z_loss / logits打点 与 ce-loss 算子fuse

1.4卡下的性能优化（100步）：

trial链接：https://seed.byteintl.net/experiment/tracking/detail?Id=project_20260317_6645be31&selectedTrial=&tab=CHART&viewId=view_20260426_77eb98a9

峰值显存Memory （pp0下降约1.5G，pp1上升约0.6G)：

Mfu 大部分step均有提升：

2.修改kernel前后的变化

a. 针对z_loss:

```python
def _z_loss_fwd_kernel(
    logits_ptr,
    z_loss_ptr,
    z_loss_weight,
    V,
    stride_n,
    BLOCK_V: tl.constexpr,      李想, 4 weeks ago • [Pretrain] feat: support
):
    row = tl.program_id(0)
    logits_row_ptr = logits_ptr + row * stride_n

    # Pass 1: max of abs(x) for numerical stability
    max_val = -float("inf")
    for start_v in range(0, V, BLOCK_V):
        v_offs = start_v + tl.arange(0, BLOCK_V)
        mask = v_offs < V
        x = tl.load(logits_row_ptr + v_offs, mask=mask, other=0.0).to(tl.flo
        abs_x = tl.abs(x)
        block_max = tl.max(tl.where(mask, abs_x, -float("inf")), axis=0)
        max_val = tl.maximum(max_val, block_max)

    # Pass 2: sum of exp(abs(x) - max_val)
    sum_exp = 0.0
    for start_v in range(0, V, BLOCK_V):
        v_offs = start_v + tl.arange(0, BLOCK_V)
        mask = v_offs < V
        x = tl.load(logits_row_ptr + v_offs, mask=mask, other=0.0).to(tl.flo
        abs_x = tl.abs(x)
        sum_exp += tl.sum(tl.where(mask, tl.exp(abs_x - max_val), 0.0), axis

    lse = max_val + tl.log(sum_exp)
    z_loss_val = lse * lse * z_loss_weight
    tl.store(z_loss_ptr + row, z_loss_val)
```

可以看到原来是两次访存logits，第一次先算出最大值，第二次算出lse；而由于ce-loss这边是做的online softmax，因此统一一次访存，区别：

1.前者一次性算出max，然后算出sum，不需要更新max

2.后者需要每次更新max

如果不同block间的max差异较大，这个更新的数 old_sum*e(old_max - new_max)非常小，可能会精度下溢，则产生微小误差，但数学上是等价的，因此可以接受。（注意和原来的实现不一定能每一步bitewise对齐）

b. 针对logits（min、max、mean）

```python
def local_logits_sums_for_mean_max_min(logits_2d: torch.Tensor) -> torch.Tensor
    """Per-rank sums for logits [N, V] (same layout as z_loss_forward input).

    Returns float32 tensor [sum(mean_row), sum(max_row), sum(min_row), N] for o
    """
    with torch.no_grad():
        lf = logits_2d.detach().float()
        if lf.numel() == 0:
            return torch.zeros(4, device=logits_2d.device, dtype=torch.float32)
        row_mean = lf.mean(dim=-1)
        row_max = lf.amax(dim=-1)
        row_min = lf.amin(dim=-1)
        n = torch.full((), lf.shape[0], device=lf.device, dtype=torch.float32)
        return torch.stack([row_mean.sum(), row_max.sum(), row_min.sum(), n])
```

区别：

- 原来是python端实现，多次访存，规约顺序一致
- 一次访存，这里采用的atomic_add做最后的row_mean,row_max,row_min的sum（非deterministic），这里不知道是否有影响，如果有影响可以换成（kernel内部保存每个token的这三个值，即3*seq_len的大小，然后再在python端做一个sum。但这种操作会增加显存和降低mfu）
  - 不能bitwise对齐，符合预期

c. 针对ce-loss

区别：

1. 原来是online softmax block级别的规约（一个block处理一个token的logits向量，block内部512个线程，每个线程每步处理8个元素(也可以是2，4，根据hidden_size)），但cuda kernel可以精细控制thread，处理顺序：每个thread跳步处理，每次处理8个数据，跳步大小为block_size*8，处理

完后，1个block中512个数据，每32个数据作为一个wrap进行处理，得到16个数据，最后再在warp间做一次规约，得到当前block的最终数据。

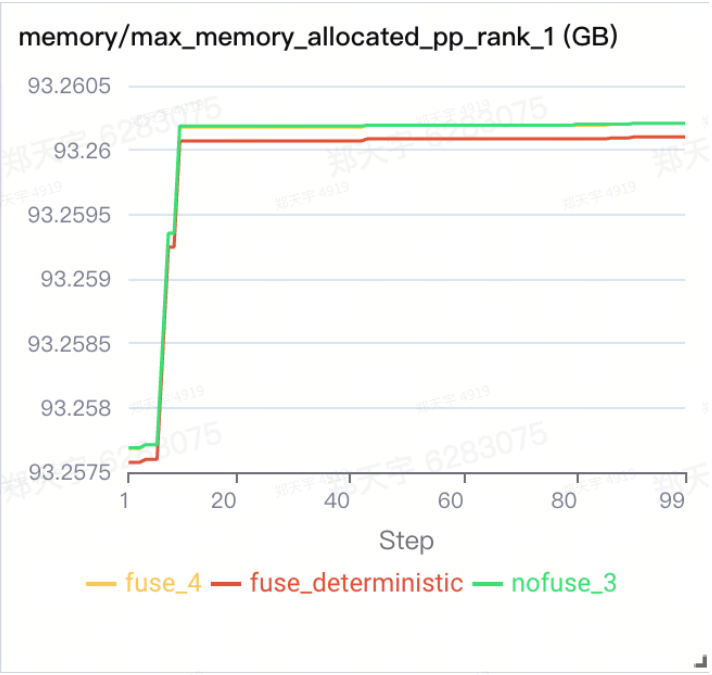
2. 当前和cuda kernel对齐，也做online softmax，triton是一个block内并行处理，但底层规约无法干预，然后顺序更新。

不能bitewise对齐，符合预期。

3.讨论是否调整为logits的指标为determinstic：

logits调整为deterministic，性能几乎差不多，相比非deterministic，部分step有所提升

memory结论也基本一致（甚至当前对比，gpu0、gpu1峰值显存都下降了）：



因此调整成logits也deterministic。

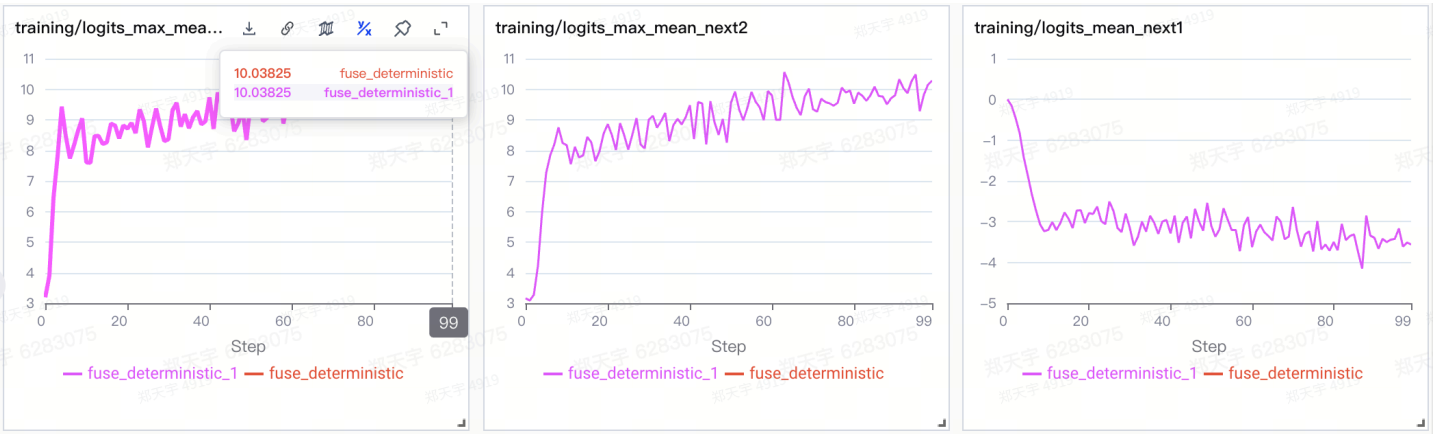
4.确定fuse kernel 是deterministic

trial:https://seed.byteintl.net/experiment/tracking/detail?Id=project_20260317_6645be31&selectedTrial=&tab=CHART&viewId=view_20260427_a2ba3a51

1. loss

2. z_loss

3. logits指标

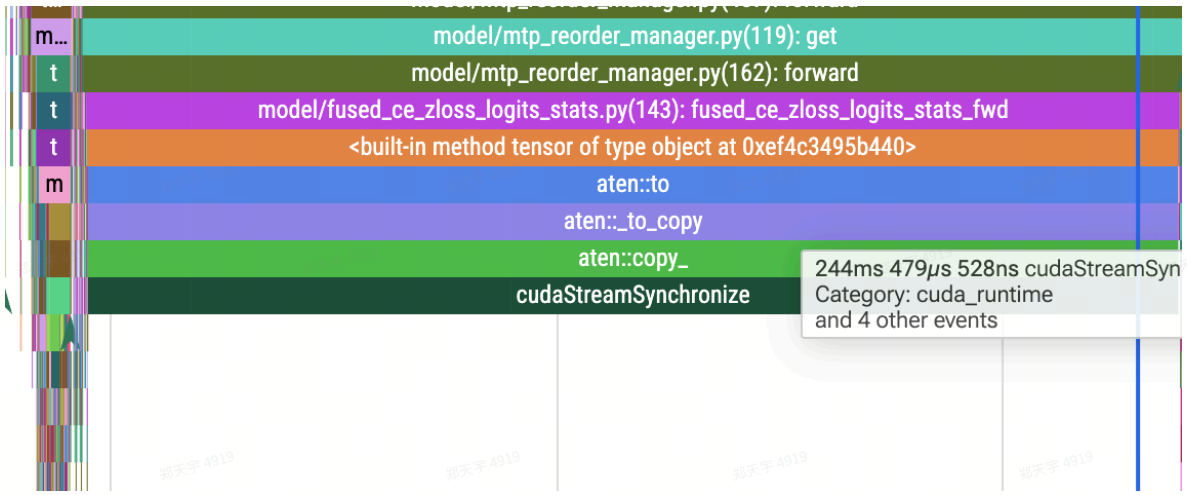


5.消除 stats 拼装的隐式 stream 同步

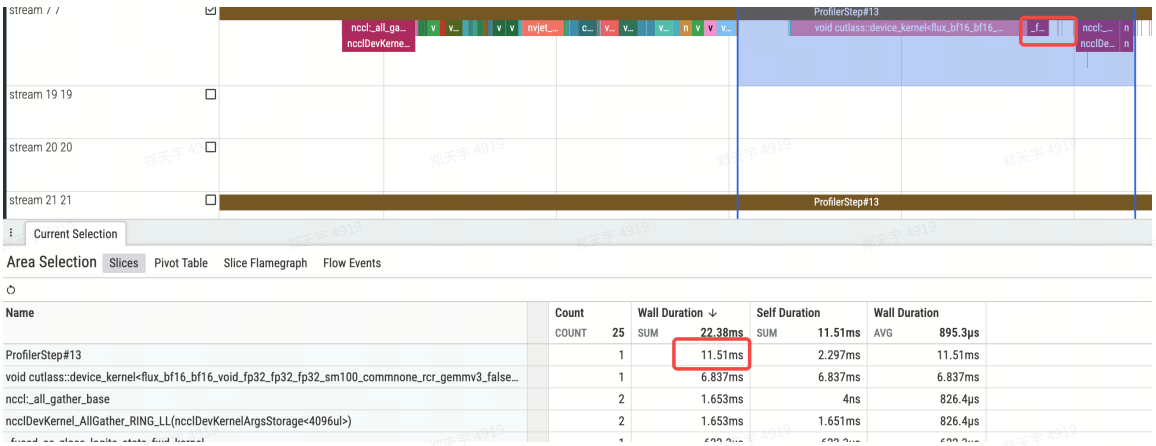
torch.tensor加torch.cat会发起一个隐式的cudastream synchronize，导致cpu straggle，且gpu kernel时间延长，改用torch.empty，并fill数据，进行优化，优化后，cpu 时间从244ms，缩短至0.568ms，gpu kernel从11.5 ms优化至9.4 ms

优化前：

CPU:

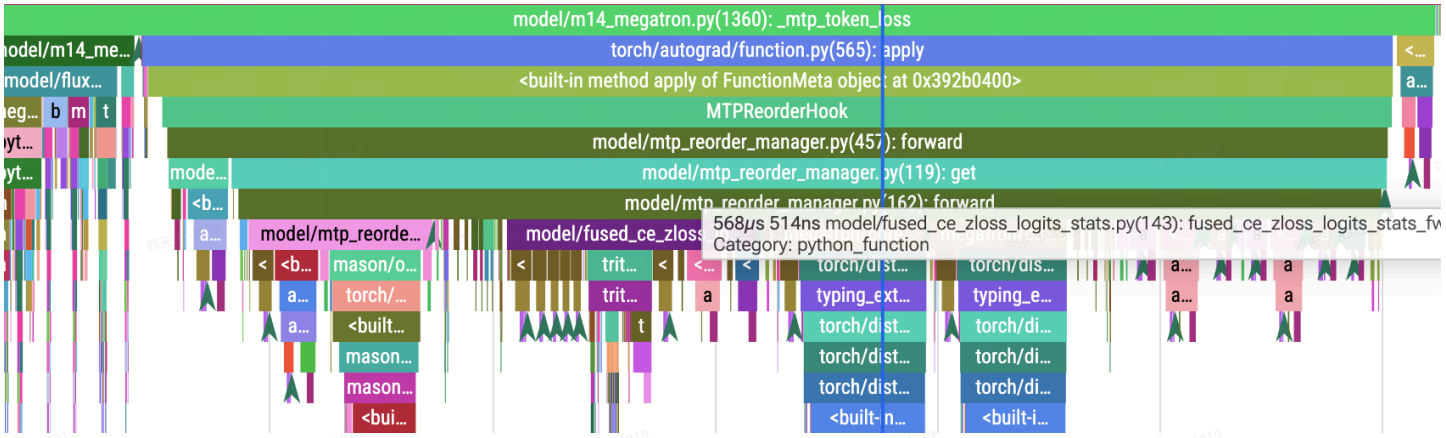


GPU:

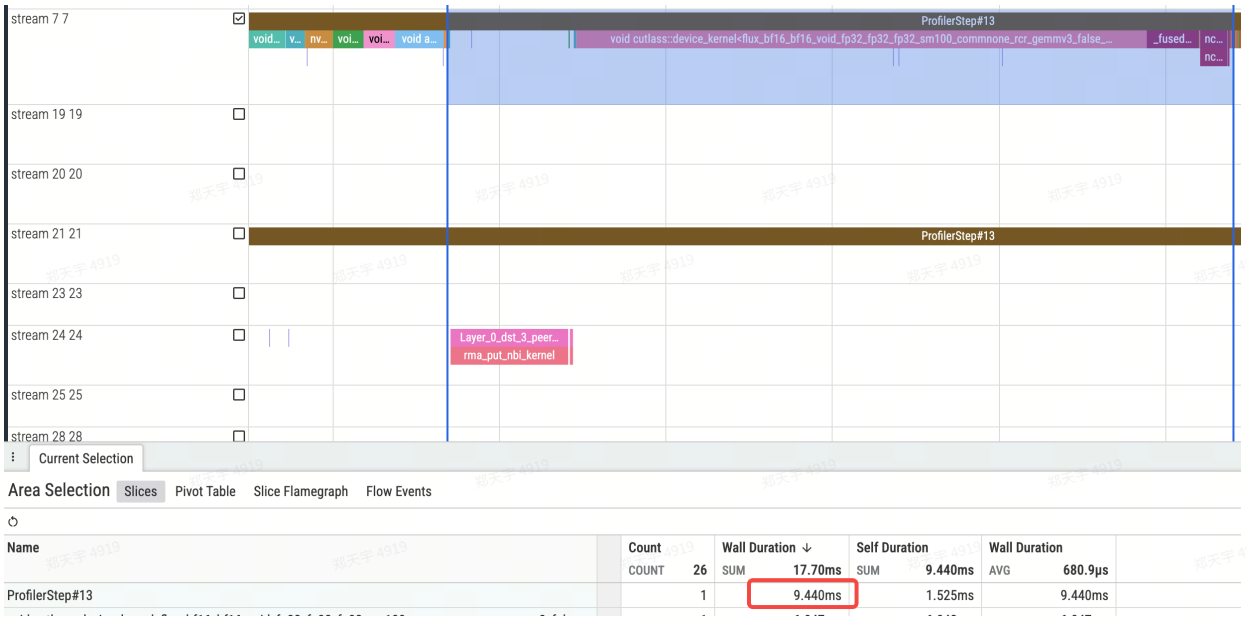


优化后：

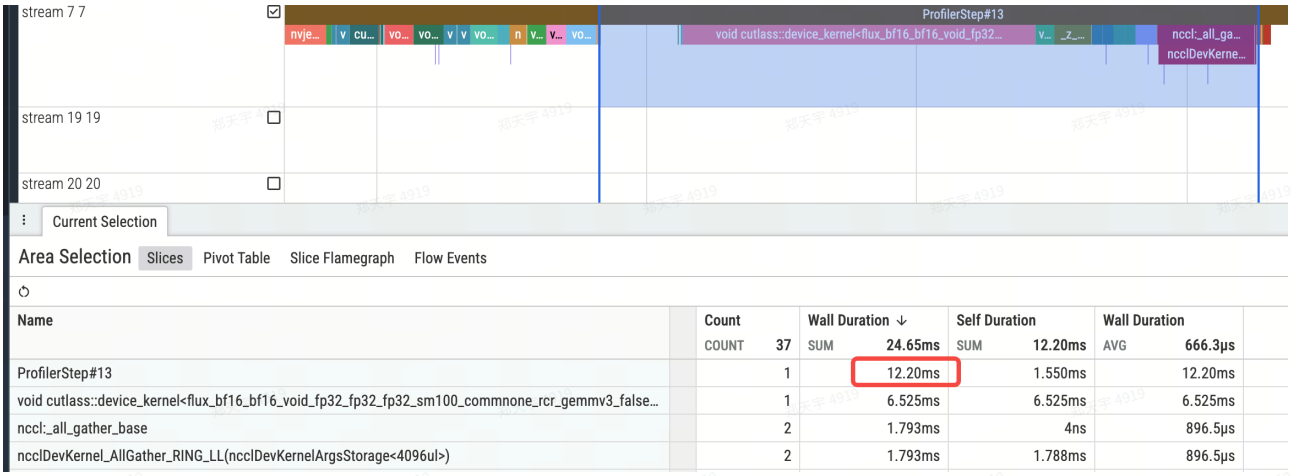
CPU:



GPU:



贴一个非fuse的baseline：



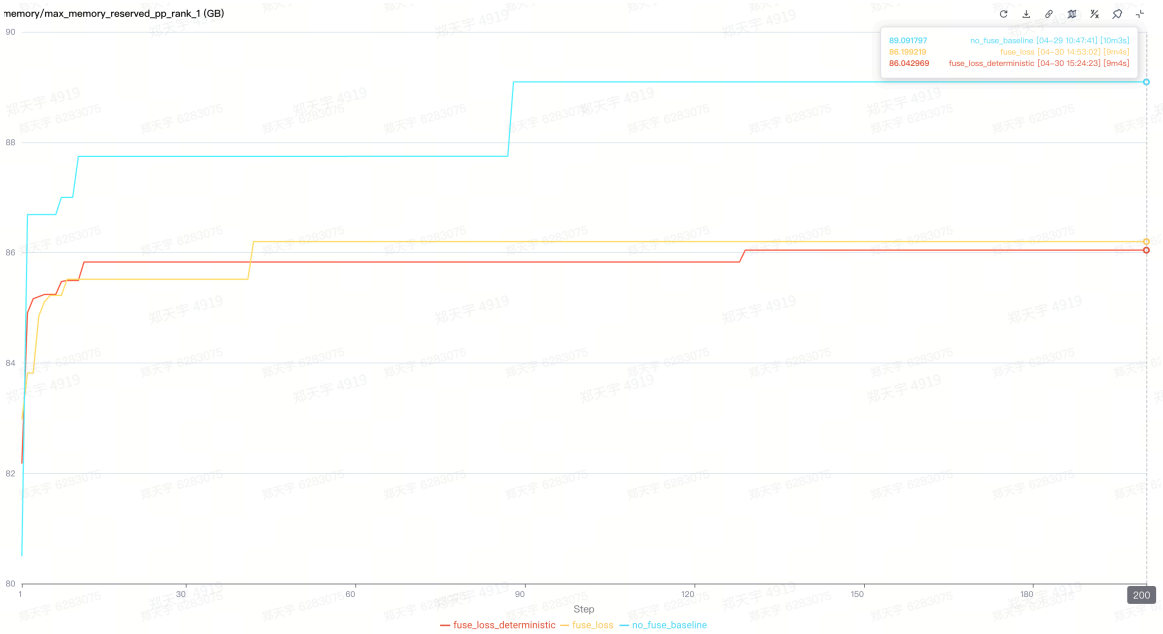
证明修改后是determinstic的：

Mfu:

显著提高

Memory：

pp-1的显存也降低比较明显



6. 实现inplace_bwd

对fuse kernel做优化，因为logits在backward本次数据计算完成之后，后续不会再被使用，因此直接用logits的地址来存grad_loss，从而节省显存

mr：

1. https://code.byted.org/seed/mariana/merge_requests/15029
2. https://code.byted.org/seed/mariana/merge_requests/15108
3. https://code.byted.org/seed/mariana/merge_requests/15165