CHINA LINUX KERNEL
中国Linux内核开发者大会

vivo

Fallocate Write Zeroes
垂直零化文件预分配

华为 欧拉突击队
张翼
2025.11

CHINA LINUX KERNEL
中国Linux内核开发者大会

vivo

目录

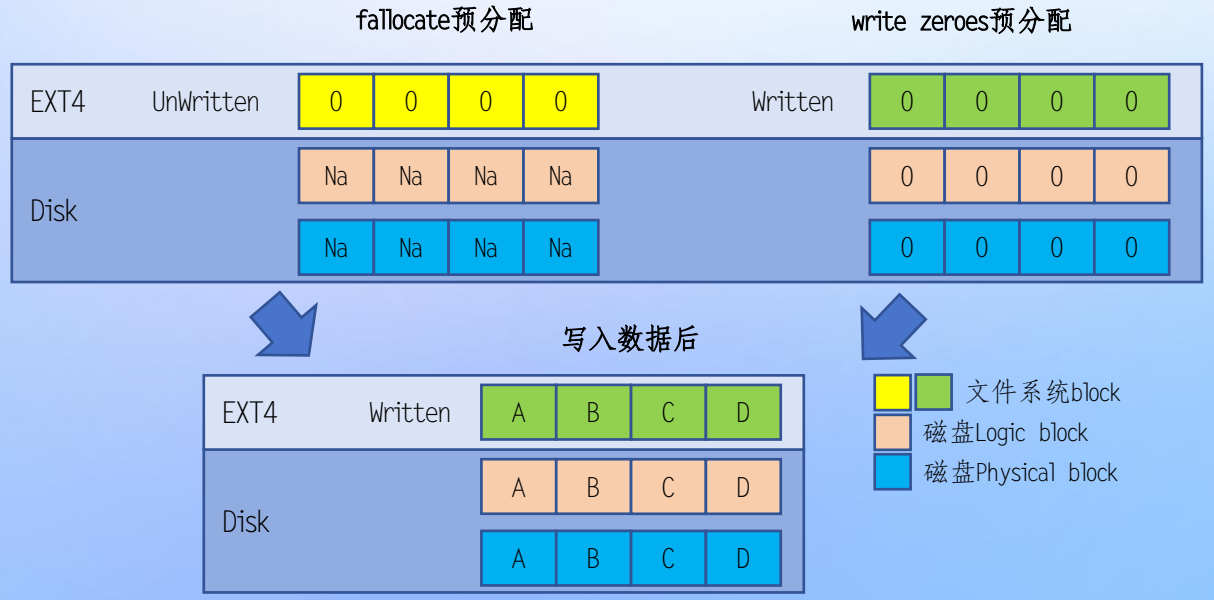
CONTENTS

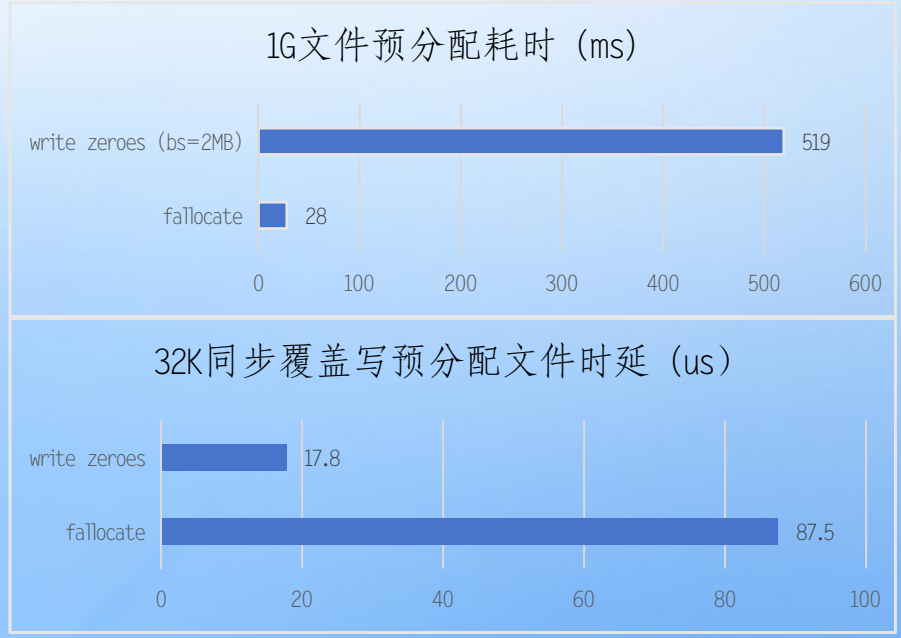
背景

- 在MySQL、Gauss Store等数据库应用场景中，一般通过**预分配文件**保证文件连续性，提升读写性能，预分配文件需先“**清零**”避免出现stale data
- 预分配文件后期存在大量**同步写（SYNC）**场景，如LOG日志事务、双写缓冲区持久化等。同步写可能会出现因大量文件“**元数据”变更持久化**而带来的额外I/O开销
- 关注如何提高**文件预分配**和**后期写入性能**，以及如何降低**文件预分配文件对其它业务I/O的干扰率**，进而提升数据库等应用场景综合性能

现有技术和问题

- fallocate / write zeroes。
  - fallocate（ FALLOC_FL_ALLOCATE_RANGE / FALLOC_FL_ZERO_RANGE ）：预分配仅文件系统元数据创建，分配速度快；后期写入 “元数据” 持久化严重I/O放大、影响I/O写入性能、磁盘干扰率和寿命
  - write zeroes：预分配需写入实际0数据，分配速度慢，占用大量磁盘I/O带宽，干扰率大；后期写放大小，I/O性能好
- **技术诉求：预分配速度快，分配干扰小，后期覆盖写I/O性能好**





垂直零化文件预分配——原理

- 软硬结合：依赖SSD硬件Unmap Write Zeroes能力，无需写入“零数据”即可完成“清零”操作
- NVMe SSDs通过Deallocate State Block和DEAC bit Write Zeroes Command实现
- SCSI SSDs通过Unmap bit Write Same 16/10 Command实现

Deallocated or Unwritten Logical Blocks

A logical block that has never been written to, or which has been deallocated using the Dataset Management command, the Write Zeroes command or the Sanitize command is called a deallocated or unwritten logical block.

Write Zeroes Command

The Write Zeroes command is used to clear a range of logical blocks or all of the logical blocks in an entire namespace to zero.

Deallocate (DEAC): If this bit is set to ‘1’, then the host is requesting that the controller deallocate the specified logical blocks. If this bit is cleared to ‘0’, then the host is not requesting that the controller deallocate the specified logical blocks.

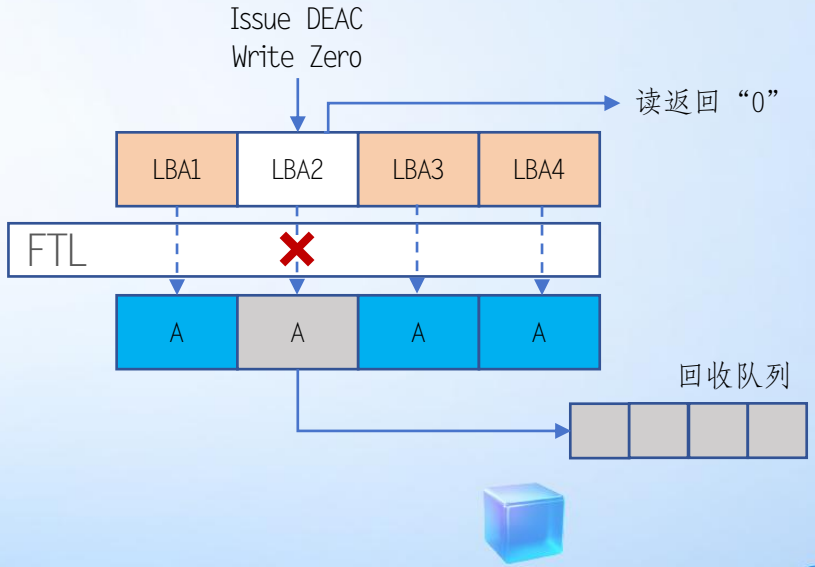
垂直零化文件预分配——原理

- NVMe SSD设备支持情况确认

Write Zeroes Deallocation Support (WZDS): If this bit is set to '1', then the controller supports the Deallocate bit in the Write Zeroes command for this namespace. If this bit is cleared to '0', then the controller does not support the Deallocate bit in the Write Zeroes command for this namespace. This bit shall be set to the same value for all namespaces in the NVM subsystem.

Deallocation Read Behavior (DRB): This field indicates the deallocated logical block read behavior.

Value	Definition
000b	The read behavior is not reported
001b	A deallocated logical block returns all bytes cleared to 0h
010b	A deallocated logical block returns all bytes set to FFh
011b to 111b	Reserved



- 使用nvme-cil工具确认

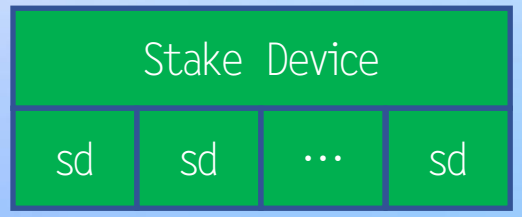
```
# nvme id-ctrl -H /dev/nvme0 | grep -i "zeroes"
  [3:3] : 0x1   Write Zeroes Supported
# nvme id-ns -H /dev/nvme0n1 | grep -i "deallocate"
  [2:2] : 0x1   Deallocated or Unwritten Logical Block error Supported
  [4:4] : 0x1   Guard Field of Deallocated Logical Blocks is set to CRC of The Value
Read
  [3:3] : 0x1   Deallocate Bit in the Write Zeroes Command is Supported
  [2:0] : 0x1   Bytes Read From a Deallocated Logical Block and its Metadata are 0x00
```

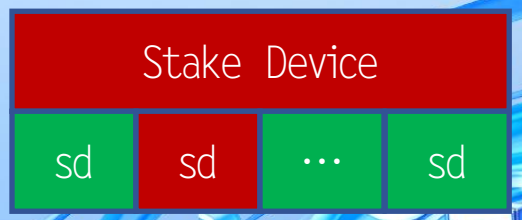
垂直零化文件预分配——实现

- 通用块层queue_limits支持max_[hwluser]wzeroes_umap_sectors参数

```
# cat /sys/block/sdc/queue/write_zeroes_unmap_max_hw_bytes
33553920
# cat /sys/block/sdc/queue/write_zeroes_unmap_max_bytes
33553920
```

- NVMe、SCSI驱动检测硬件并设置max_hw_wzeroes_umap_sectors
  - SCSI：检测lbprz && lbpws/lbpws10支持
  - NVMe：检测DEAC bit、DRB bits和WZDS bits支持
- Stacked devices
  - DM
  - MD-Linear
  - MD-RAID0
  - RAID 1/10/5 (TODO)

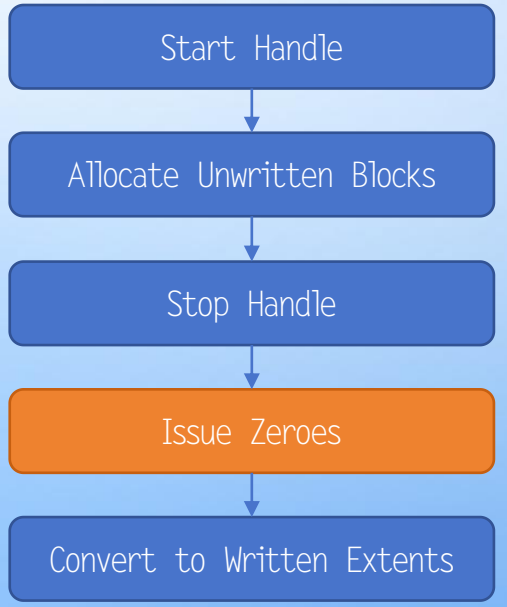




垂直零化文件预分配——实现

- VFS fallocate系统调用新增FALLOC_FL_WRITE_ZEROES参数支持Unmap Write Zeroes
- 文件系统和块设备支持
  - 检查底层硬件是否支持Unmap Write Zeroes Command，否则返回-EOPNOTSUPP；
  - 块设备层优先使用UNMAP类型下发Write Zeroes Command，若失败则回退写零操作；
  - 原地写文件系统/文件支持
    - EXT4
    - XFS(TODO)
    - …
- 外围工具支持
  - linux-util: fallocate -w -l $size foo
  - xfsprogs: xfs_io -fc “fwzero $offset $size” foo

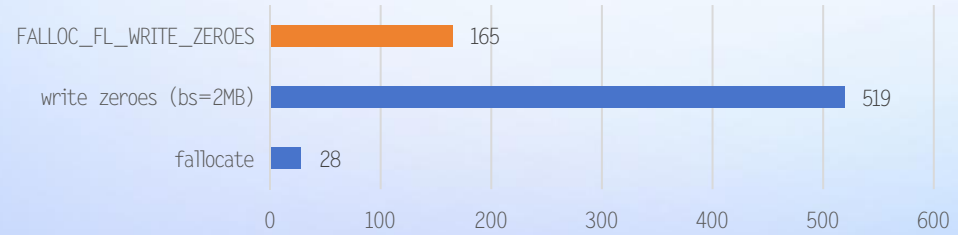
Linux Kernel Commit Link:



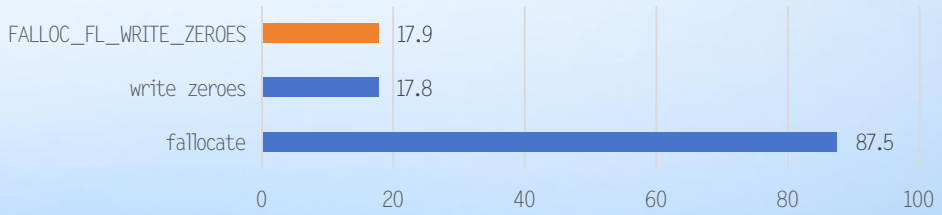
技术效果

- **创建**：文件预分配时间相较write zeroes大幅缩短，高于fallocate，不占用磁盘I/O带宽
- **覆盖写性能**：预分配文件后期覆盖写时延约等于write zeroes，相较于fallocate大幅缩短
- **干扰**：带宽干扰约为0，可跑满磁盘带宽，时延干扰~12%

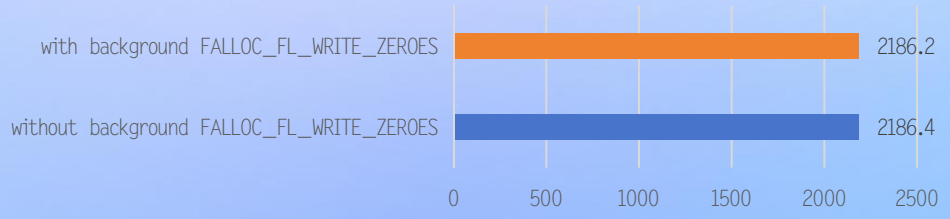
1G文件预分配耗时(ms)



32K同步覆盖写预分配文件时延（us）

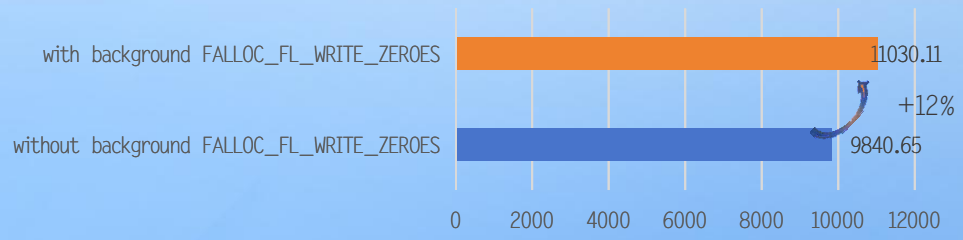


FALLOC_FL_WRITE_ZEROES 磁盘带宽影响(MiB/s)



FIO: write iodepth=10 bs=2M libaio numjobs=10

FALLOC_FL_WRITE_ZEROES 磁盘时延干扰(ns)

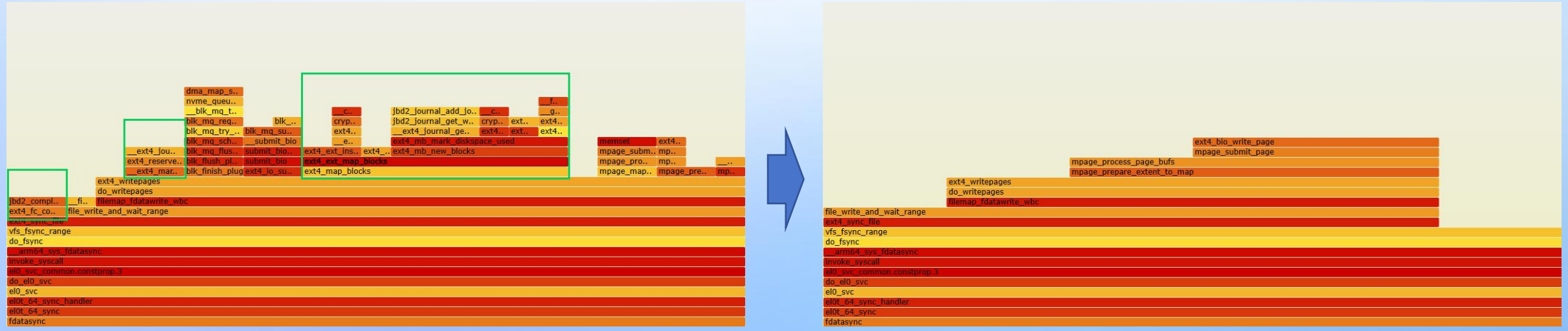


FIO: write iodepth=1 bs=4K psync numjobs=1

硬件环境：Intel Xeon Gold 6248R CPU + KaiXia 7TB KCD61LUL7T68 NVMe SSD环境测试

技术效果

- **创建**：文件预分配时间相较write zeroes大幅缩短，高于fallocate，不占用磁盘I/O带宽
- **覆盖写性能**：预分配文件后期覆盖写时延约等于write zeroes，相较于fallocate大幅缩短
- **干扰**：带宽干扰约为0，可跑满磁盘带宽，时延干扰~12%



回写时不涉及元数据和日志相关操作

技术效果

- **创建**：文件预分配时间相较write zeroes大幅缩短，高于fallocate，不占用磁盘I/O带宽
- **覆盖写性能**：预分配文件后期覆盖写时延约等于write zeroes，相较于fallocate大幅缩短
- **干扰**：带宽干扰约为0，可跑满磁盘带宽，时延干扰~12%

	Fallocate	Write Zeroes	Unmap Zeroes
预分配耗时	短	长	较短
覆盖写(同步)性能	差	好	好
带宽干扰	小	大	小
时延干扰	小	大	较小

收益和限制场景讨论

- 收益场景
  - 数据库 Redo Log/Binlog 预分配，动态文件扩展；
  - 文件系统格式化、block初始化和ext4 lazy init等；
  - …
- 限制
  - 依赖硬件FTL Unmap Write Zeroes能力，非对齐下发可能还是会实际写“0”，性能不一定最优；
  - 适用“原地写”文件系统，“Always COW”文件无收益；
  - …

社区状态

- 社区状态：特性已合入Linux kernel v6.17，blktests已ready，xfstests相关测试套补充中；
- Commit 链接：
  https://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux.git/commit/?id=278c7d9b5e0c
- 社区News：
  - https://www.phoronix.com/news/Linux-6.17-fallocate-Write-Zero
  - https://kernelnewbies.org/LinuxChanges#Linux_6.17.Introduce_a_new_fallocate.282.29_flag.2C_for_more_efficient_writing_of_zeroes

THANKS