# 赛题 14 #

StratoVirt 的 virtio-block 设备后端支持配置 io_uring

直播导师：张亮

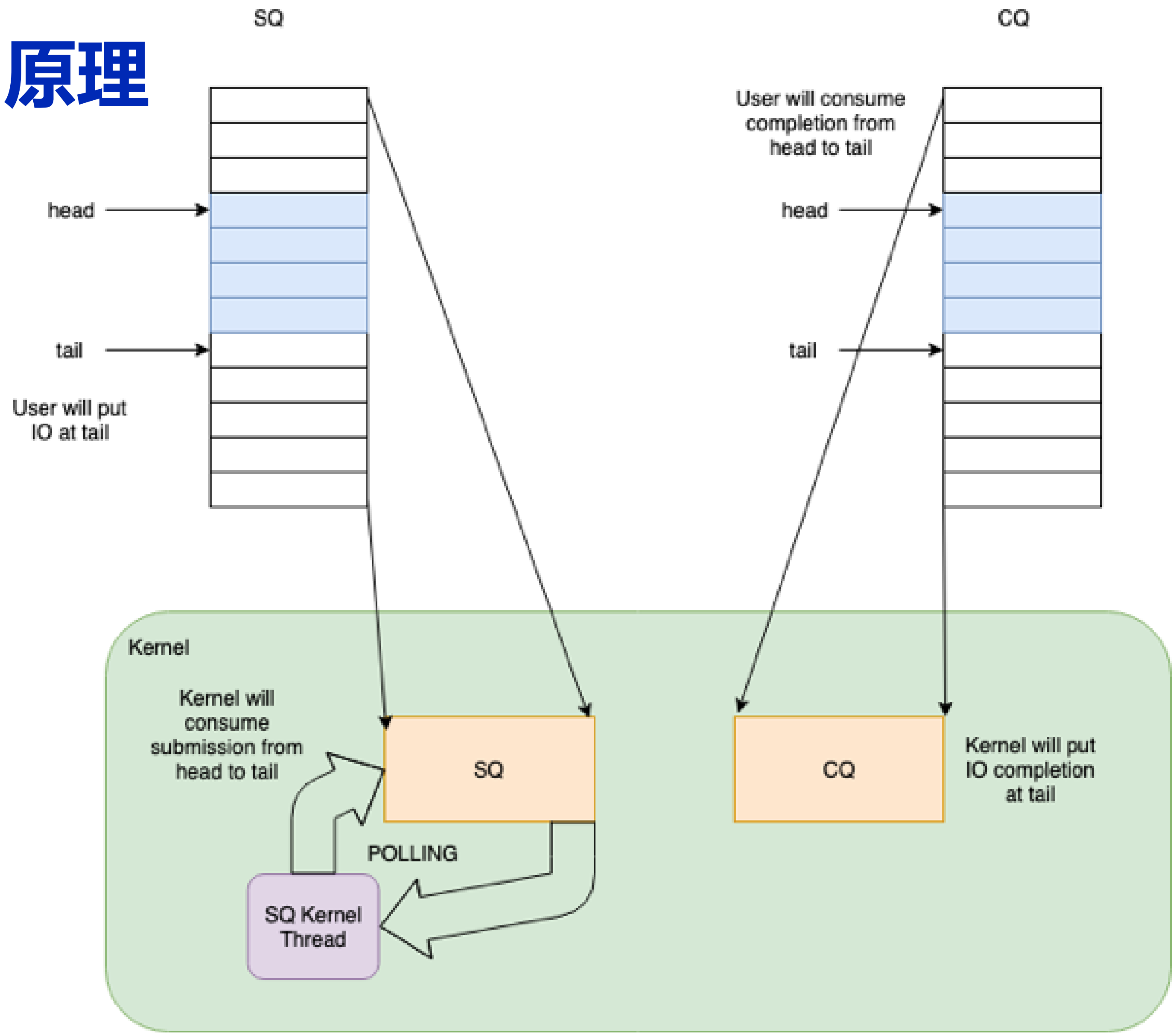
目标

StratoVirt 目前已经集成异步 io（aio），性能较竞品提升 50% 以上。但是在云化场景下，追求性能提升是永恒的主题。

在 linux 内核 5.1 版本中合入的 io_uring 是 aio 的升级版，在一些场景下，性能较 aio 提升明显。

为了进一步提升 StratoVirt 的 io 性能，需要合入 io_uring 特性，目标提升 4K 随机读写性能 30% 以上。

io_uring 原理


SQ
CQ
User will consume completion from head to tail
head
tail
head
tail
User will put IO at tail
Kernel
Kernel will consume submission from head to tail
SQ
CQ
Kernel will put IO completion at tail
POLLING
SQ Kernel Thread

示例伪代码：（参考链接）

```
#include <linux/io_uring.h>
int io_uring_setup(usigned entries, struct io_uring_params *p) {
    return (int) syscall(__NR_io_uring_setup, entries, p);
}

int io_uring_enter(int ring_fd, unsigned int to_submit, unsigned int min_complete, unsigned int flags) {
    return (int) syscall(__NR_io_uring_enter, ring_fd, to_submit, min_complete, flags, NULL, 0);
}

int io_uring_register(int ring_fd, unsigned int opcode, void *arg, unsigned int nr_args) {
    return (int) syscall(__NR_io_uring_enter, ring_fd, opcode, arg, nr_args);
}

void main() {
    struct io_uring_params p;
    int fd = io_uring_setup(128, &p);   // 128是队列深度

    struct io_sqring_offsets *sq_off = p.sq_off;
    struct io_cqring_offsets *cq_off = p.cq_off;

    int sring_size = sq_off.array + p.sq_entries * sizeof(unsigned);
    int cring_size = cq_off.cqes + p.cq_entries * sizeof(struct io_uring_cqe);

    void *sq_ptr = mmap(0, sring_size, PROT_READ | PROT_WRITE, MAP_SHARED | MAP_POPULATE, fd, IORING_OFF_SQ_RING);
    void *cq_ptr = mmap(0, cring_size, PROT_READ | PROT_WRITE, MAP_SHARED | MAP_POPULATE, fd, IORING_OFF_CQ_RING);

    struct io_uring_sqe *sqes = mmap(0, p.sq_entries * sizeof(struct io_uring_sqe),  PROT_READ | PROT_WRITE, MAP_SHARED | MAP_POPULATE, fd, IORING_OFF_SQES);

    char buff[100];
    struct io_uring_sqe sqe = sqes[0];
    sqe->fd = file_fd;   // 要读的文件fd
    sqe->flags = 0;
    sqe->opcode = IORING_OP_READ_FIXED;  // enum
    sqe->addr = buff;
    sqe->len = 100;
    sqe->off = 0;
    sqe->user_data = NULL;

    io_uring_enter(fd, 1, 1, IORING_ENTER_GETEVENTS);
}
```

详细要求

1、用 rust 语言实现 linux/io_uring.h 的 5 个结构体和 1 个枚举：

struct **io_uring_params**

struct **io_sqring_offsets**

struct **io_cqring_offsets**

struct **io_uring_sqe**

struct **io_uring_cqe**

enum **IORING_OP_NOP ~ IORING_OP_LAST**

2、用 rust 语言封装 3 个系统调用：

syscall(__NR_io_uring_setup, entries, p)

syscall(__NR_io_uring_enter, ring_fd, to_submit, min_complete, flags, NULL, 0)

syscall(__NR_io_uring_enter, ring_fd, opcode, arg, nr_args)

3、用 rust 语言实现上页的示例代码

4、集成 rust 实现到 StratoVirt 中