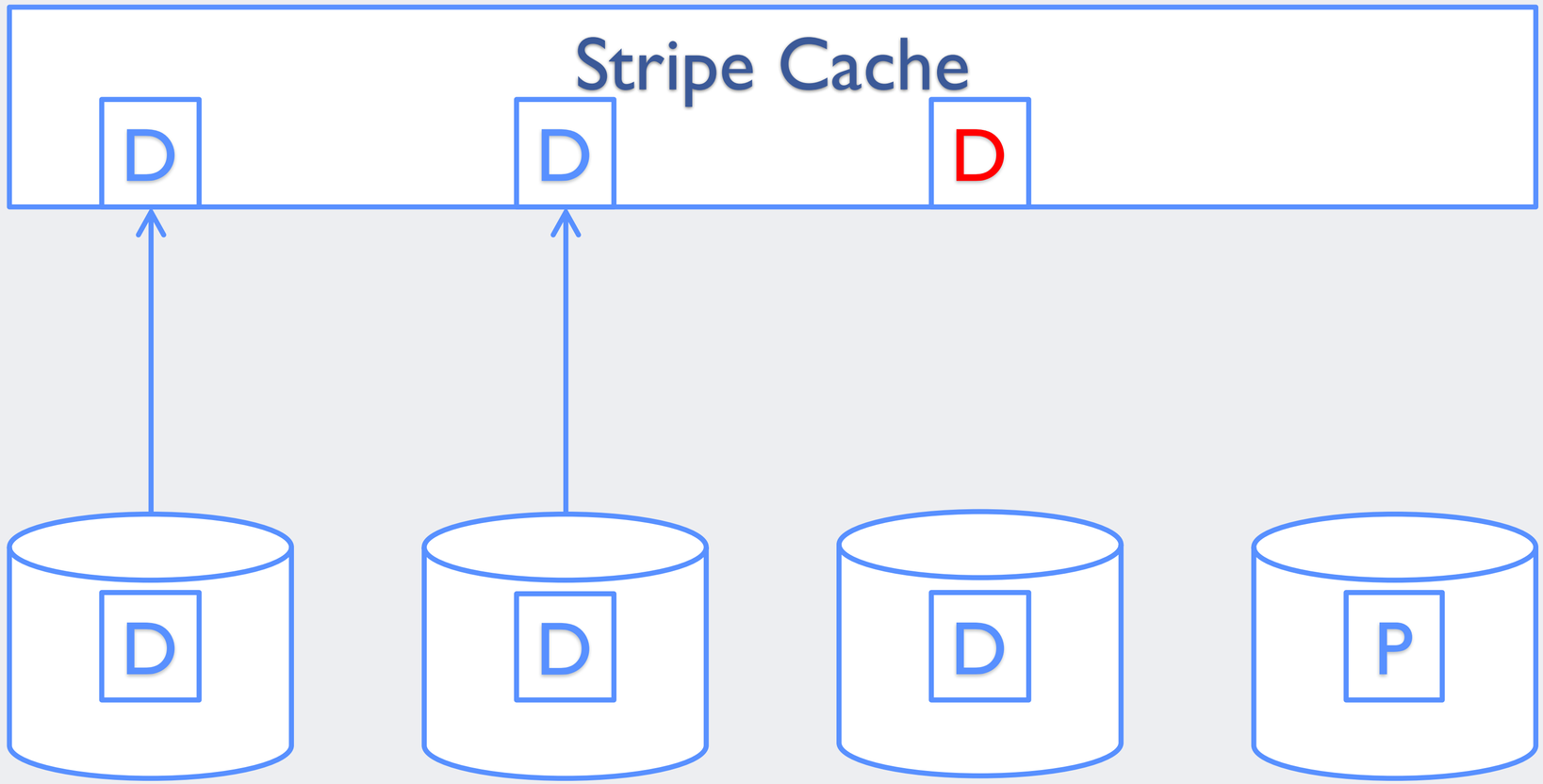
RAID-456 Write

- Step 1: update data and parity in stripe cache
  - Option 1: Reconstruct
  - Option 2: R-M-W
- Step 2: write data and parity to RAID disks
- Step 3: bio_endio

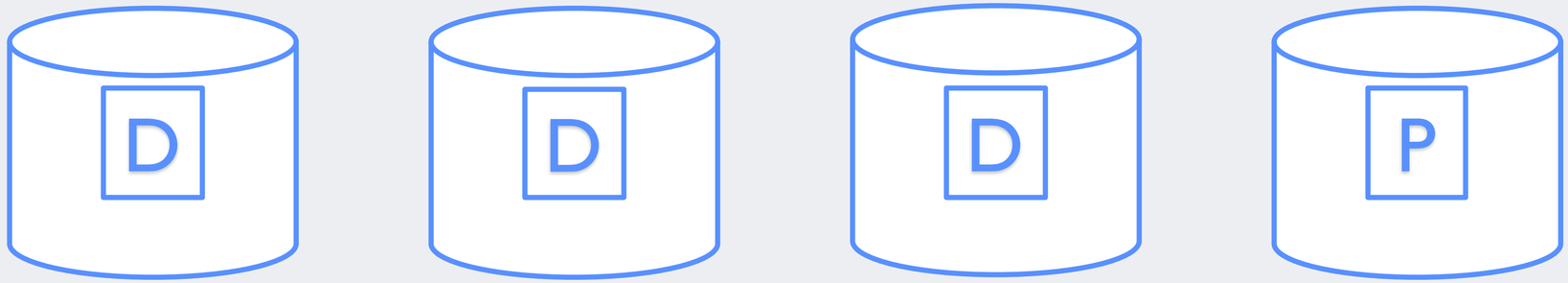
RAID-456 Reconstruct Write



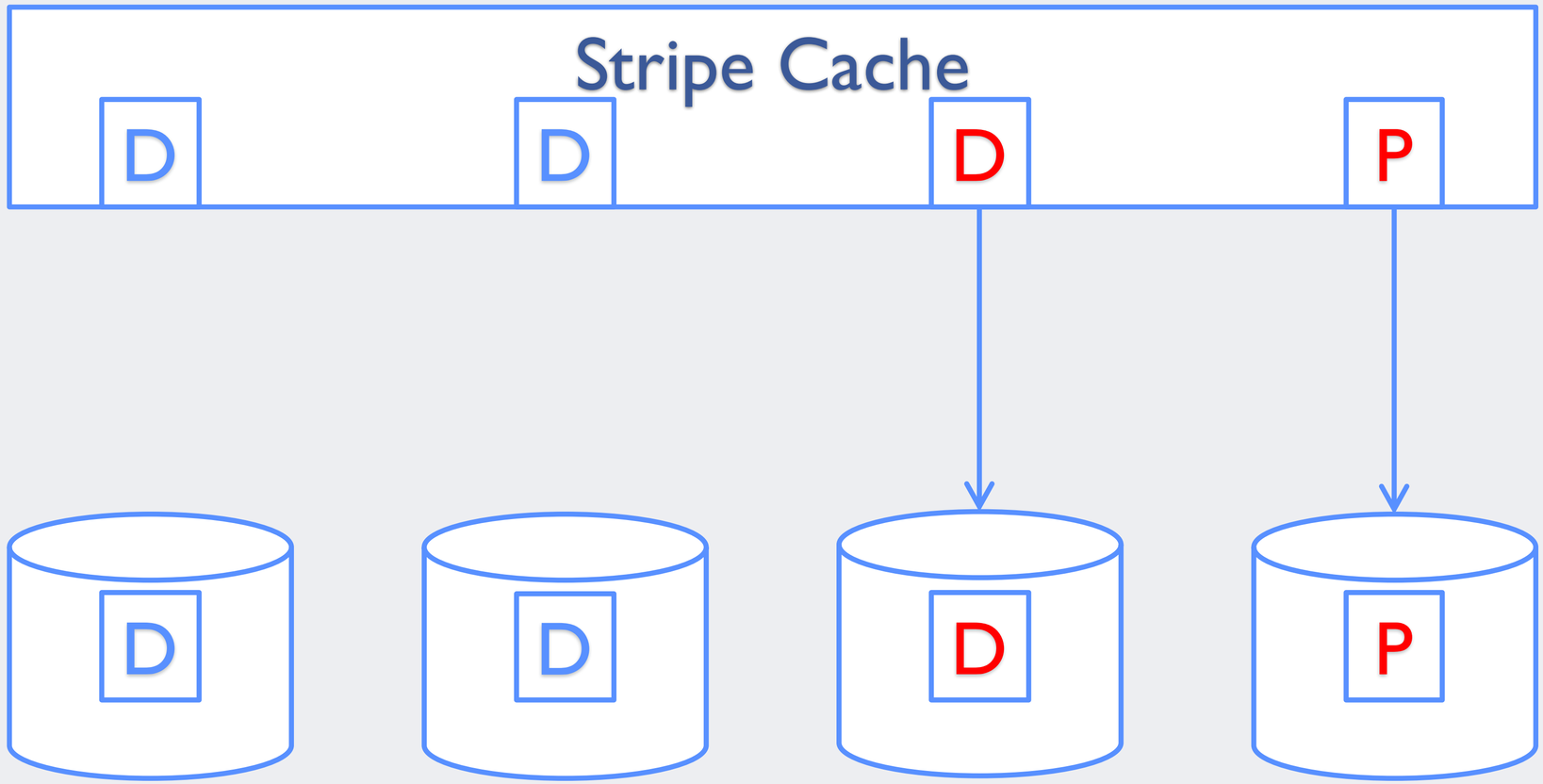
RAID-456 Reconstruct Write



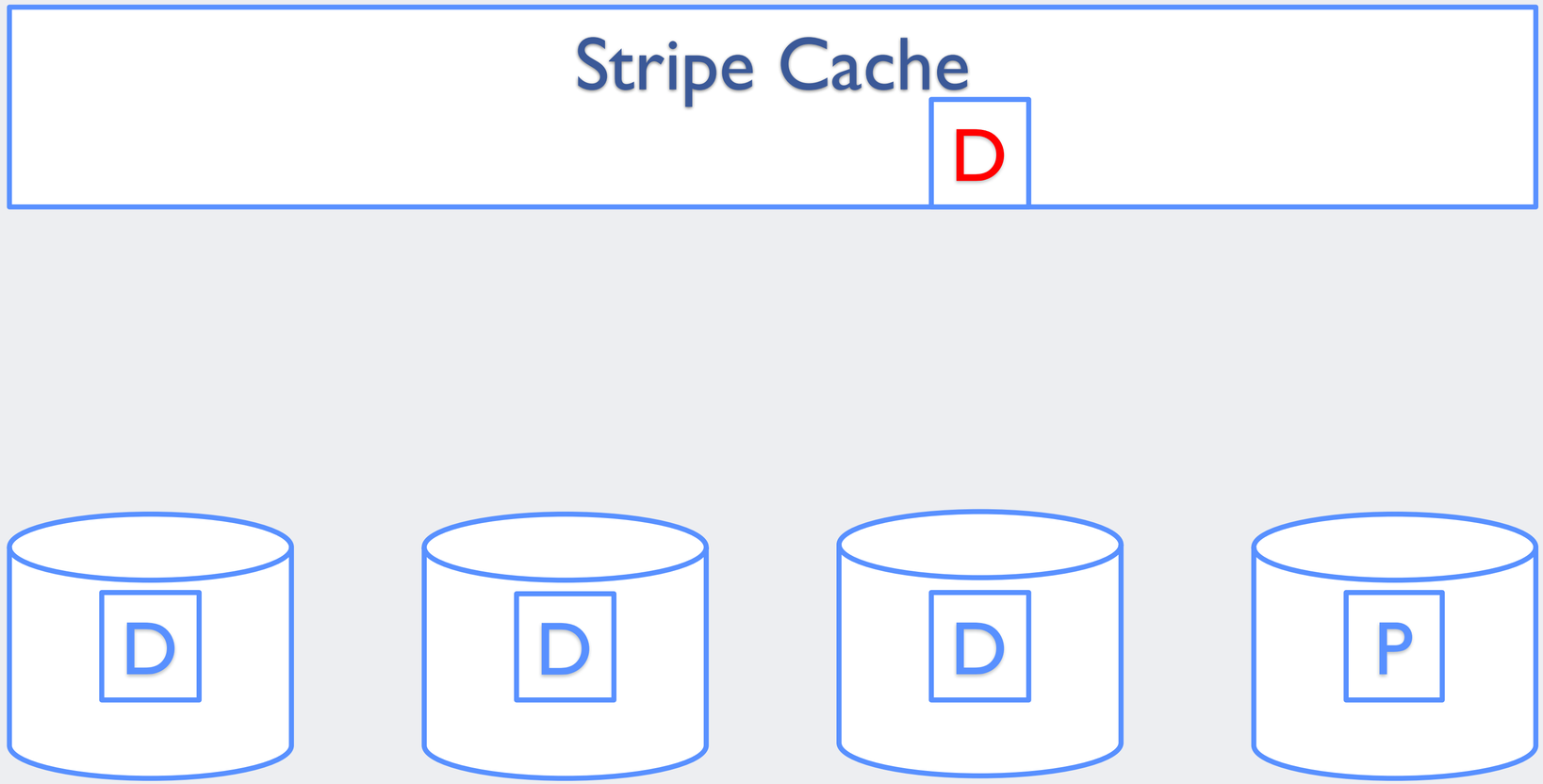
RAID-456 Reconstruct Write



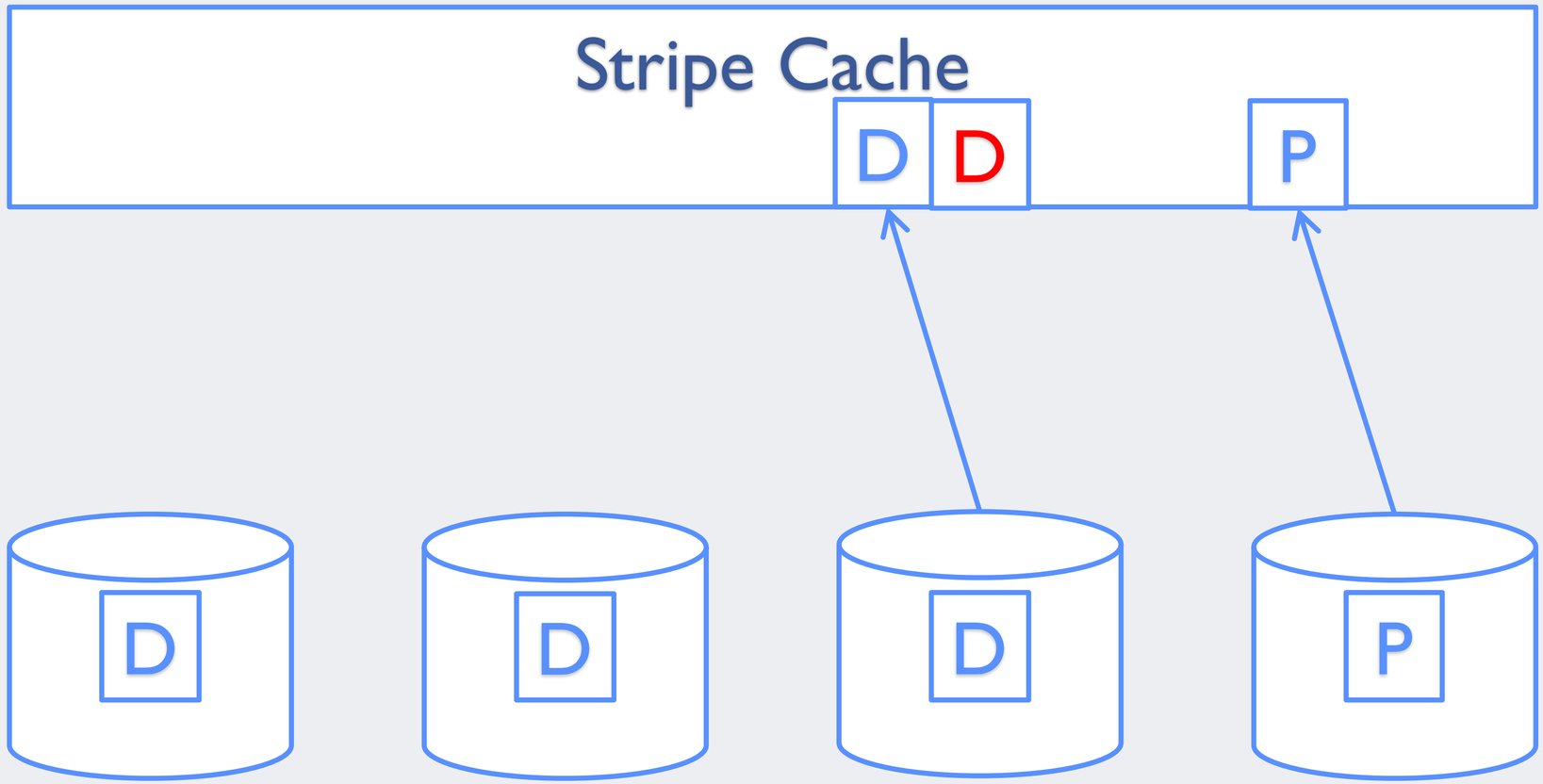
RAID-456 Reconstruct Write



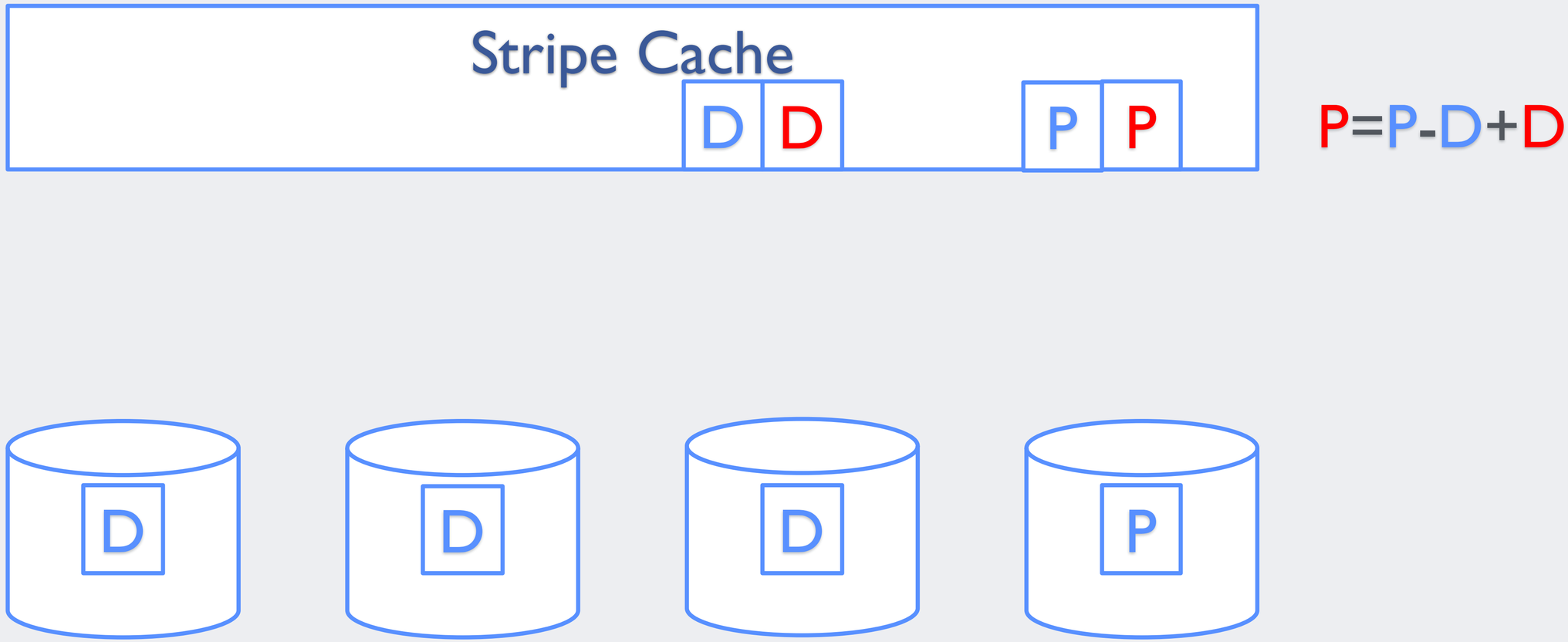
RAID-456 R-M-W Write



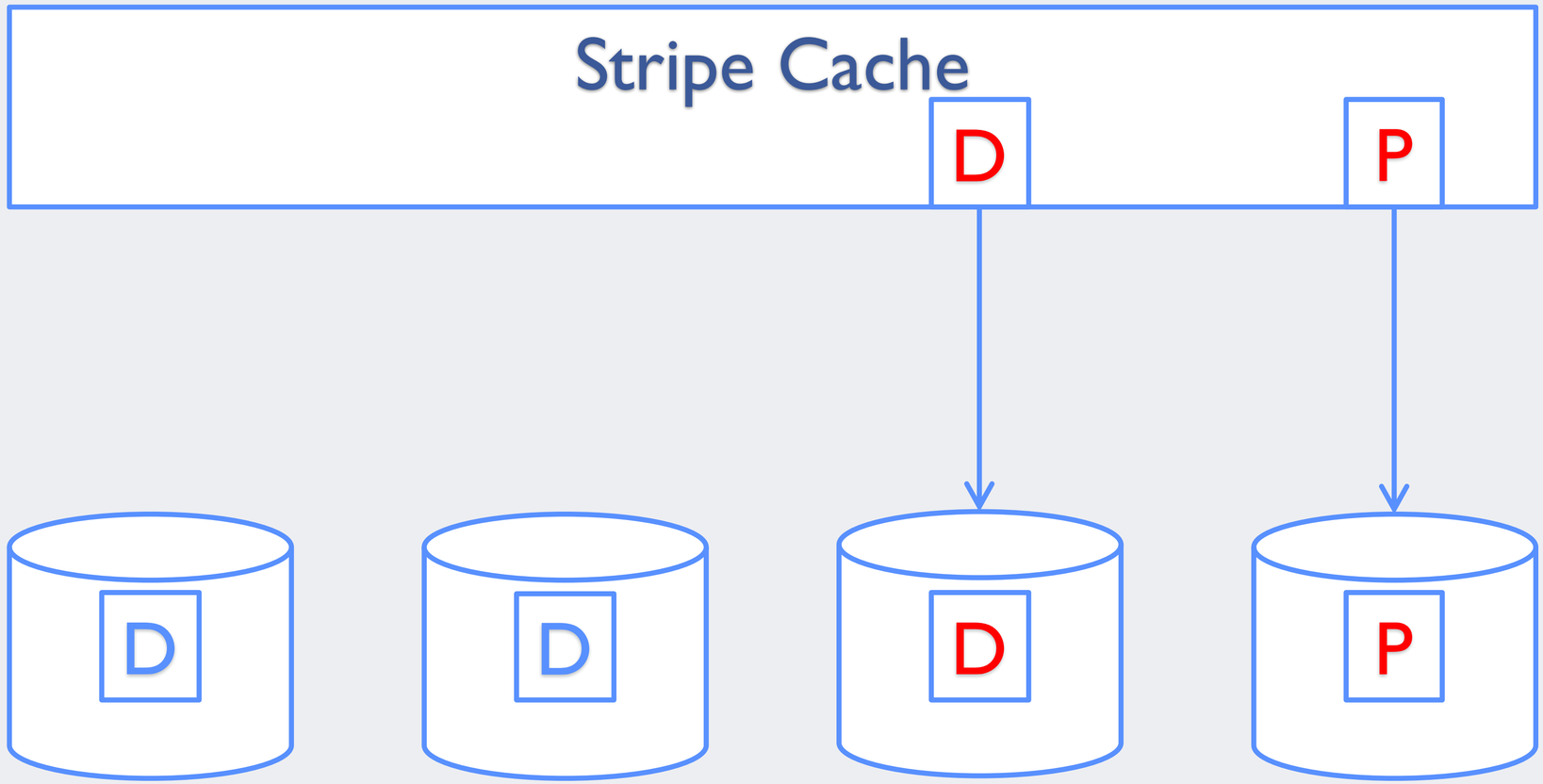
RAID-456 R-M-W Write



RAID-456 R-M-W Write



RAID-456 R-M-W Write



MD/RAID-456 Write Journal

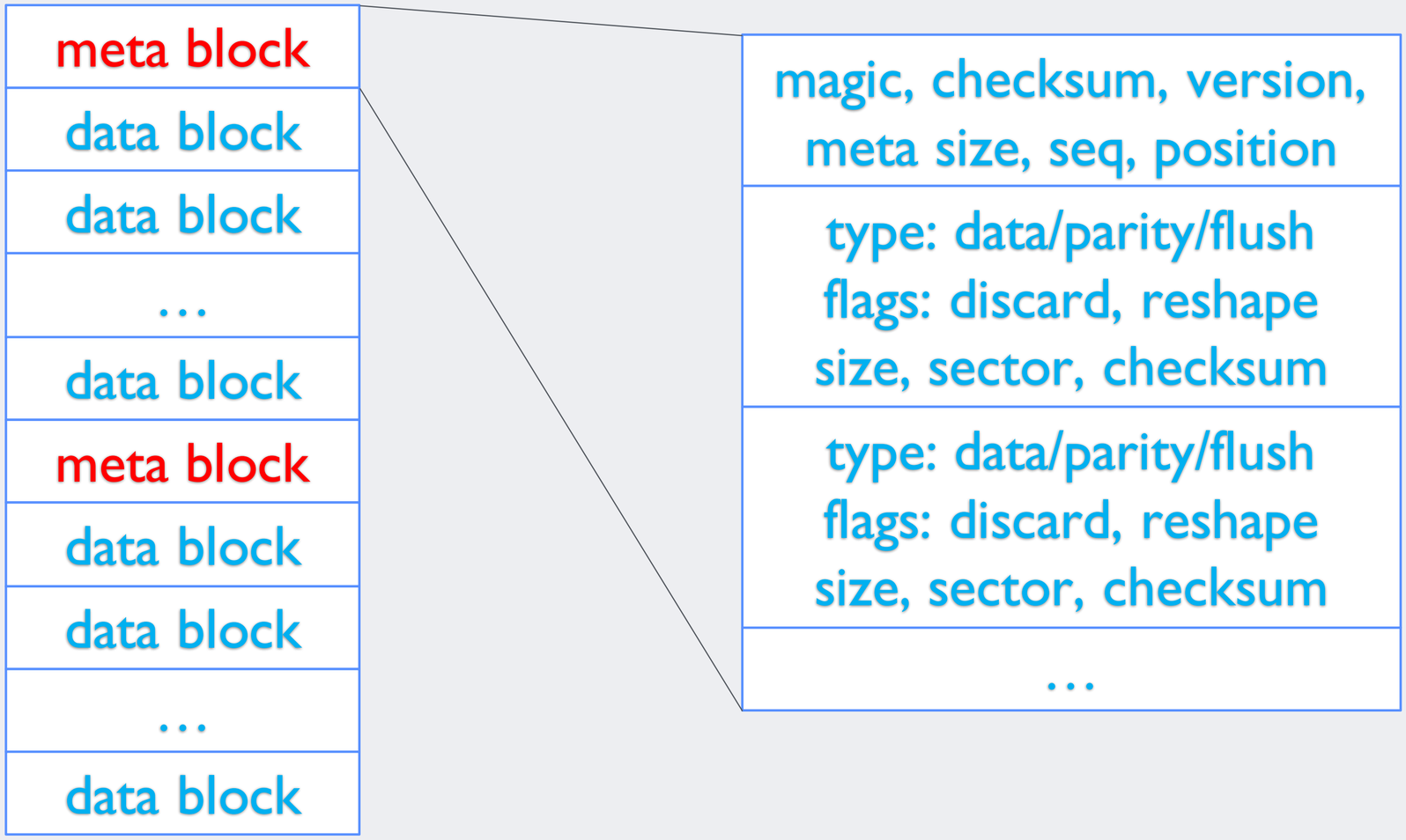
RAID-456 Write Journal

- Use block device (SSD, NVM, etc.) as the journal
- No change to read path
- All writes (data and parity) hit journal before committing to RAID array
- For each stripe, “commit all” or “commit nothing”
- Journal replay after power failure (no need for resync)

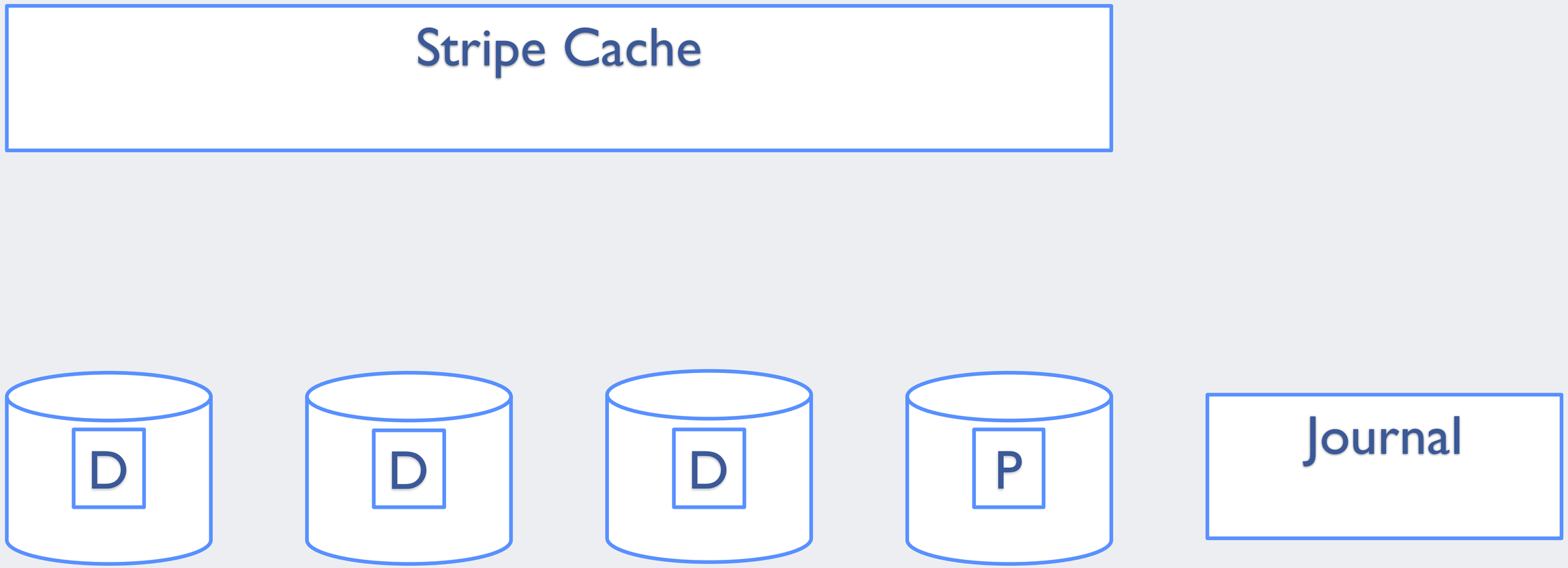
RAID-456 Write Journal: Write Path

- Step 1: update data and parity in stripe cache
- Step 2: write data and parity to journal device
- Step 3: flush journal device cache
- Step 4: write data and parity to RAID disks
- Step 5: bio_endio

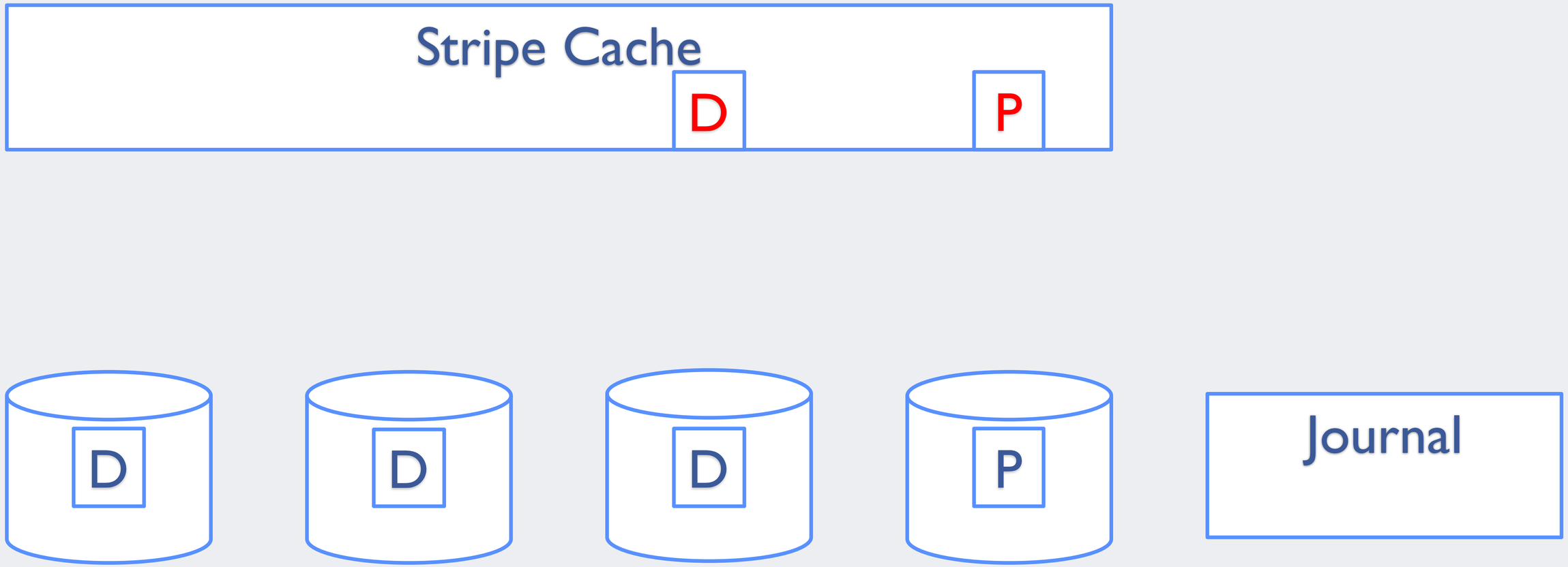
RAID-456 Write Journal: Disk Format



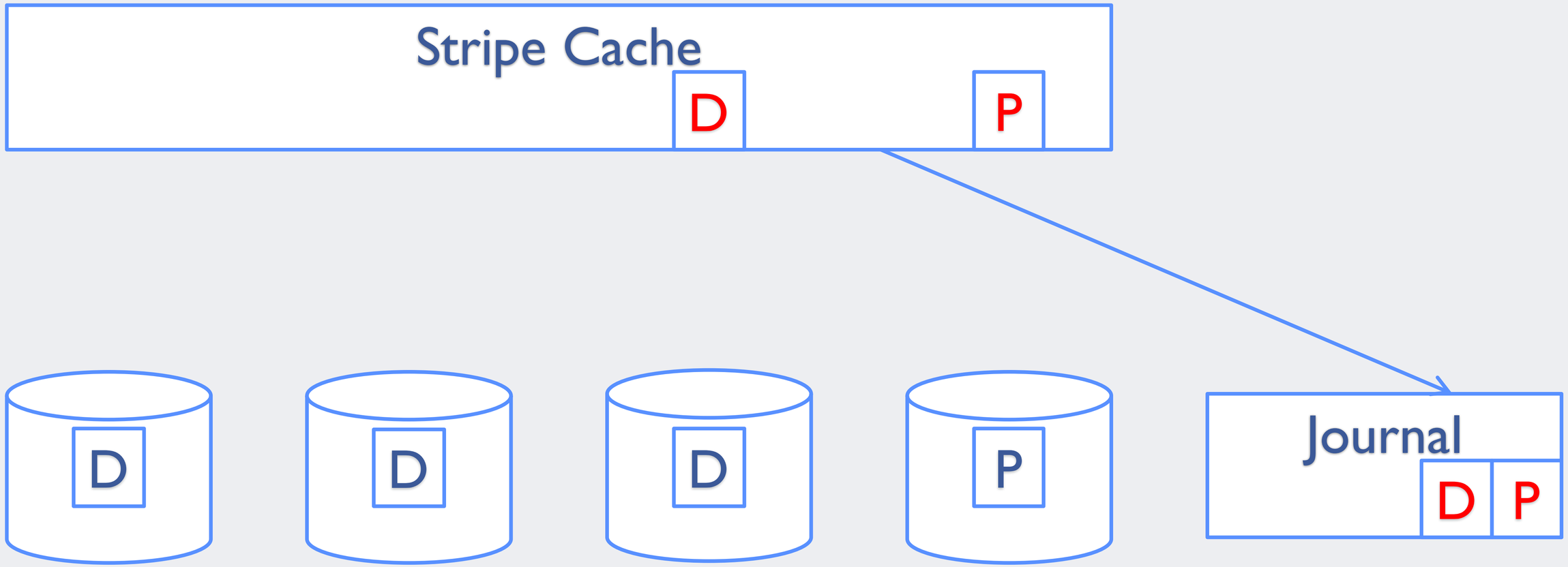
RAID-456 Write Journal



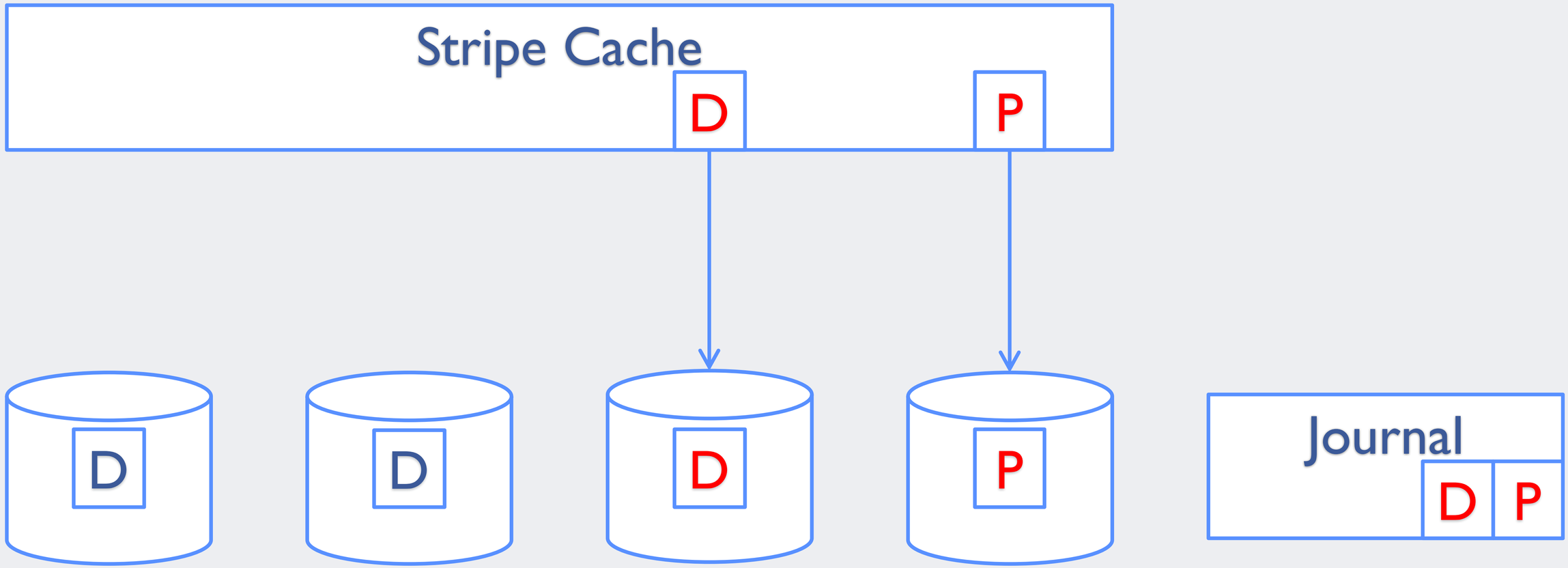
RAID-456 Write Journal



RAID-456 Write Journal



RAID-456 Write Journal



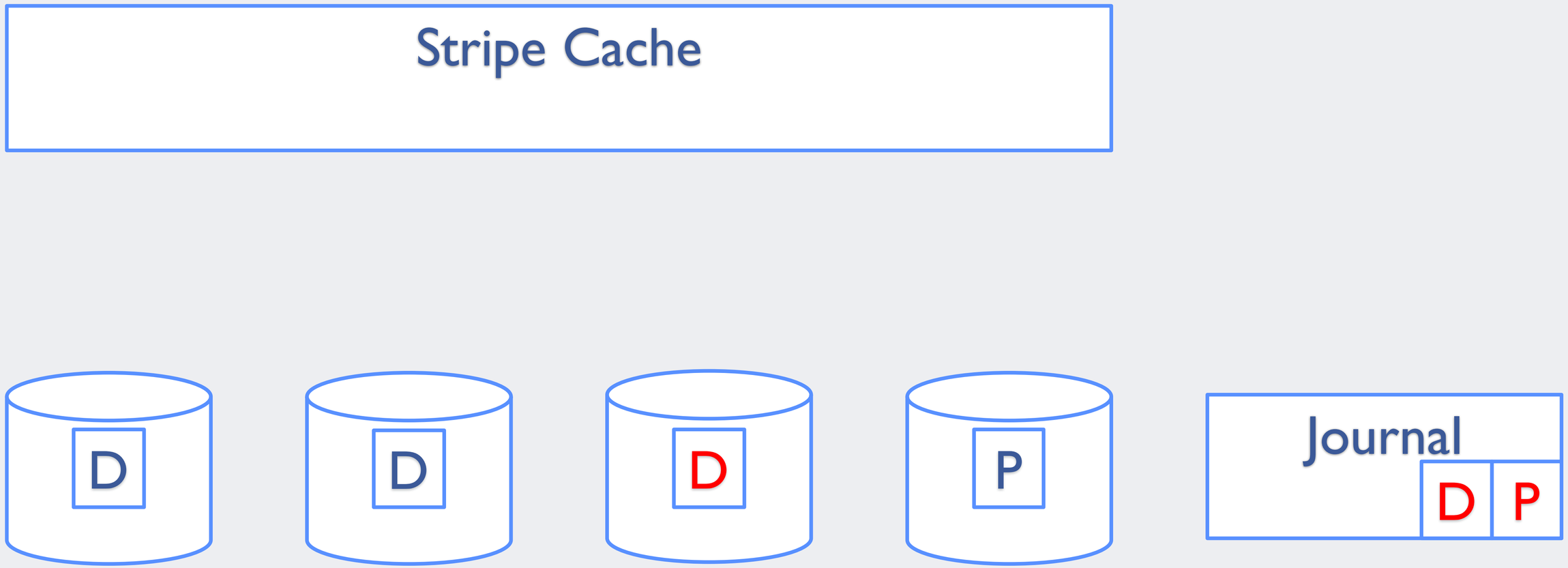
RAID-456 Write Journal: Reclaim Path

- Step 1: update journal device super block
- Step 2: issue discard to journal device

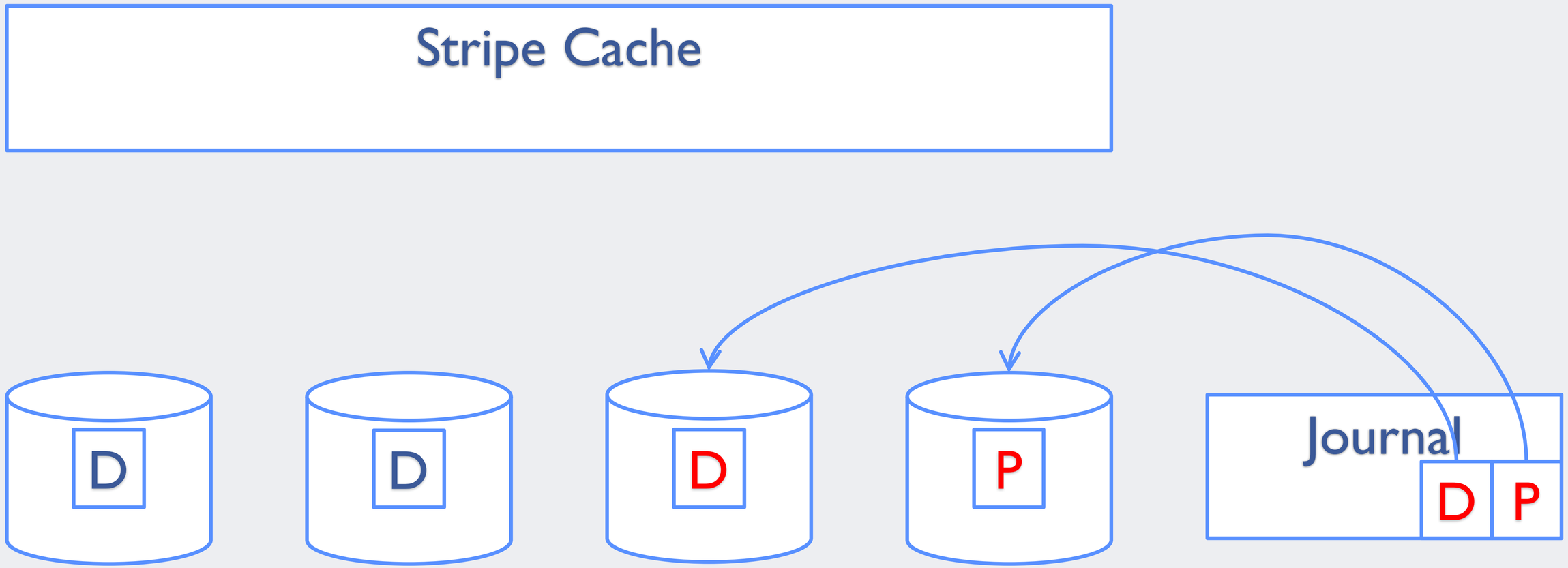
RAID-456 Write Journal: Recovery Path

- For complete stripe (with data and parity) in journal
  - Replay all data/parity to RAID disks
- For partial stripe (data only) in journal
  - Drop the journal entry

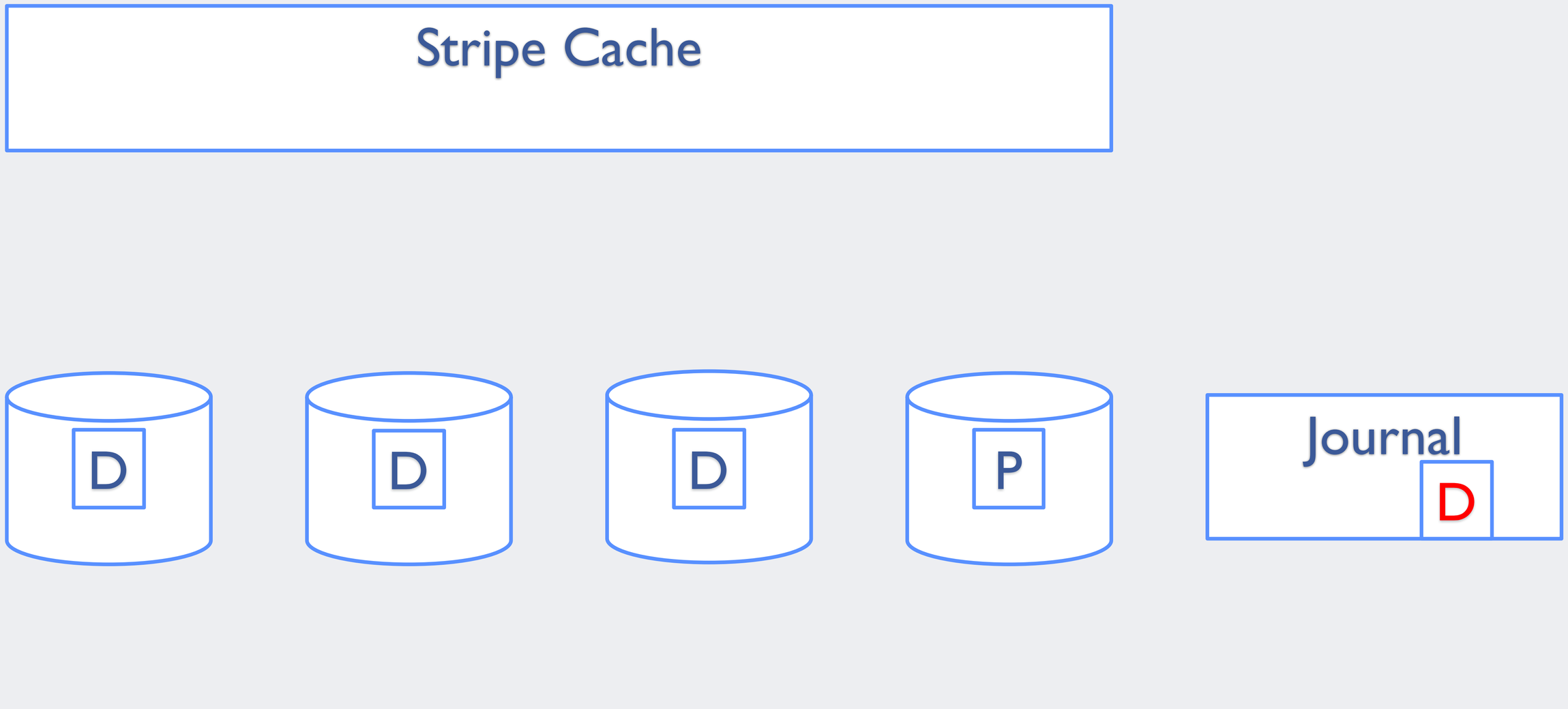
After Power Failure: Replay Writes



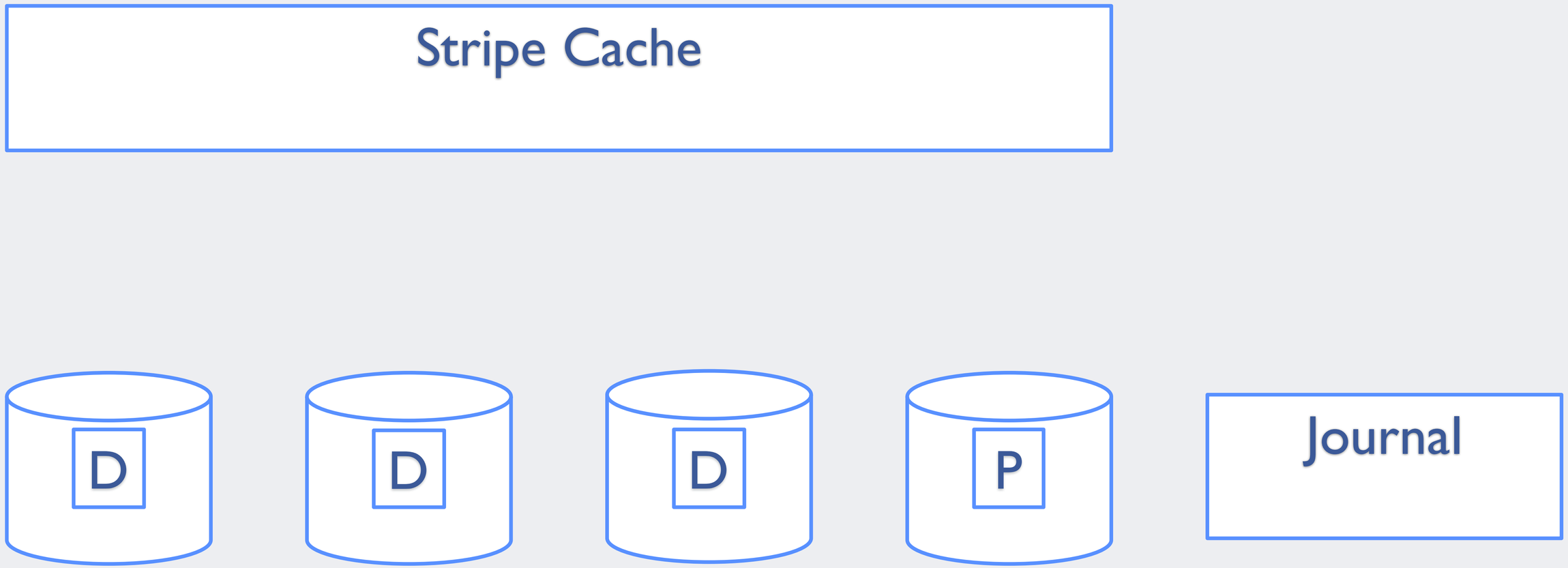
After Power Failure: Replay Writes



After Power Failure: Drop Partial Journal



After Power Failure: Drop Partial Journal



MD/RAID-456 Write Cache

RAID-456 Write Cache

- Use same disk format as the write journal
- Move bio_endio to a much earlier stage
- Hold data in stripe cache
  - Read path must look up in stripe cache
  - Need reclaim and smarter recovery
  - Opportunity for more full stripe writes

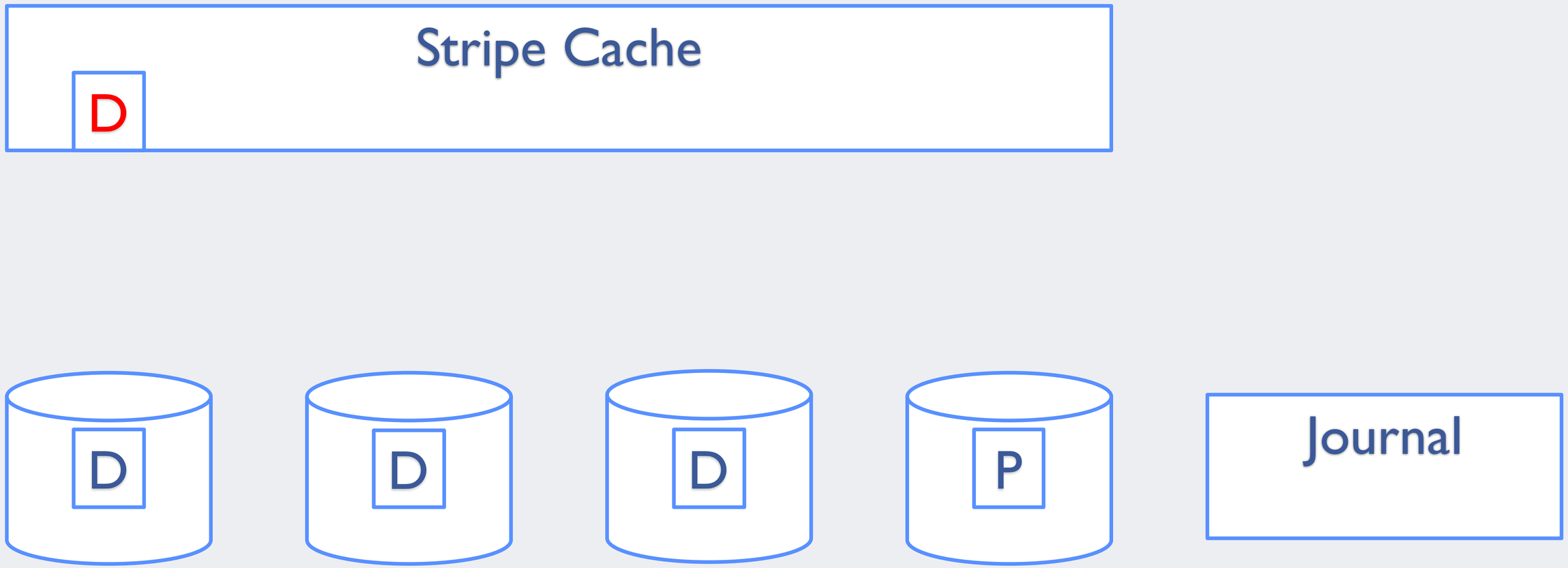
RAID-456 Write Cache: Read Path

- Chunk aligned read (bypass stripe cache, optimal state)
  - Step 1: look up data in stripe cache
  - Step 2: when missed stripe cache, read from disk
  - Step 3: amend data from disk with latest data in stripe cache
- None chunk aligned read
  - No changes

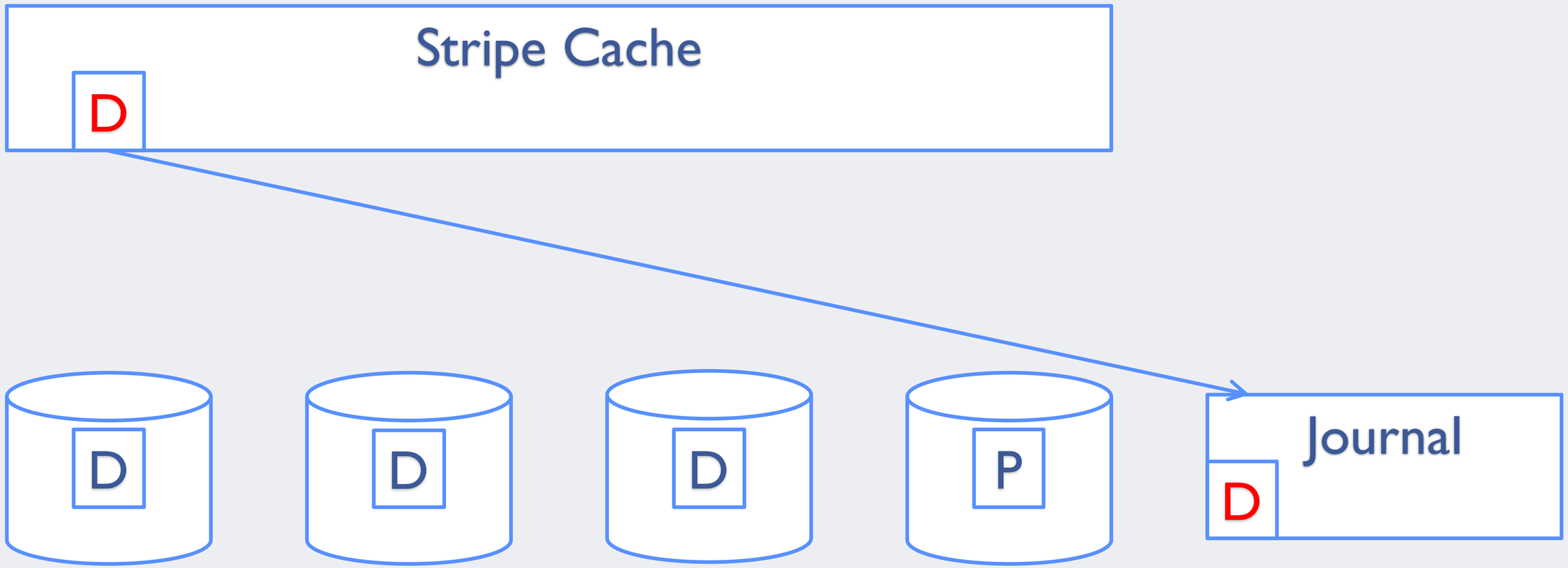
RAID-456 Write Cache: Write Path

- Step 1: write data to journal device
- Step 2: flush journal device cache
- Step 3: bio_endio
- ~~Step 4: update data and parity in stripe cache~~
- ~~Step 5: write parity to journal device~~
- ~~Step 6: flush journal device cache~~
- ~~Step 7: write data and parity to RAID disks~~

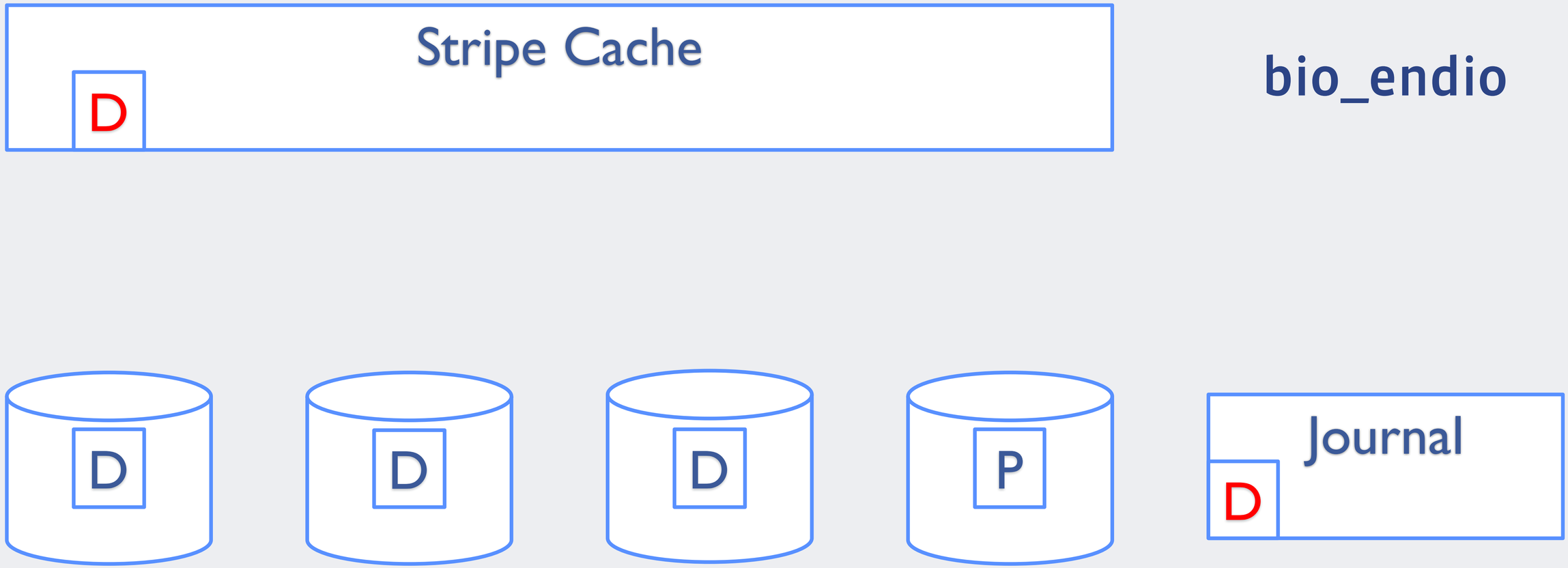
RAID-456 Write Cache: Write Path



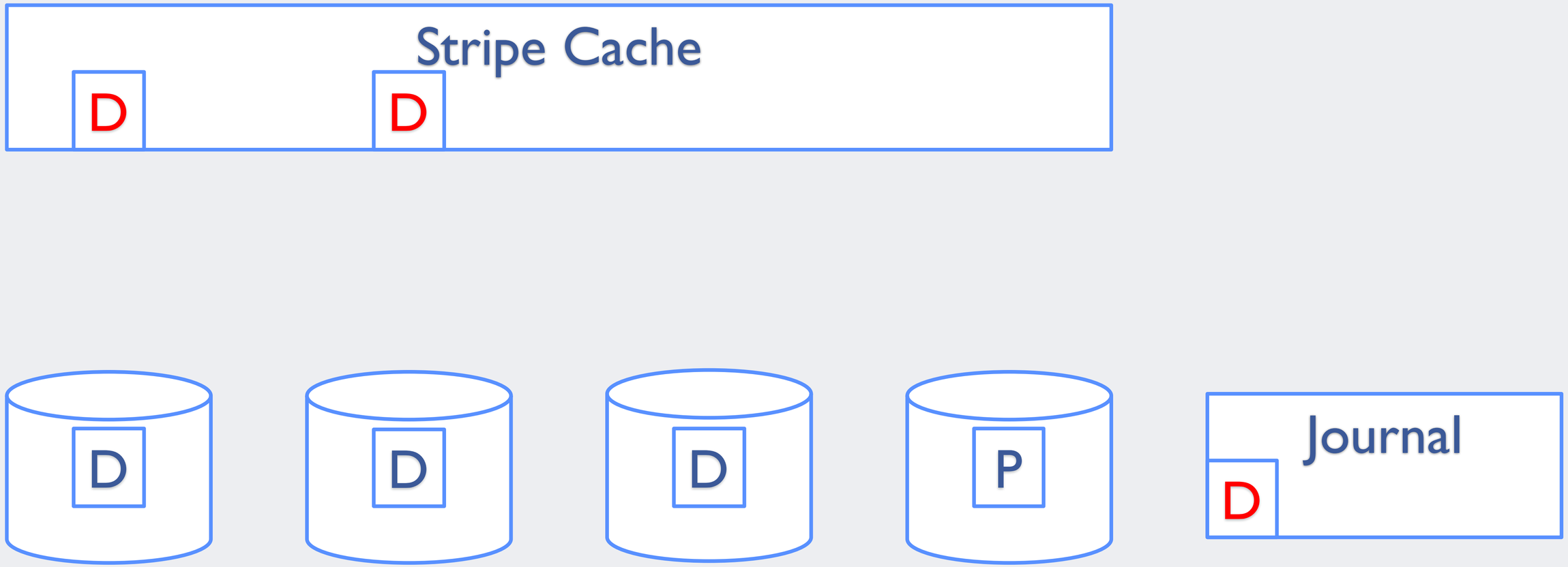
RAID-456 Write Cache: Write Path



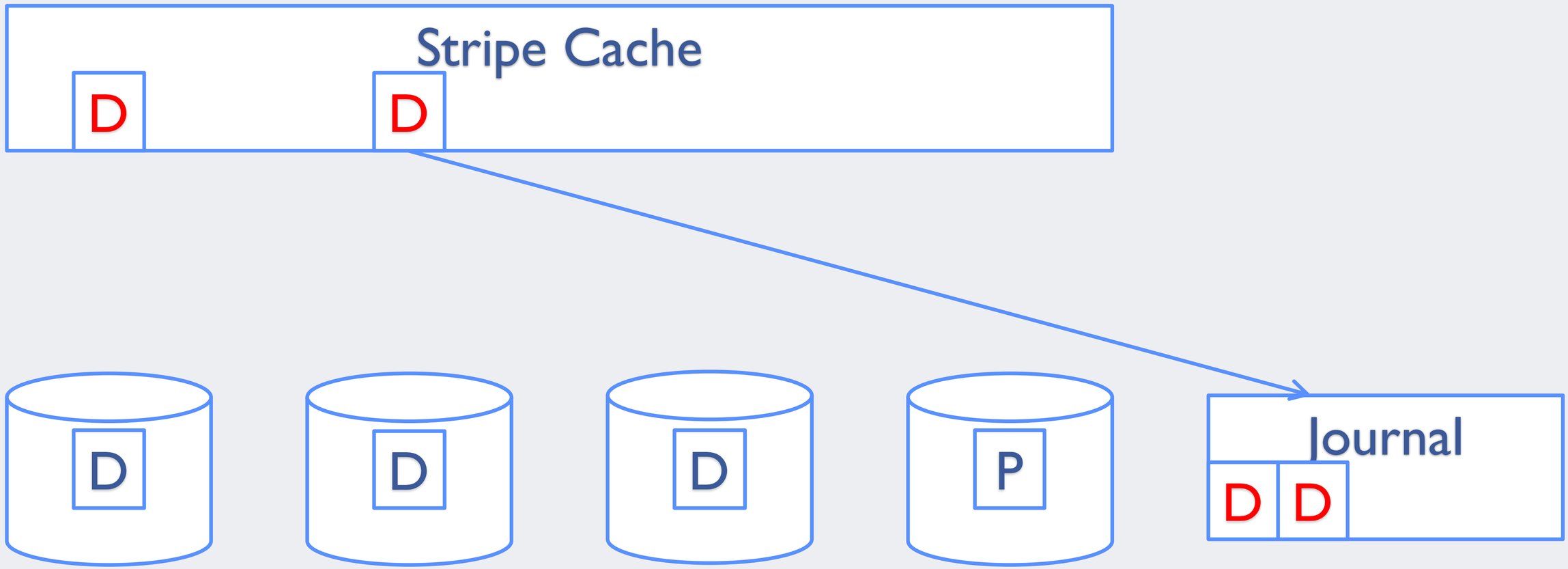
RAID-456 Write Cache: Write Path



RAID-456 Write Cache: Write Path

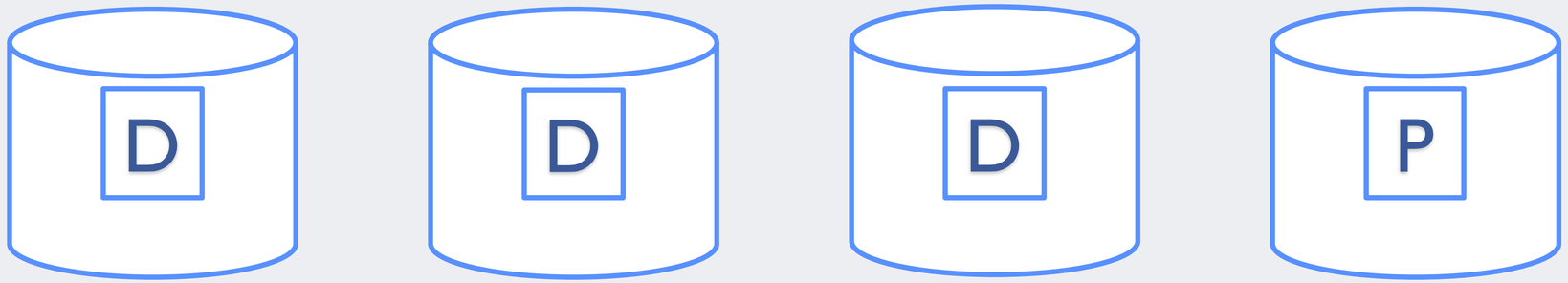


RAID-456 Write Cache: Write Path



RAID-456 Write Cache: Write Path

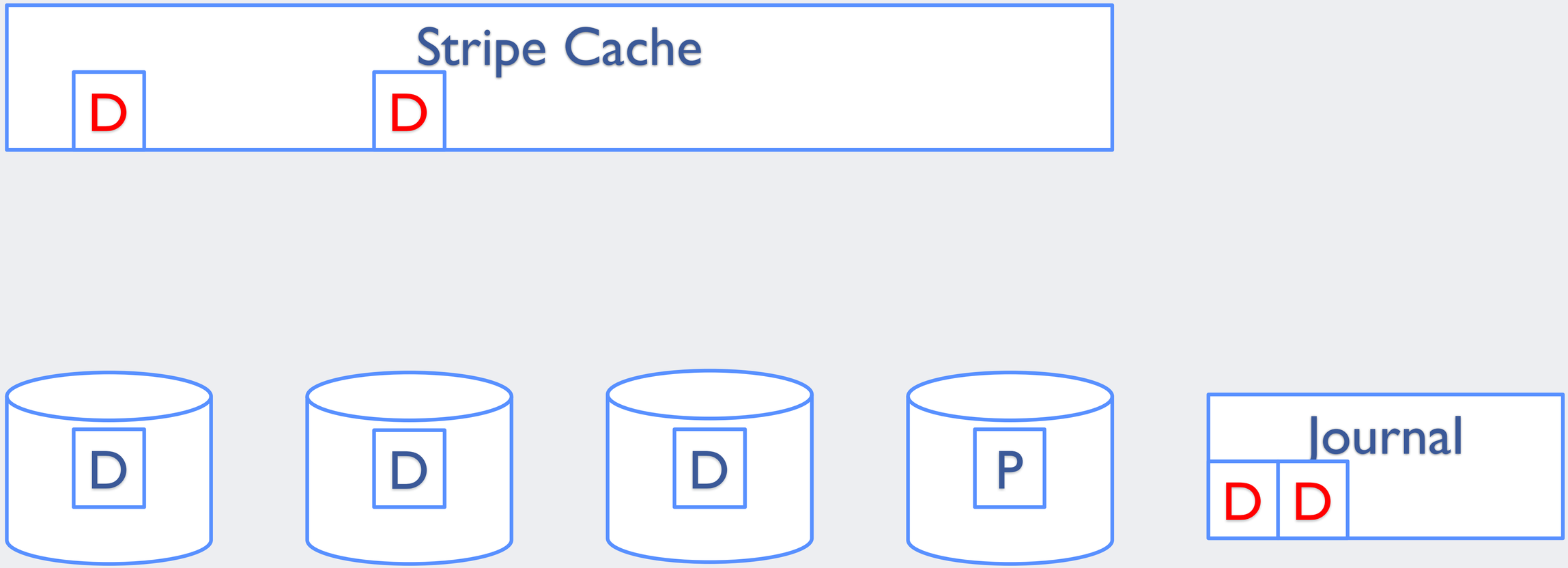
bio_endio



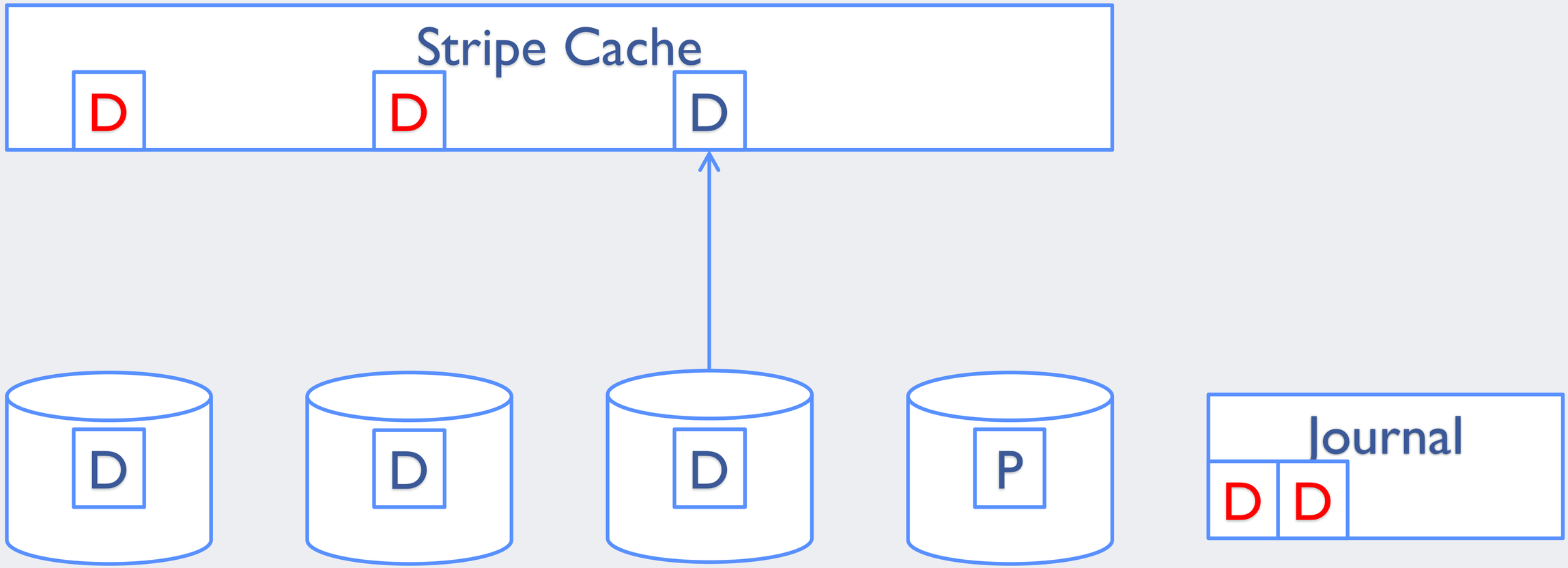
RAID-456 Write Cache: Reclaim Path

- Step 1: update data and parity in stripe cache
- Step 2: write parity to journal device
- Step 3: flush journal device cache
- Step 4: write data and parity to RAID disks

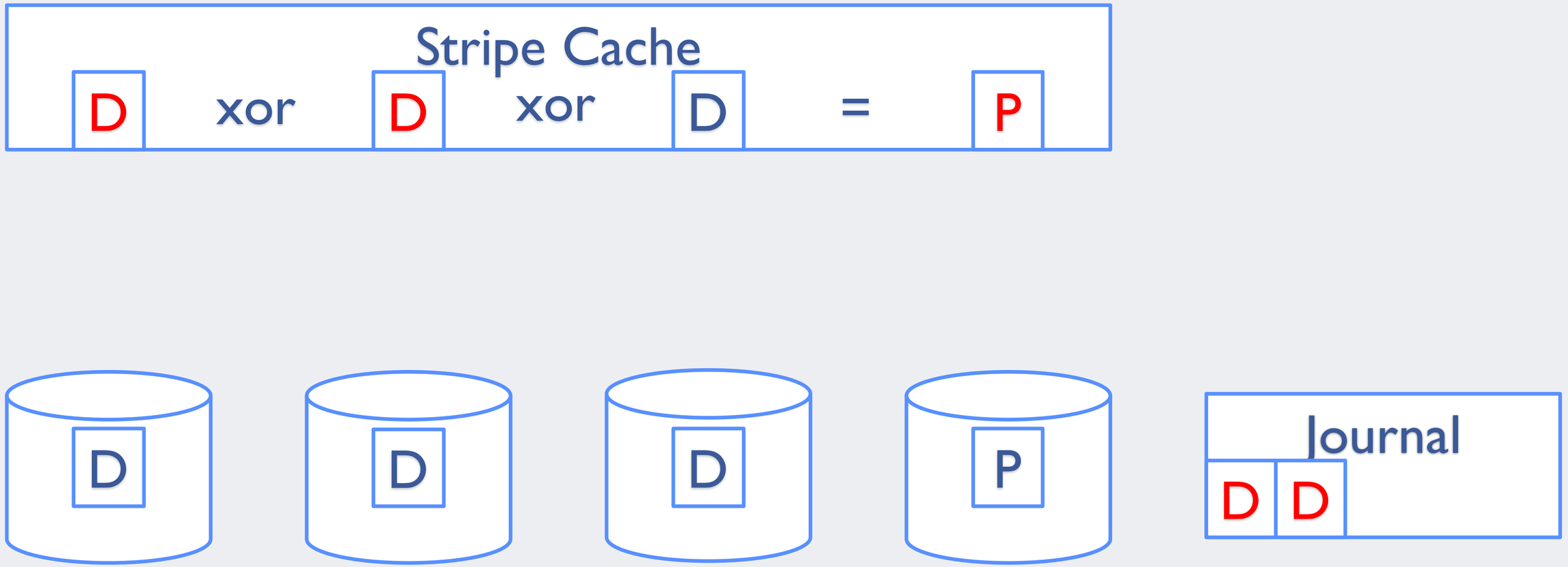
RAID-456 Write Cache: Reclaim Path



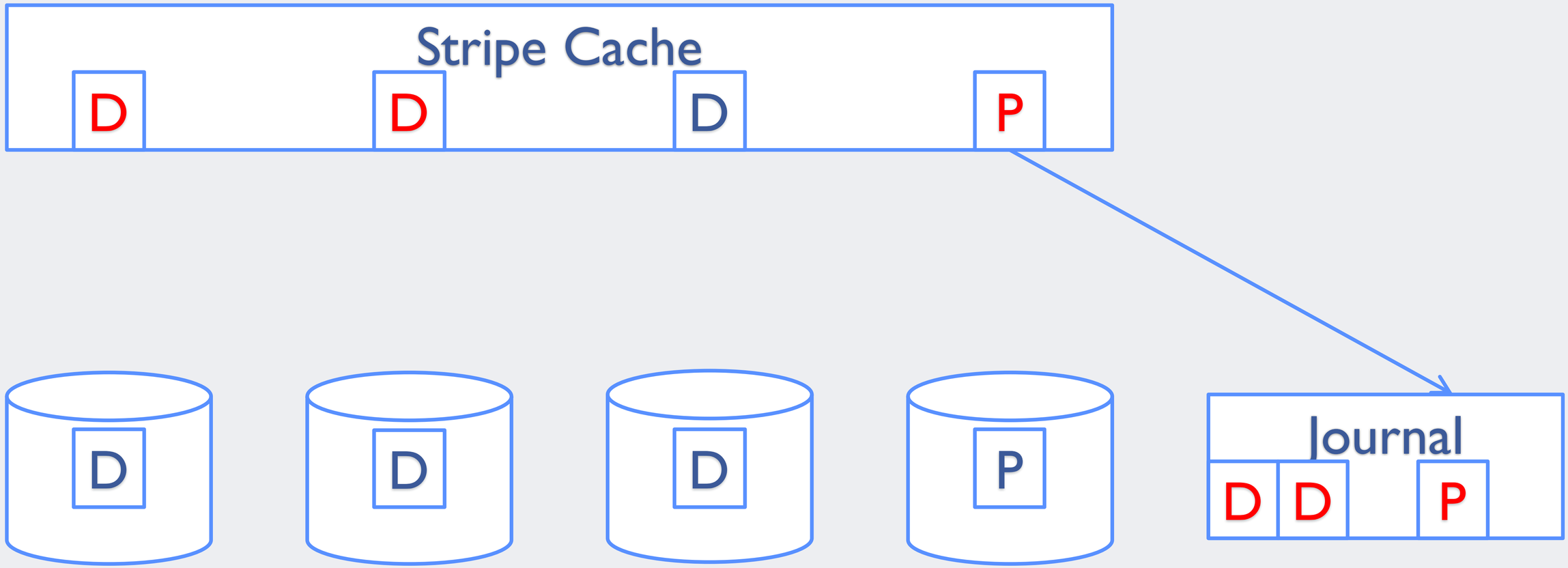
RAID-456 Write Cache: Reclaim Path



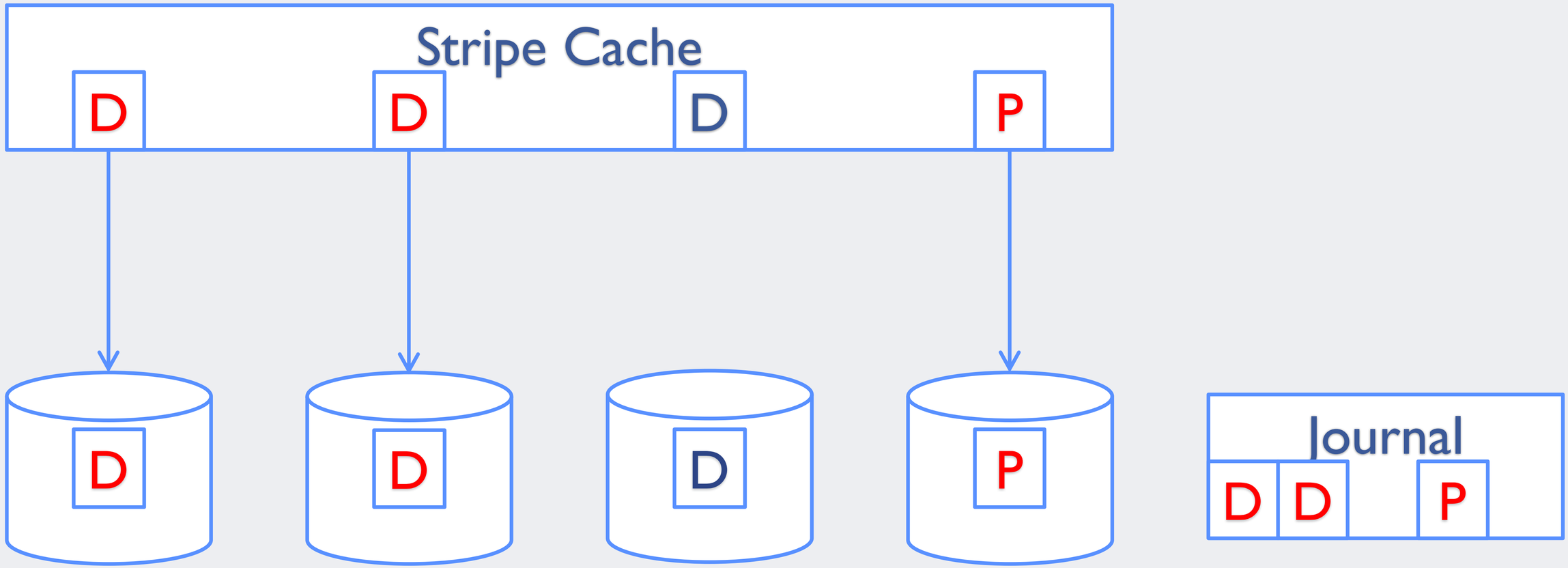
RAID-456 Write Cache: Reclaim Path



RAID-456 Write Cache: Reclaim Path



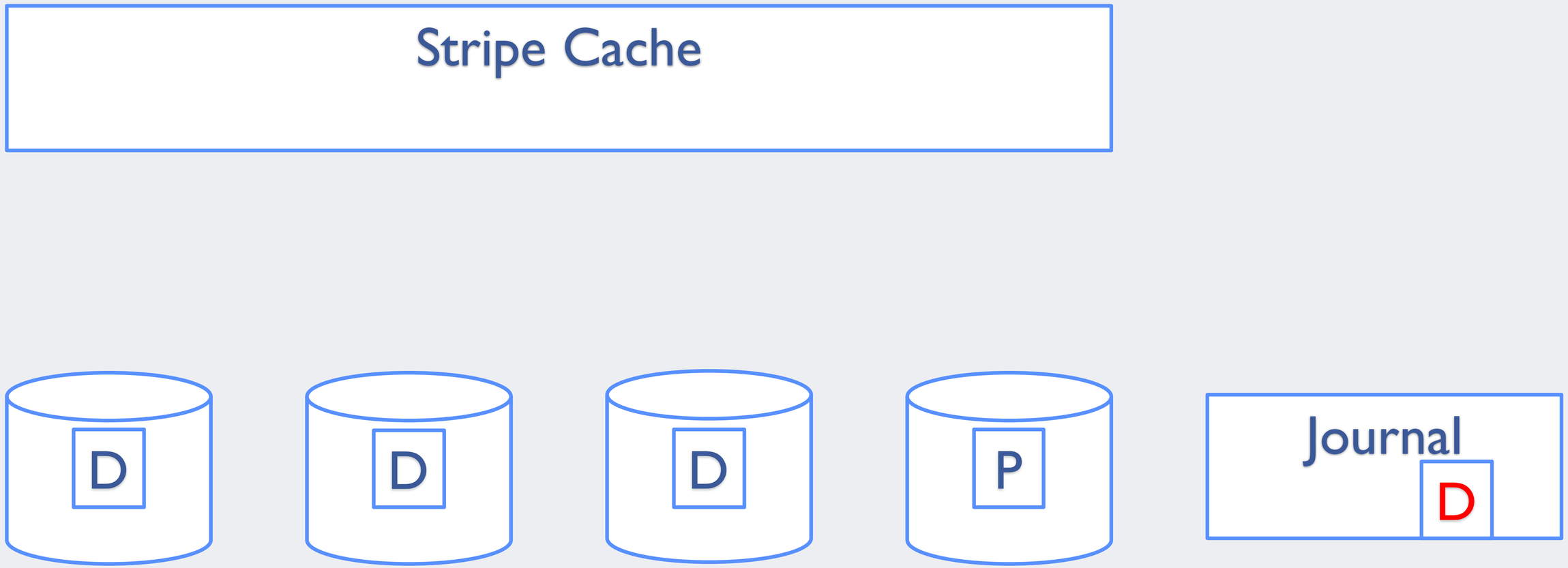
RAID-456 Write Cache: Reclaim Path



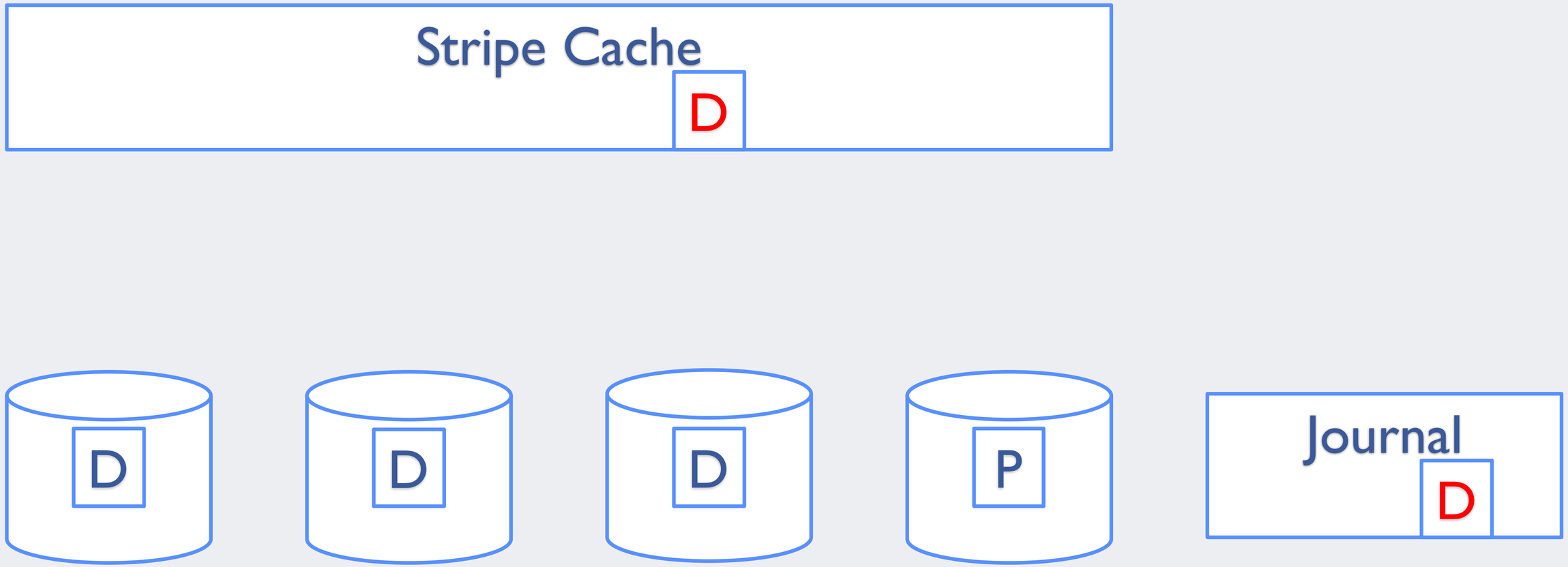
RAID-456 Write Cache: Recover Path

- Stripes with data and parity in journal
  - Replay writes of data and parity
- Stripes with data in journal
  - Repeat full recontruct write or R-M-W write

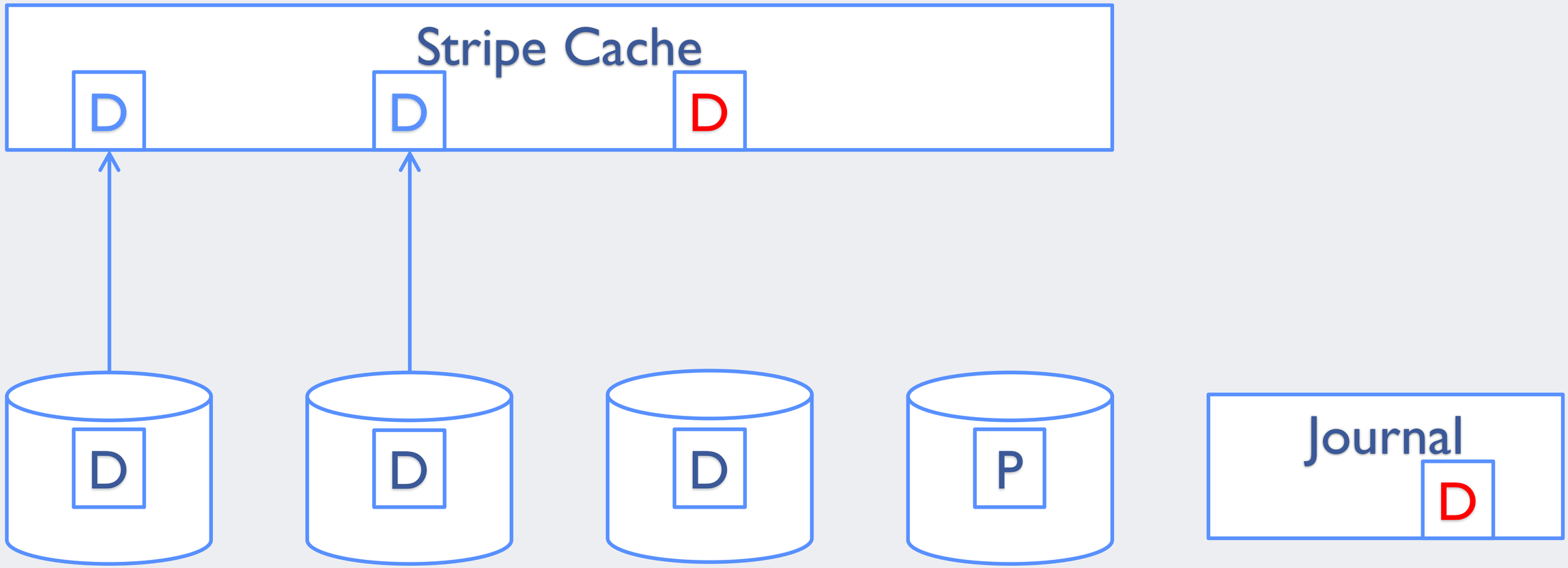
Recover Stripe Data: Reconstruct



Recover Stripe Data: Reconstruct

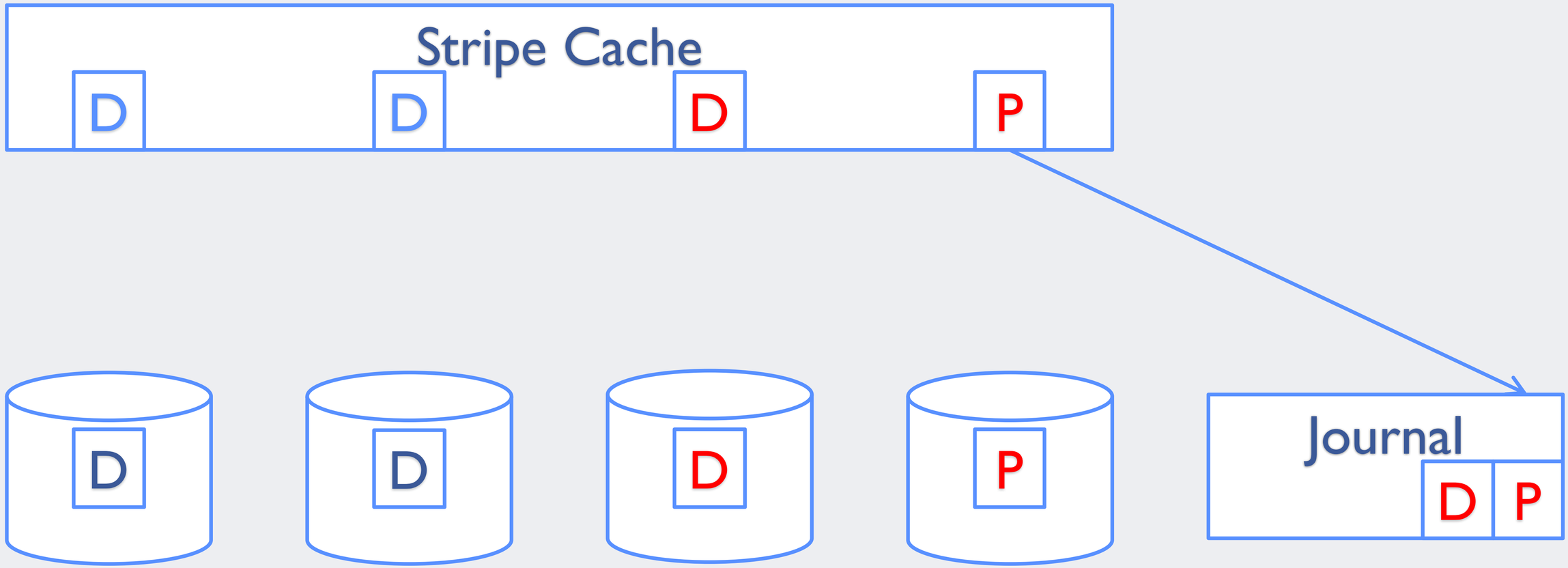


Recover Stripe Data: Reconstruct

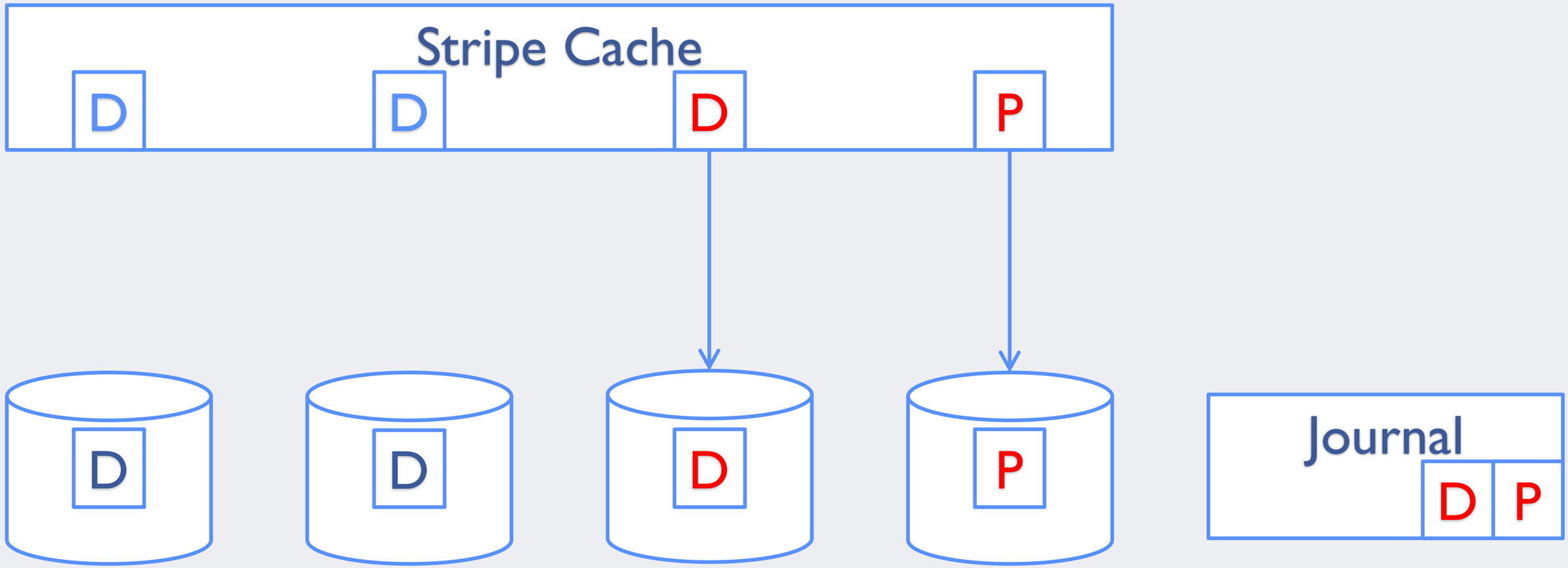


Recover Stripe Data: Reconstruct

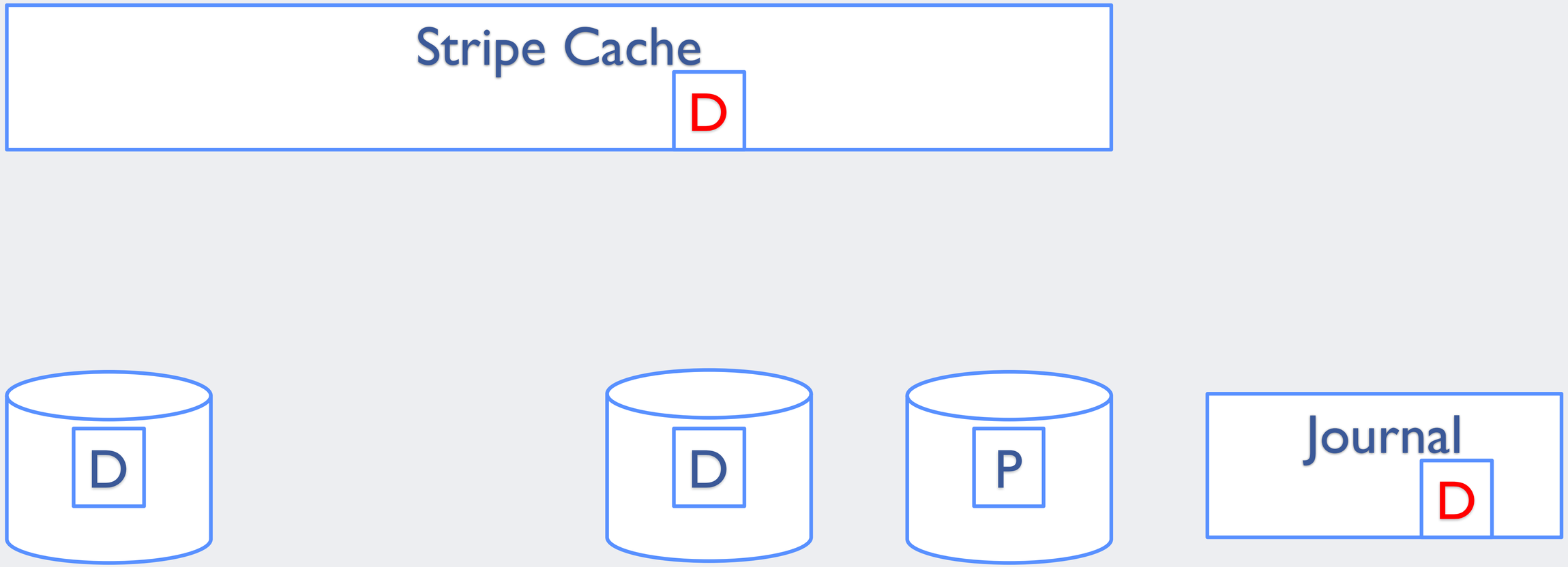
Recover Stripe Data: Reconstruct



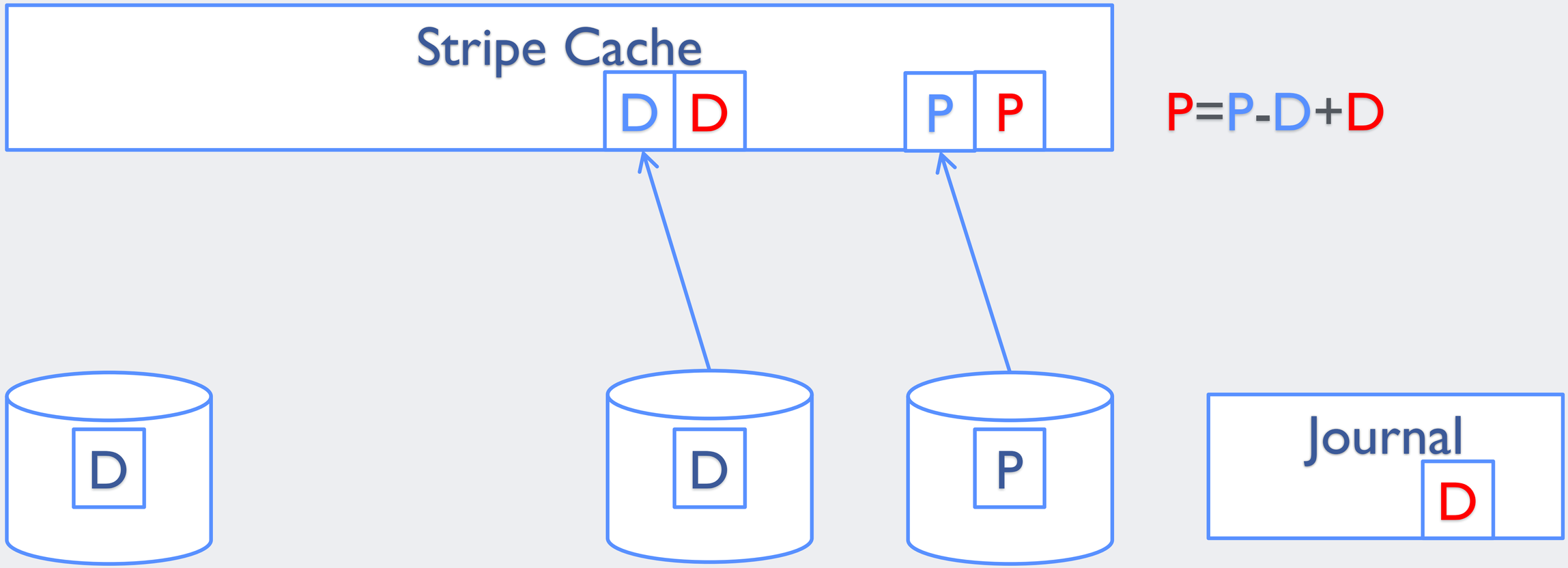
Recover Stripe Data: Reconstruct



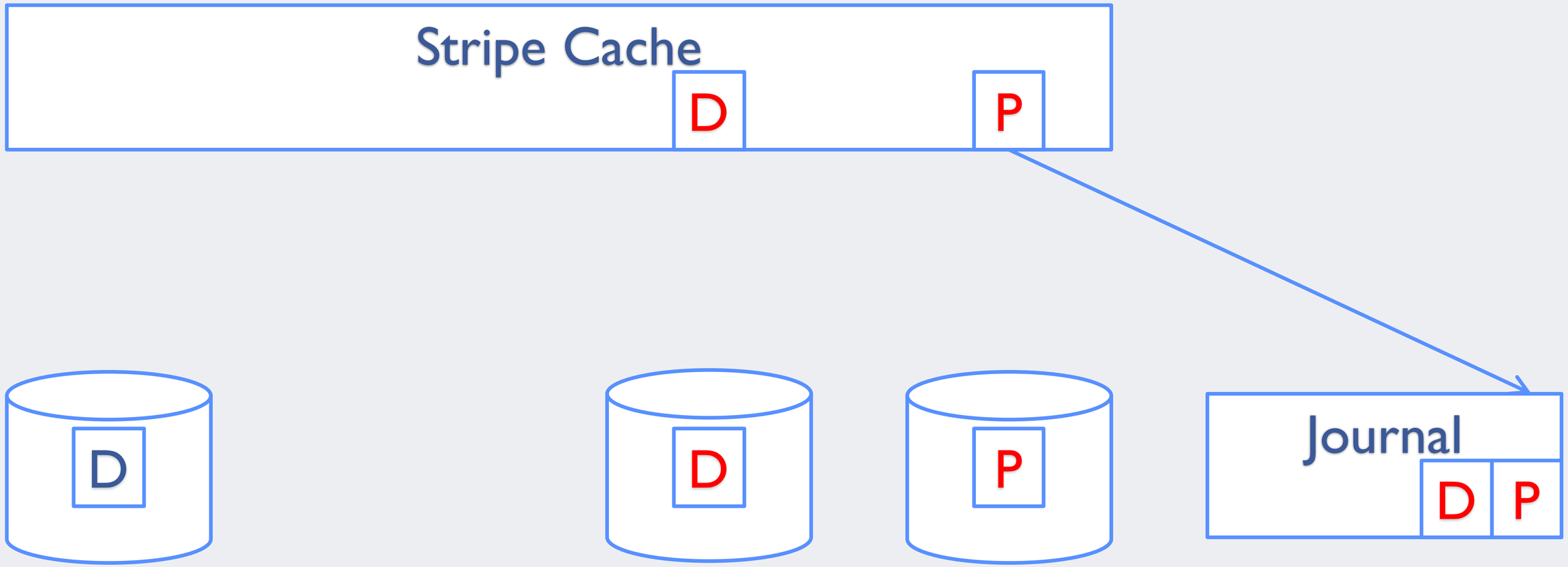
Recover Stripe Data: R-M-W



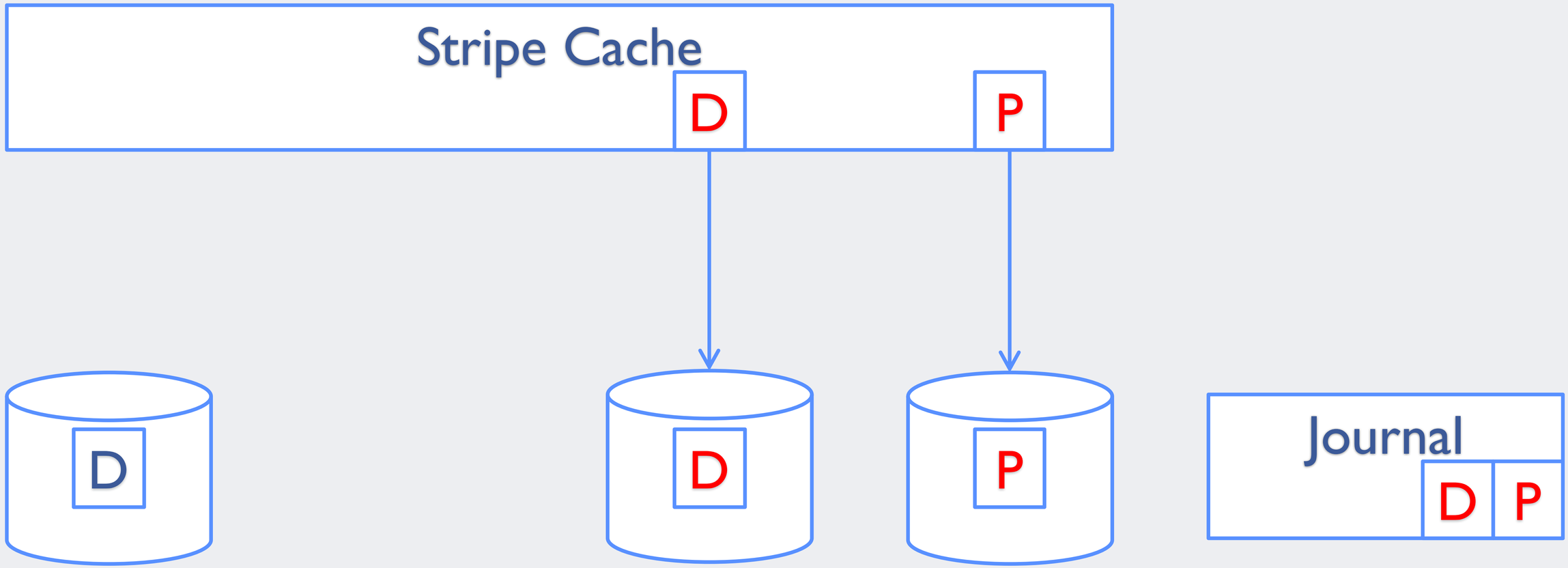
Recover Stripe Data: R-M-W



Recover Stripe Data: R-M-W



Recover Stripe Data: R-M-W



Current Status

- Write Journal
  - Kernel changes released with kernel 4.4
  - mdadm changes released with mdadm-3.4
- Write Cache
  - Kernel changes in progress
  - No change required for mdadm

Examples

```
# create array with write journal
mdadm --create -f /dev/md0 -c 64 --raid-devices=4 --
level=5 /dev/sd[b-e] --write-journal /dev/sdf

# check array with journal
cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]md0 : active raid5
sdf[4](J) sde[3] sdd[2] sdc[1] sdb[0]

# add journal to existing array
mdadm --manage /dev/md0 --add-journal /dev/sdf
```

facebook