CHINA LINUX KERNEL
中国Linux内核开发者大会

華中科技大學
网络空间安全学院
School of Cyber Science and Engineering, HUST

第19届中国Linux内核开发者大会

赞助单位

华中科技大学 | HUAWEI | 阿里云 | FUJITSU | 腾讯云
oppo | 火山引擎 | vivo | 龙芯中科 LOONGSON TECHNOLOGY | 蚂蚁集团 ANT GROUP

支持单位

清华大学 Tsinghua University | 迪捷软件 DIGIPROTO | OpenEuler | OpenAnolis 龙蜥社区 | OpenCloudOS

支持社区&媒体

CSDN | 云巅论剑 | sf 思否 | 阅码场 yomocode | InfoQ 极客传媒
51CTO | 开源江湖

2024年10月　湖北 • 武汉

華中科技大學

Device Mapper：减少 flush 和 verity 耗时的方案

王建政 vivo 存储系统工程师

杨　阳 vivo 存储系统工程师

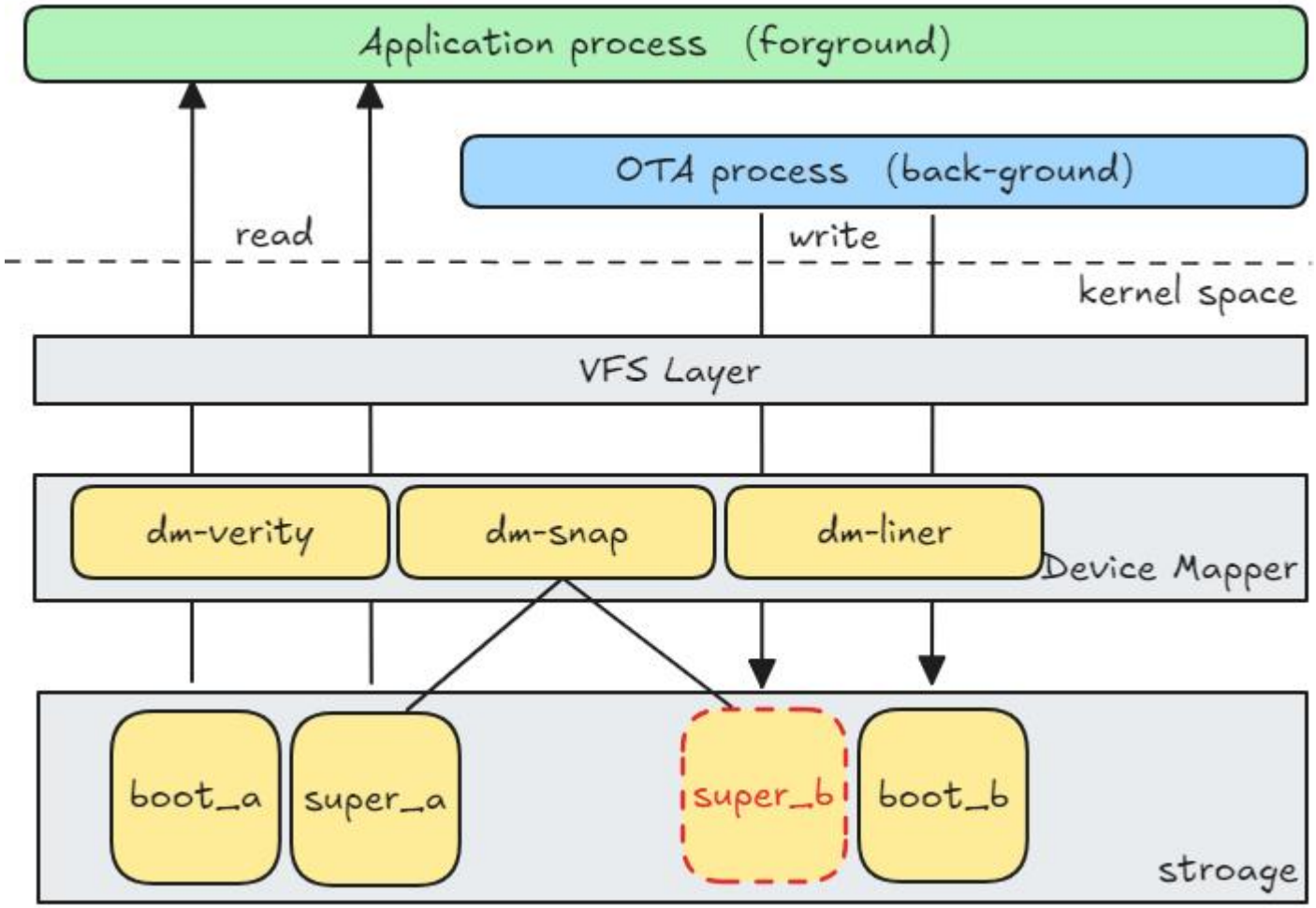
目录

Part 1

背景介绍

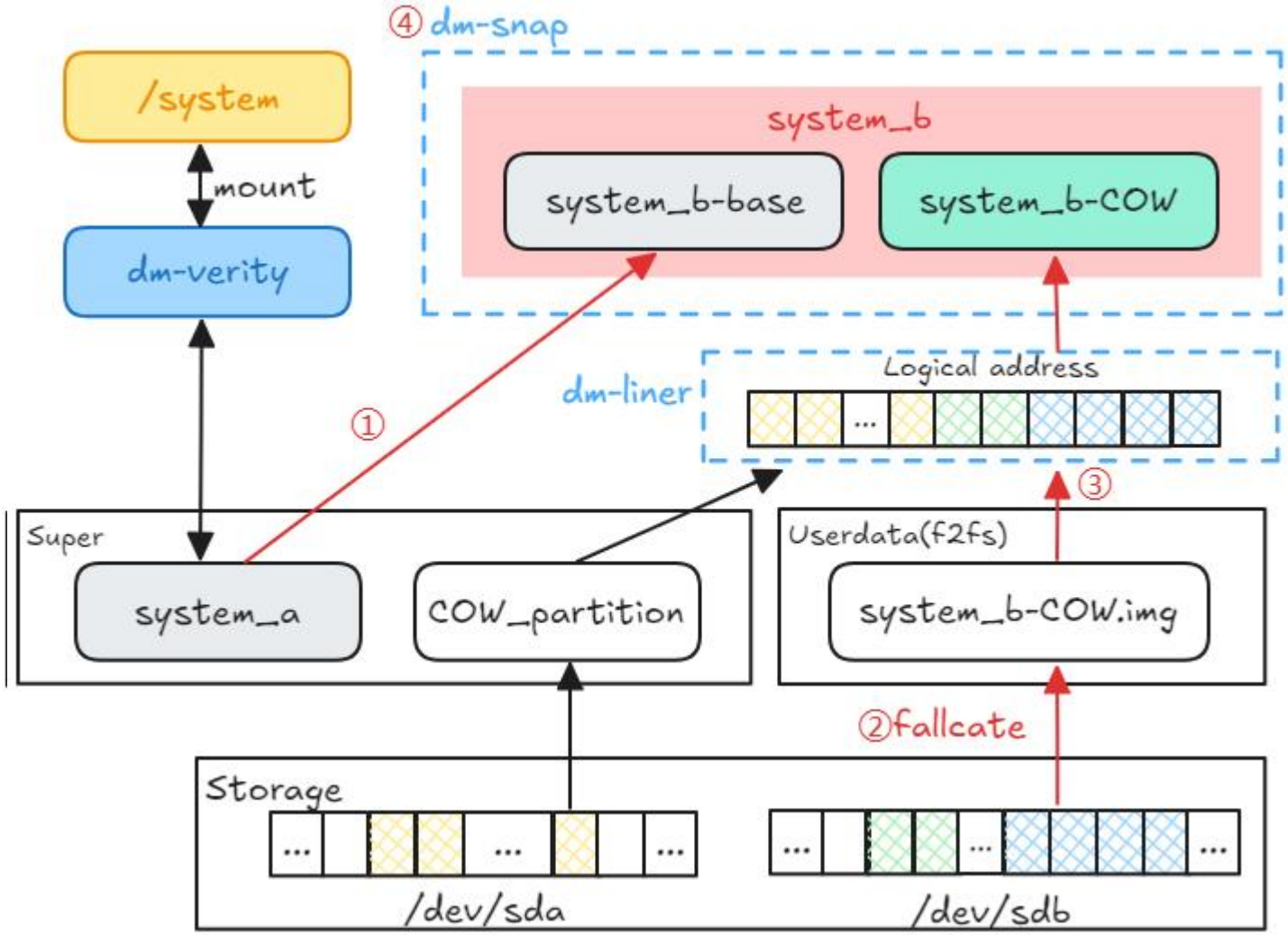
虚拟 AB 升级 —— 当前主流的 OTA 升级方式

- 自 Android 11 开始使用虚拟 AB 升级
- A/B 双分区满足同时使用和升级
- Device Mapper 为实现虚拟分区的构建和 IO 重定向提供了技术基础



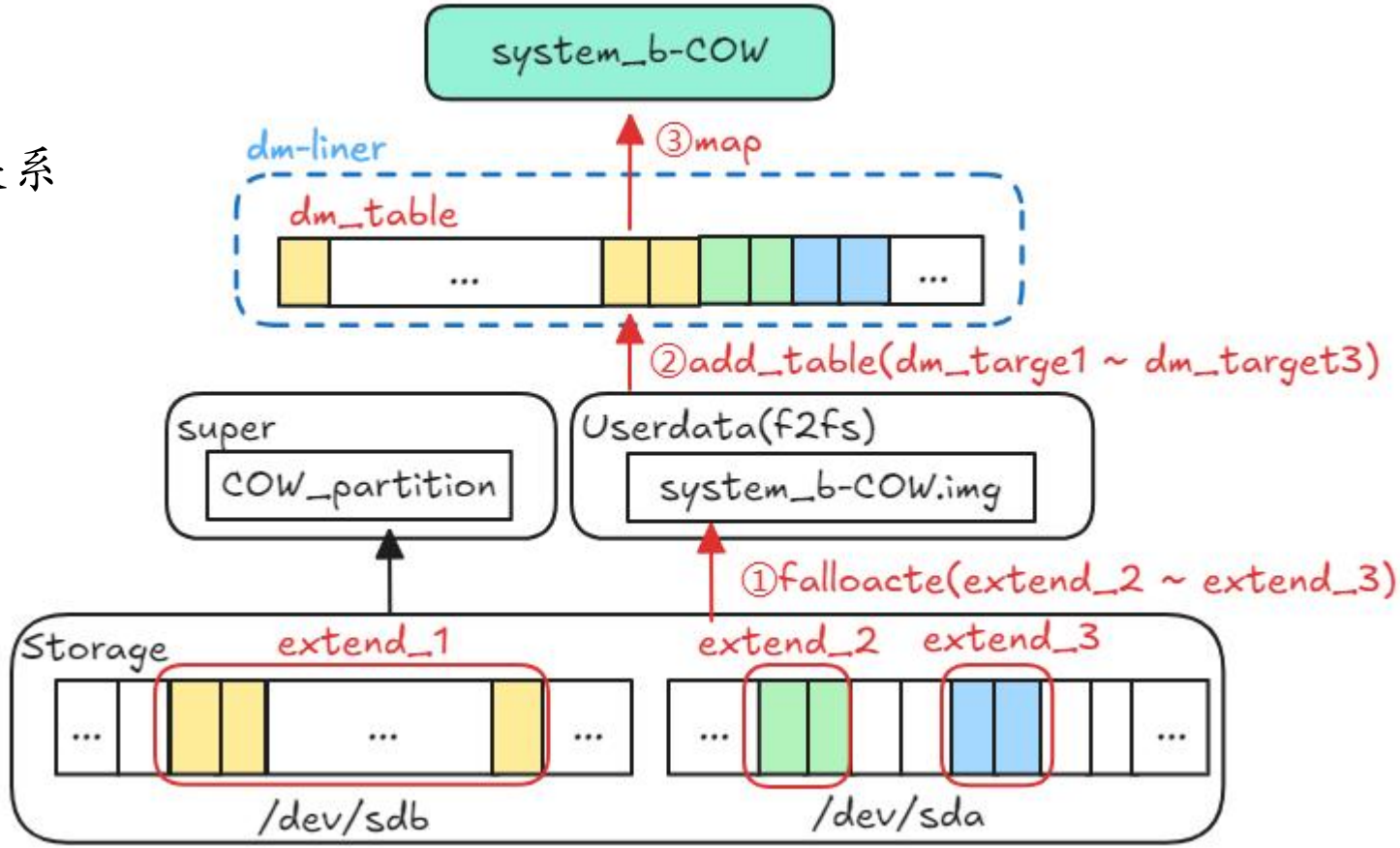
虚拟AB升级 —— 写入时通过 dm-snap 创建虚拟 B 分区

- 源设备（system_a）
- 基础设备（system_b-base）
- 申请存储空间（system_b-COW.img）
- 写时复制设备（system_b-COW）



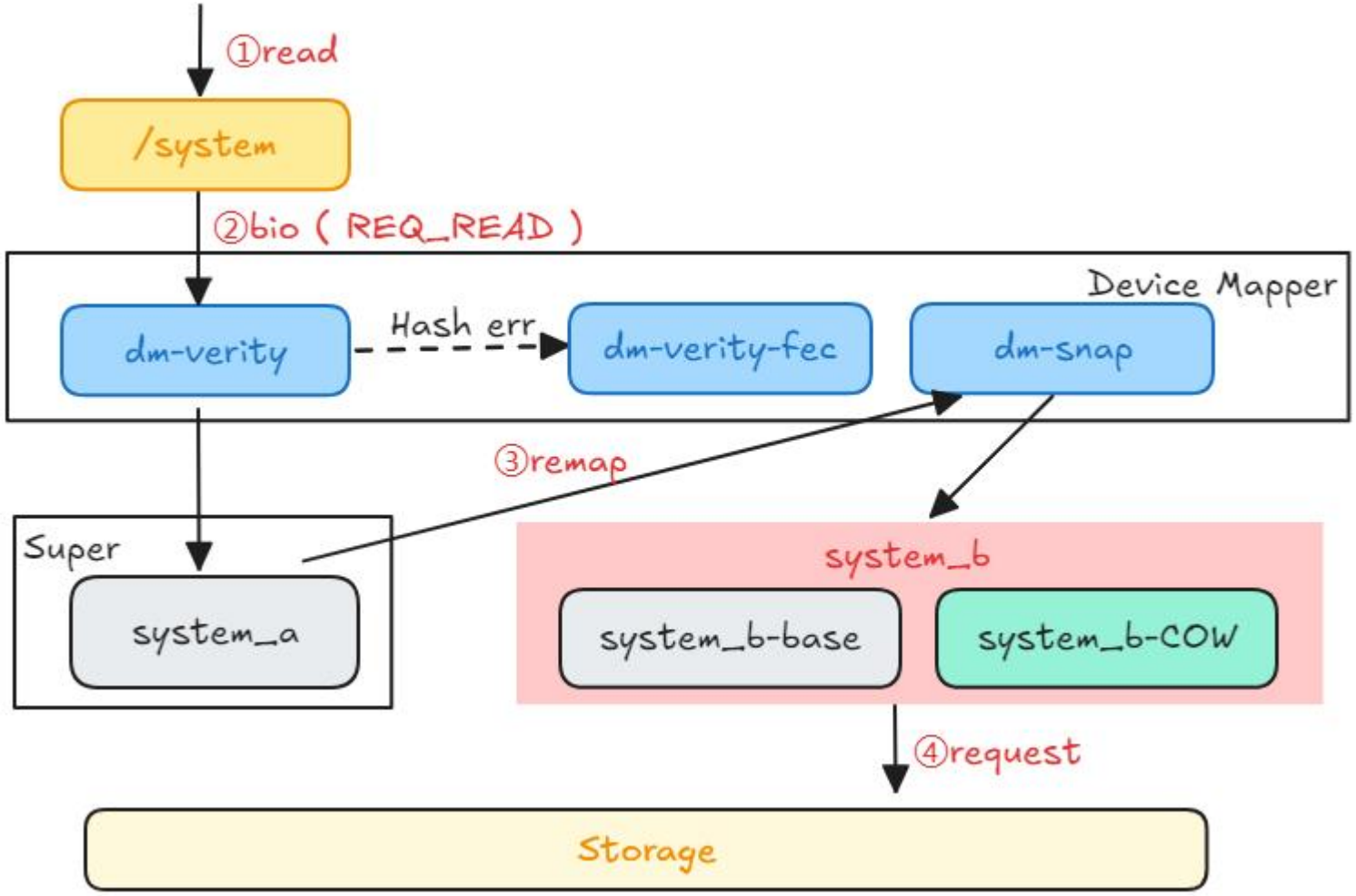
虚拟AB升级 —— 通过 dm-liner 映射 COW 设备

- 1个 extend 对应1个 dm_target
- Extend 记录文件和存储器地址映射关系
- 目标空间可能来自不同的存储器



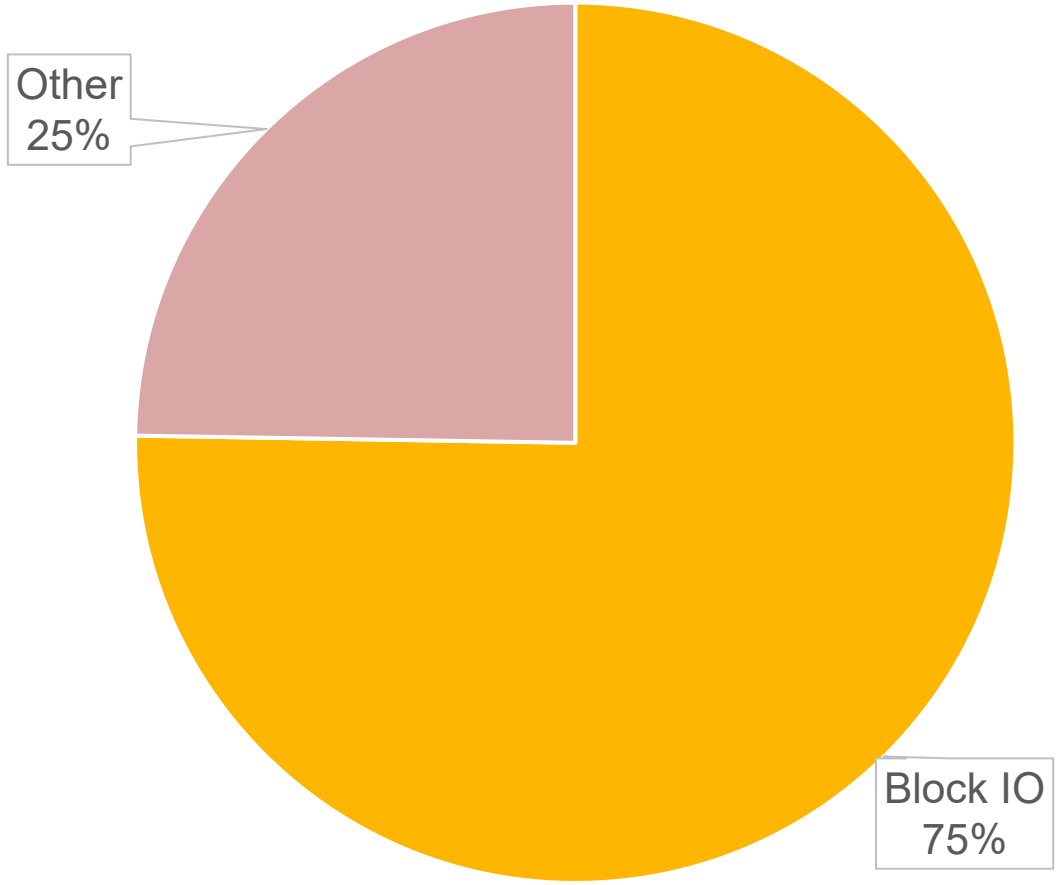
虚拟AB升级 —— 读取时通过 dm-verity 校验数据完整性

- dm-snap 模块重定向 IO 到 sytem_b
- dm-verity 模块校验数据块哈希值
- Hash error 会触发 FEC 纠错流程



虚拟 AB 升级存在耗时长的问题

- Flush 耗时较大
- Verity 耗时较大

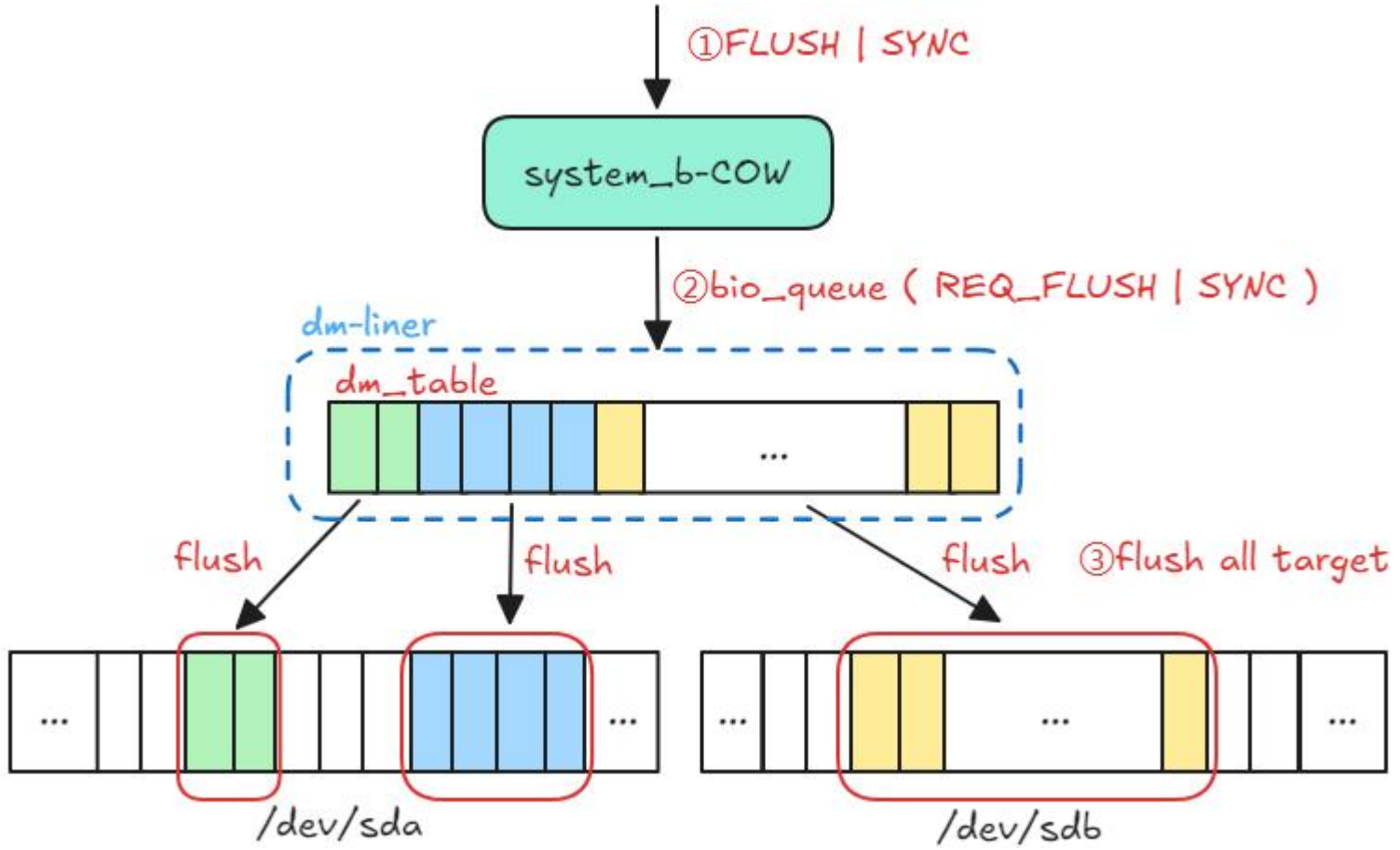


Part 2

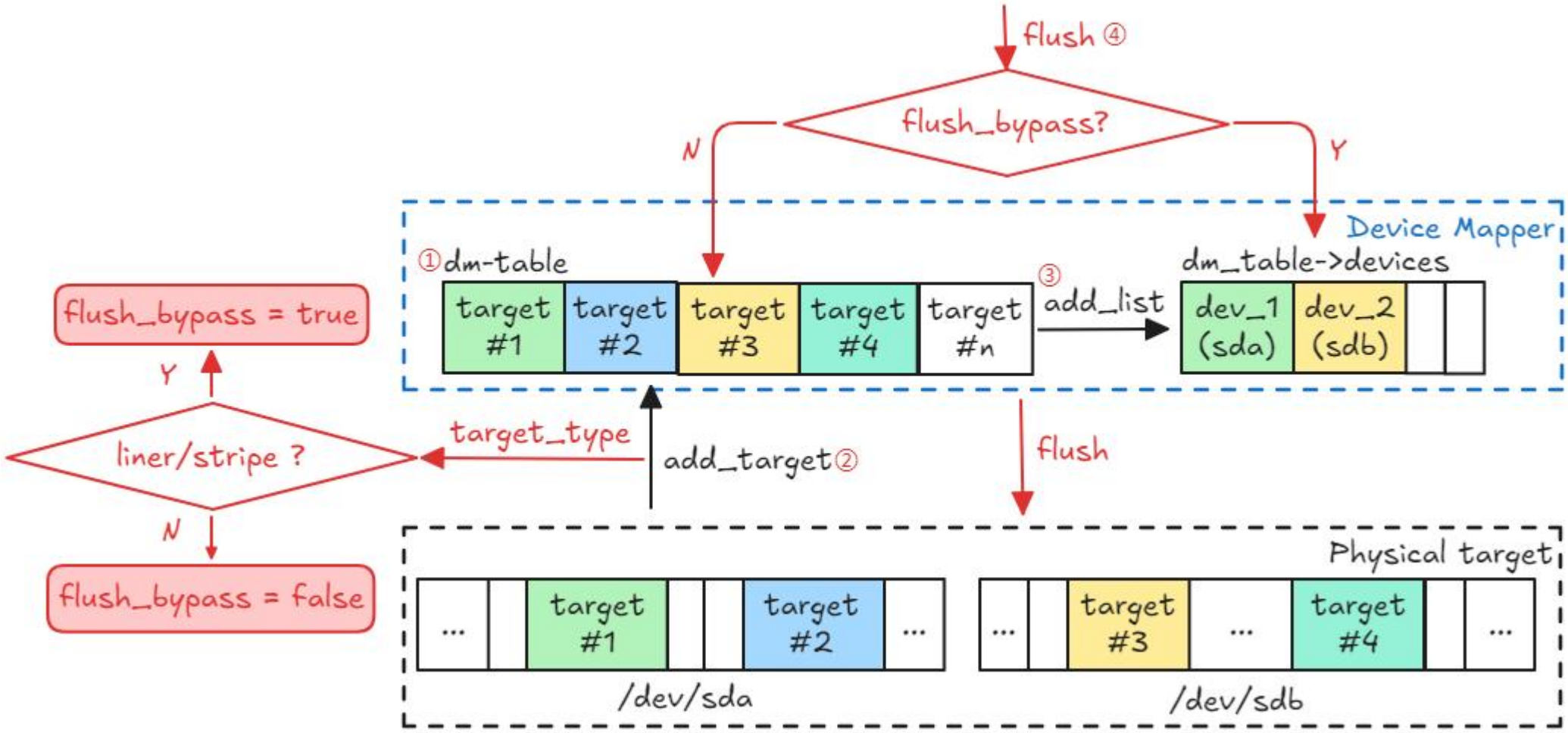
Flush 耗时

Dm-liner 写入时发生 flush 放大

- Flush 请求派发到每个 target
- 单存储器设备 flush 请求会重复
- 存储器碎片化会增加 num_target



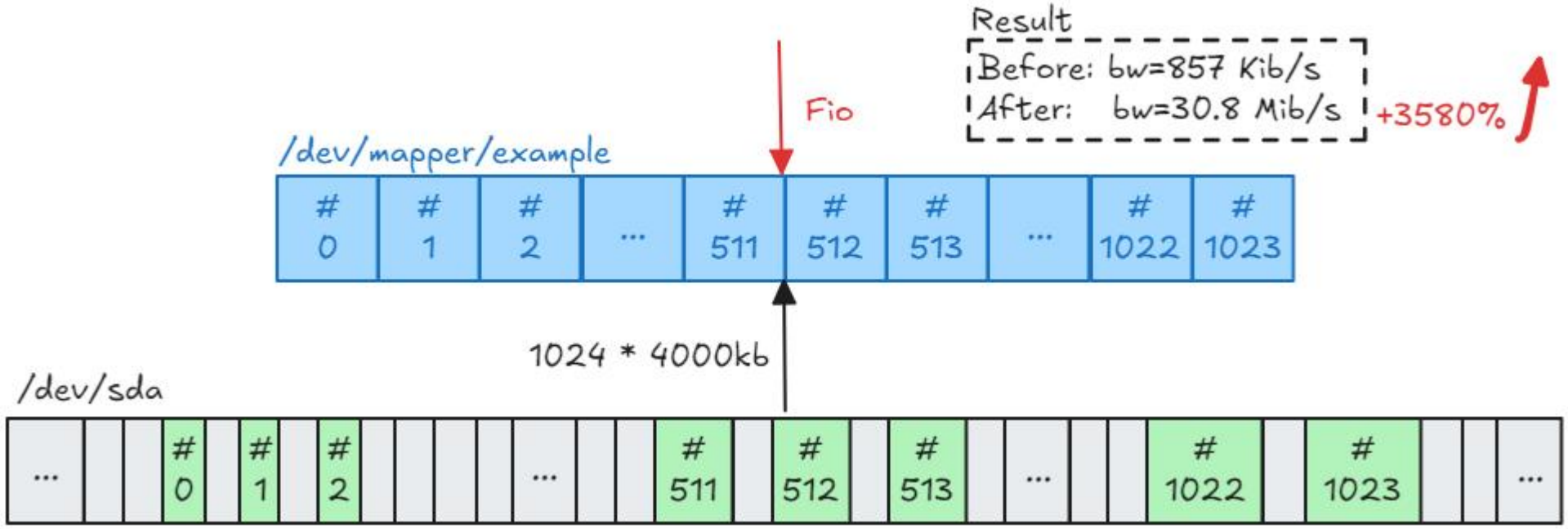
优化 flush 方案 —— 减少重复 flush，提升性能



单存储设备收益 —— 性能提升 3580%

本地单存储设备环境验证：

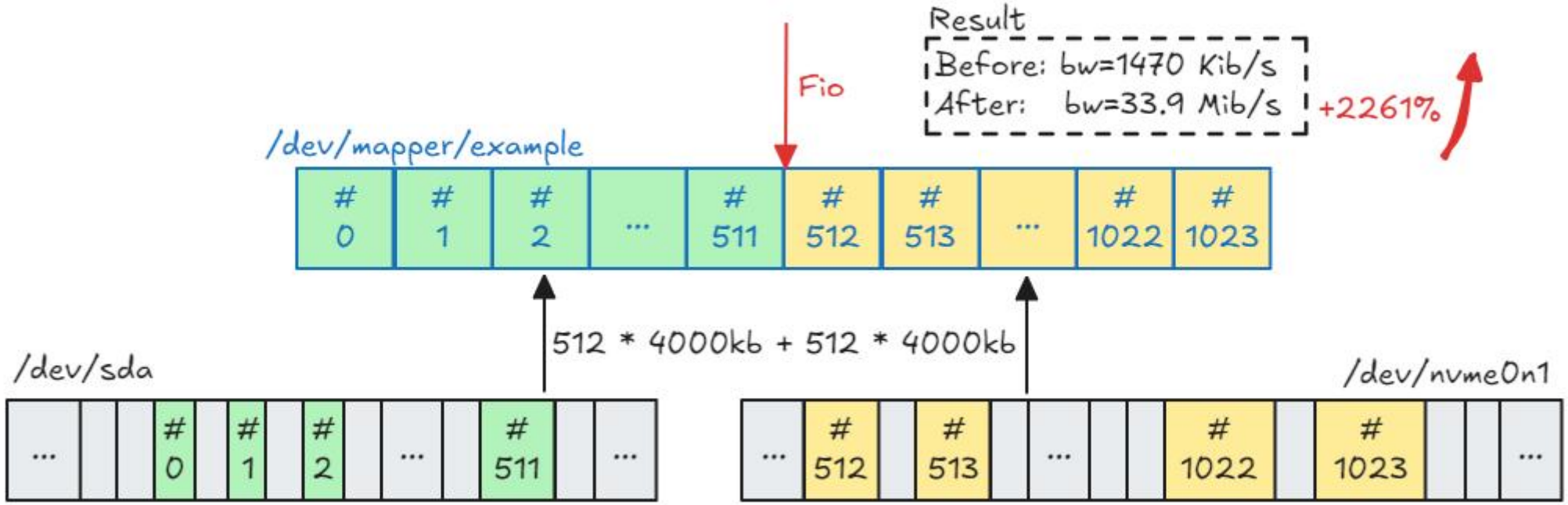
```
fio --group_reporting --name=benchmark --filename=/dev/mapper/example \
    --ioengine=sync --invalidate=1 --numjobs=16 --rw=randwrite \
    --blocksize=4k --size=2G --time_based --runtime=30 --fdatasync=1
```



多存储设备收益 —— 性能提升 2261%

本地多存储设备环境验证：

```
fio --group_reporting --name=benchmark --filename=/dev/mapper/example \
    --ioengine=sync --invalidate=1 --numjobs=16 --rw=randwrite \
    --blocksize=4k --size=2G --time_based --runtime=30 --fdatasync=1
```

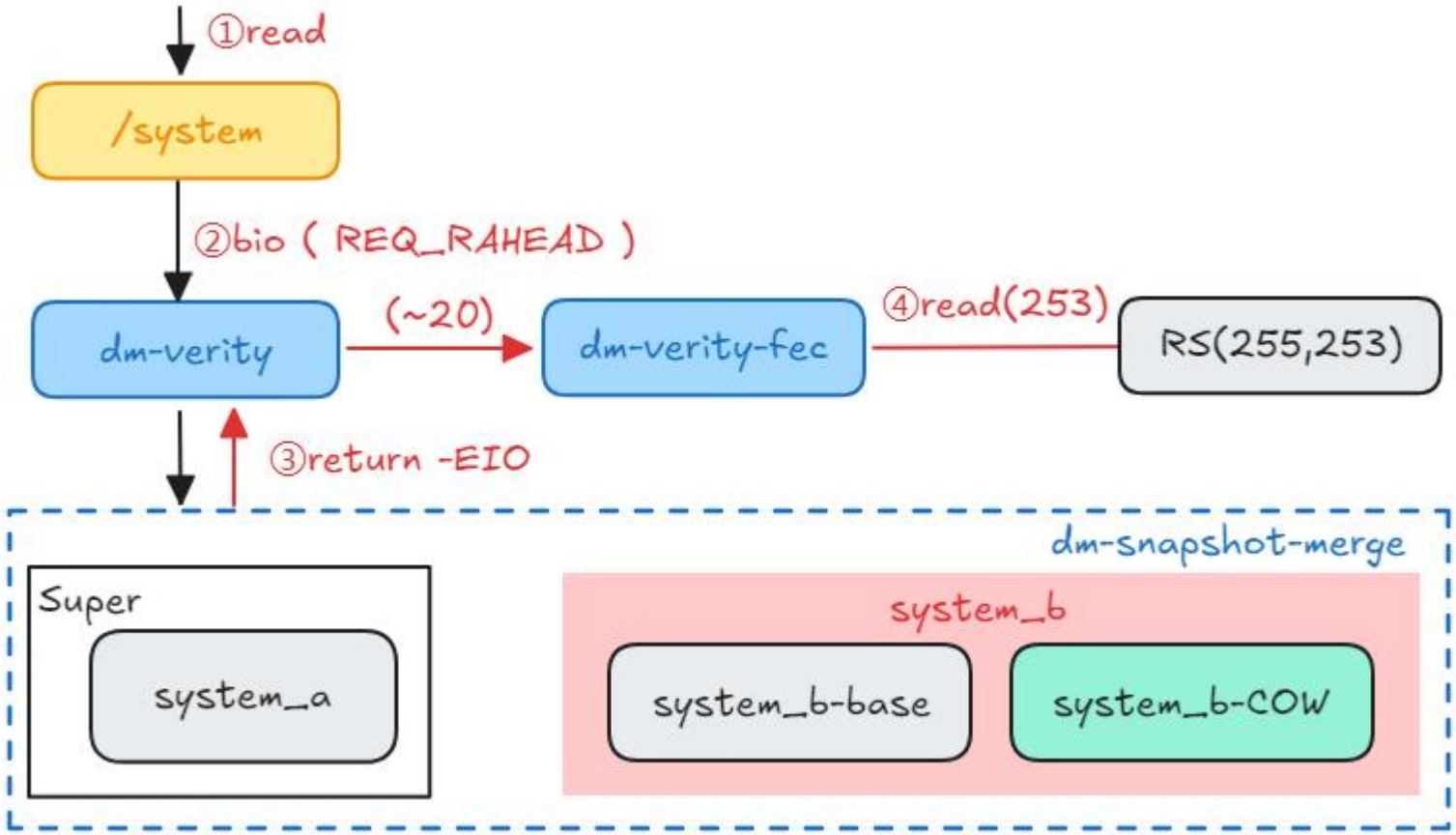


Part 3

Verity 耗时

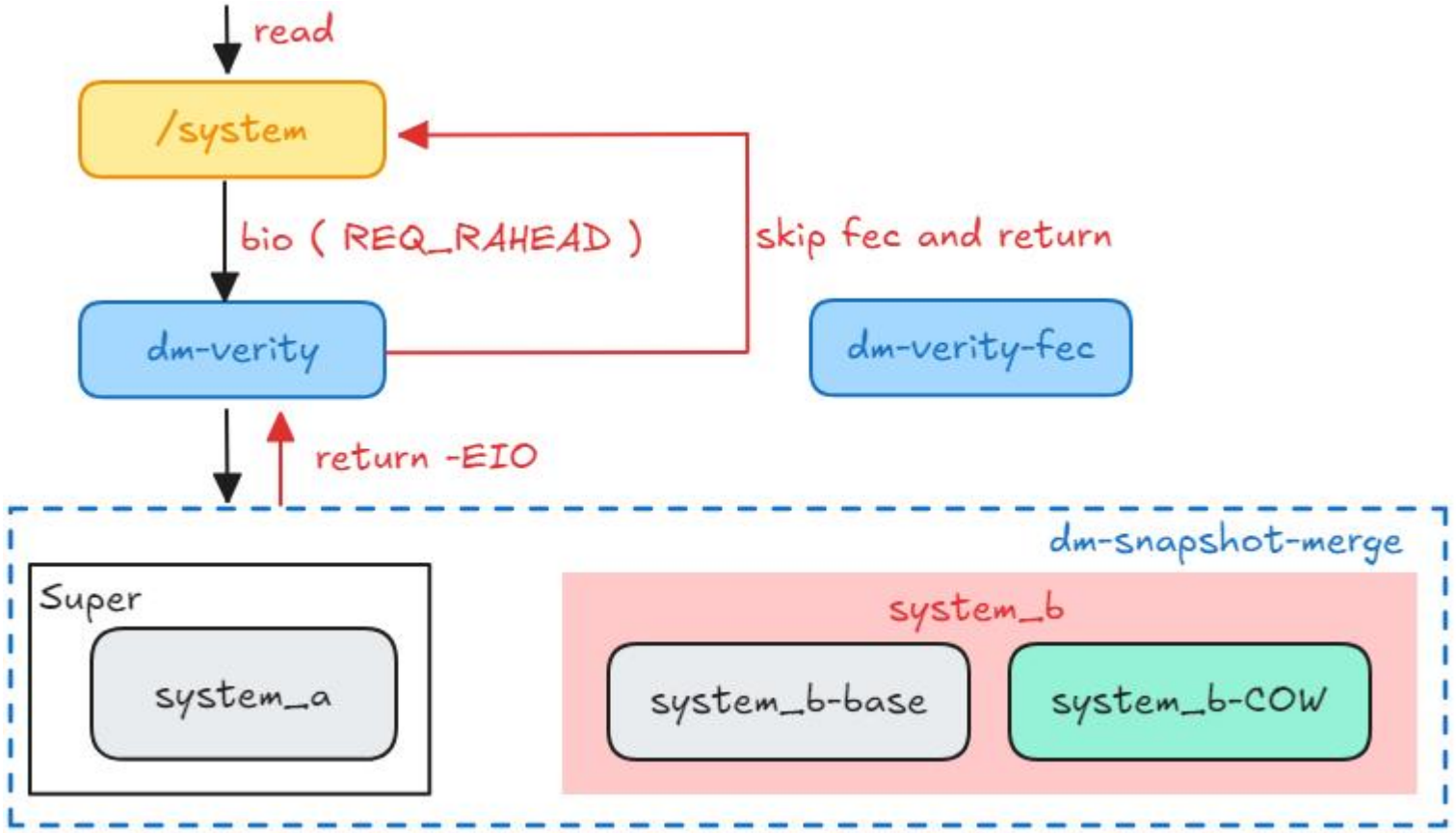
Dm-verity-FEC 纠错时发生 IO 放大

- dm-snap-merge 时设备 suspend
- 期间产生 ~300 次 readahead 请求
- readahead 请求中 ~20 个blocks
- 每个 block 需要一个 RS 矩阵



优化 dm-verity 方案 —— 避免 IO 放大

- 避免了 FEC 流程 5000 倍的 IO 放大



代码开源

Flush方案：[PATCH 0/5] dm: empty flush optimization （已合入社区主线）

[Linux Kernel Patch] https://lore.kernel.org/dm-devel/20240514090445.2847-1-yang.yang@vivo.com/

Verity方案：dm verity: don‘t perform FEC for failed readahead IO（已合入社区主线）

[Linux Kernel Patch] https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=0193e3966ceeeef69e235975918b287ab093082b

```
From: Yang Yang <yang.yang@vivo.com>
To: Alasdair Kergon <agk@redhat.com>,
        Mike Snitzer <snitzer@kernel.org>,
        Mikulas Patocka <mpatocka@redhat.com>,
        dm-devel@lists.linux.dev, linux-kernel@vger.kernel.org
Cc: Yang Yang <yang.yang@vivo.com>
Subject: [PATCH 0/5] dm: empty flush optimization
Date: Tue, 14 May 2024 17:04:39 +0800    [thread overview]
Message-ID: <20240514090445.2847-1-yang.yang@vivo.com> (raw)

Yang Yang (5):
  dm: introduce flush_pass_around flag
  dm: add __send_empty_flush_bios() helper
  dm: support retrieving struct dm_target from struct dm_dev
  dm: Avoid sending redundant empty flush bios to the same block device
  dm linear: enable flush optimization function

 drivers/md/dm-core.h          |  3 +++
 drivers/md/dm-ioctl.c         |  4 ++++
 drivers/md/dm-linear.c        |  1 +
 drivers/md/dm-table.c         | 39 ++++++++++++++++++++++++++++++++++++++
 drivers/md/dm.c               | 37 ++++++++++++++++++++++++++++++++++++-------
 include/linux/device-mapper.h |  8 +++++++
 6 files changed, 83 insertions(+), 9 deletions(-)

--
2.34.1
```

```
author      Wu Bo <bo.wu@vivo.com>          2023-11-21 20:51:50 -0700
committer   Mike Snitzer <snitzer@kernel.org>  2023-11-29 12:55:31 -0500
commit      0193e3966ceeeef69e235975918b287ab093082b (patch)
tree        97d0cc70a0552d2957ec45133e7122c2a3c45495
parent      7be05bdfb4efc1396f7692562c7161e2b9f595f1 (diff)
download    linux-0193e3966ceeeef69e235975918b287ab093082b.tar.gz

dm verity: don't perform FEC for failed readahead IO

Diffstat
-rw-r--r-- drivers/md/dm-verity-target.c 4

1 files changed, 3 insertions, 1 deletions

diff --git a/drivers/md/dm-verity-target.c b/drivers/md/dm-verity-target.c
index beec14b6b0442a..14e58ae705218f 100644
--- a/drivers/md/dm-verity-target.c
+++ b/drivers/md/dm-verity-target.c
@@ -667,7 +667,9 @@ static void verity_end_io(struct bio *bio)
        struct dm_verity_io *io = bio->bi_private;

        if (bio->bi_status &&
-           (!verity_fec_is_enabled(io->v) || verity_is_system_shutting_down
+           (!verity_fec_is_enabled(io->v) ||
+            verity_is_system_shutting_down() ||
+            (bio->bi_opf & REQ_RAHEAD))) {
                verity_finish_io(io, bio->bi_status);
                return;
        }
```

Thank You~