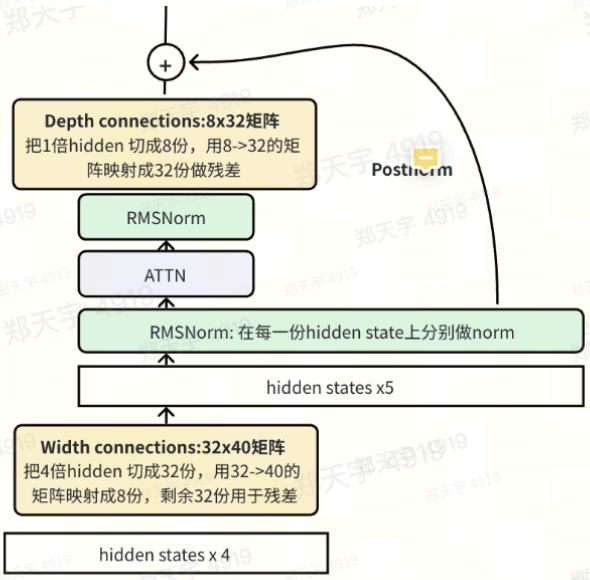
VWN_V2 fuse kernel接入big_op

1.v2的一些改进

1. 分块处理

当前n=32，m=8，计算时，A和B矩阵拼在一起，是[:,mnm,:]，mnm=48，直接计算从48 padding到64比较浪费，因此分成32和16来处理

2. width_connection和depth_connection都做fuse kernel
3. residual放在depth_connection计算并相加



M14.2

如果是postnorm，这里需要对每一份hidden state做norm，而这个操作可以fuse到depth connections，且对于5*h里的其中的后面4*h，可以不在width connection中计算，而是放到depth connection计算，并和resnorm做fuse。

4. width connections仅输出h大小的数据用于attn/ffn计算，并且beta变成 alpha_res_beta，alpha_res即用于算residual那部分的alpha。

2.v2目前可优化的点

1. Todo： width connection的输出一定会做resnorm，因此也可以fuse进kernel里面

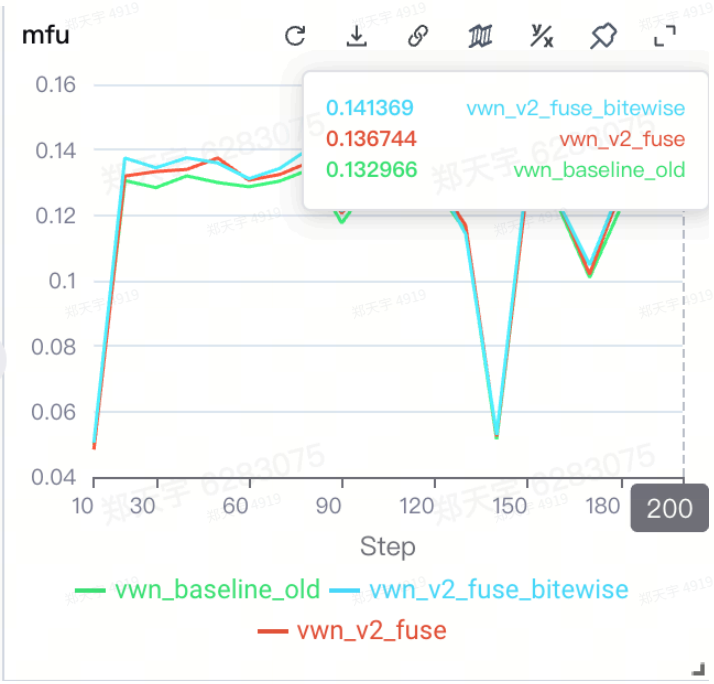
3.当前的实验效果

实验的trail:https://seed.byteintl.net/experiment/tracking/detail?Id=project_20260317_6645be31&selectedTrial=&tab=CHART&viewId=view_20260508_c543a772

loss：不能bitweise对齐，误差在预期范围内，符合预期。fuse_v2 验证是deterministic的。

mfu：

mfu提升0.009符合预期，且多次实验均显著高于baseline



二次实验，mfu提升比较稳定

memory：

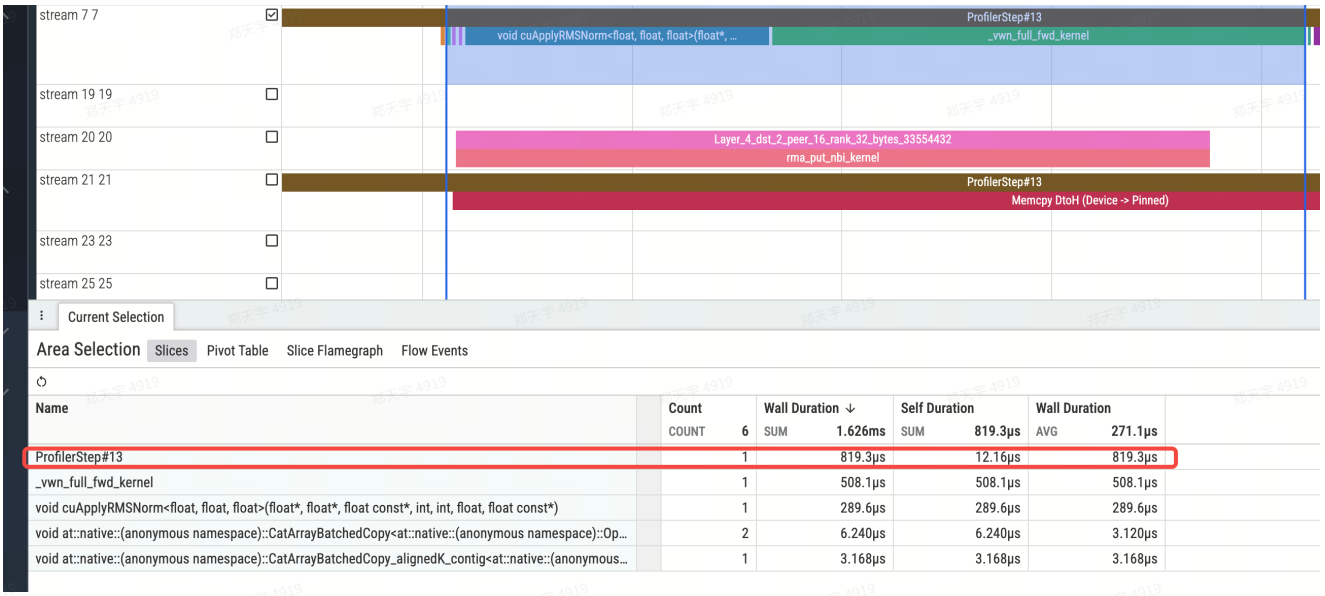
memory在pp-3上节省约4个G，符合预期，且每个pp rank上均显著下降

kernel执行时间：

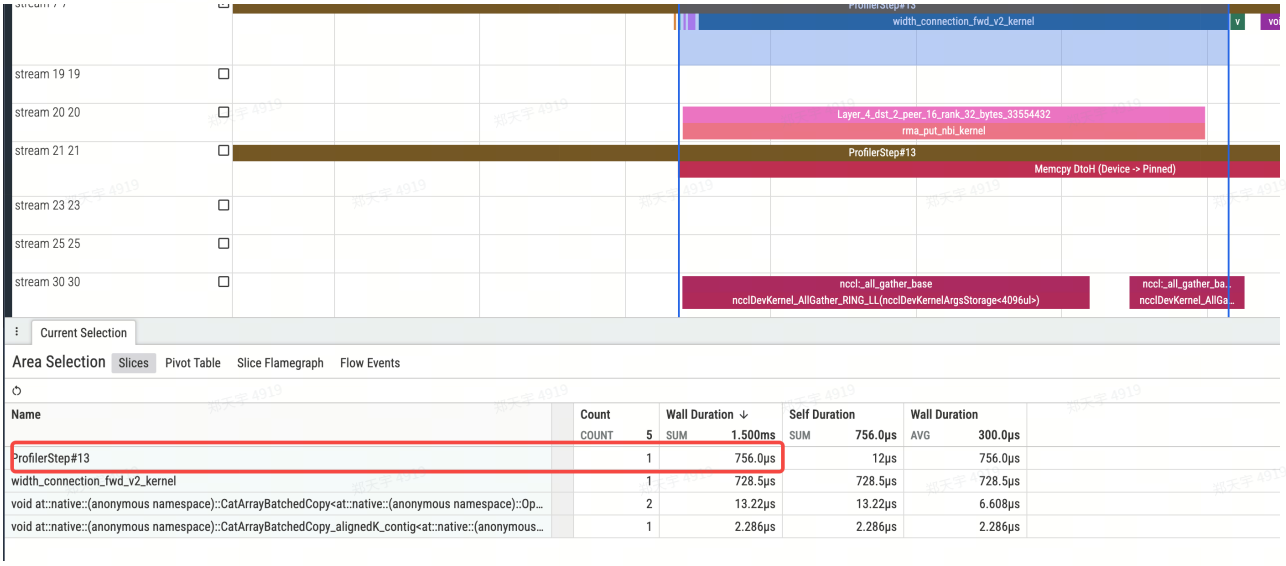
- width_connection

819us->756us

before：



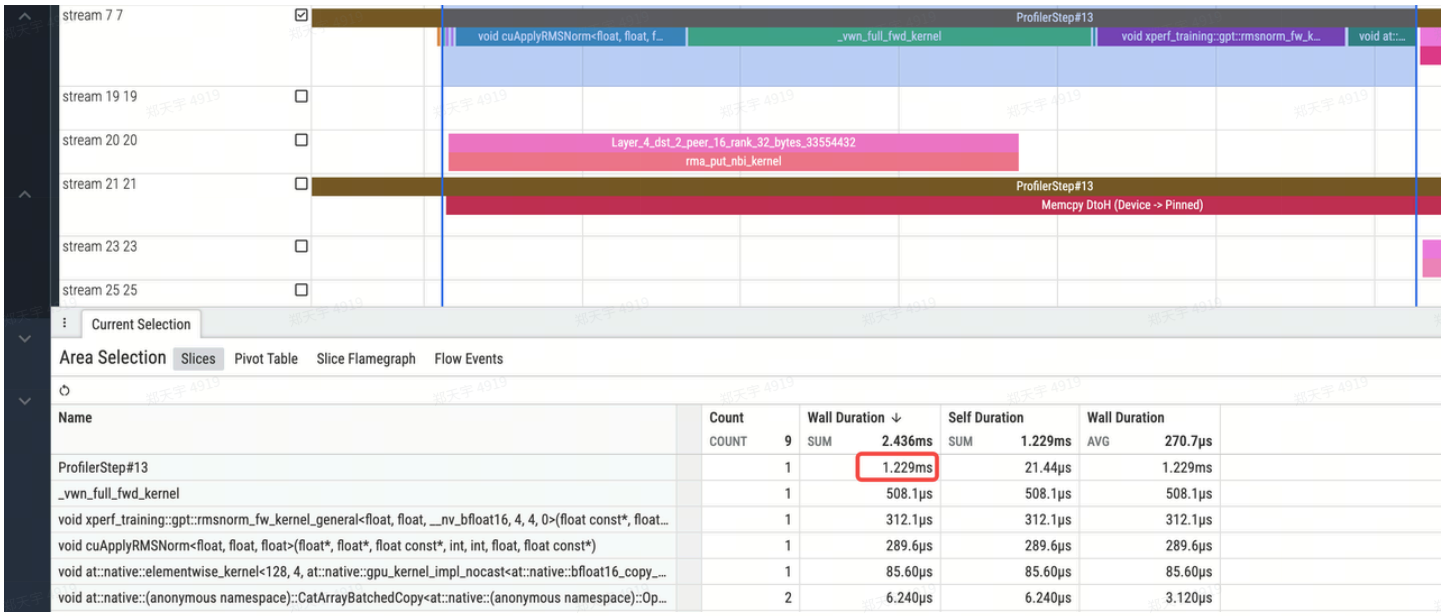
after:



- width_connection+rmsnorm

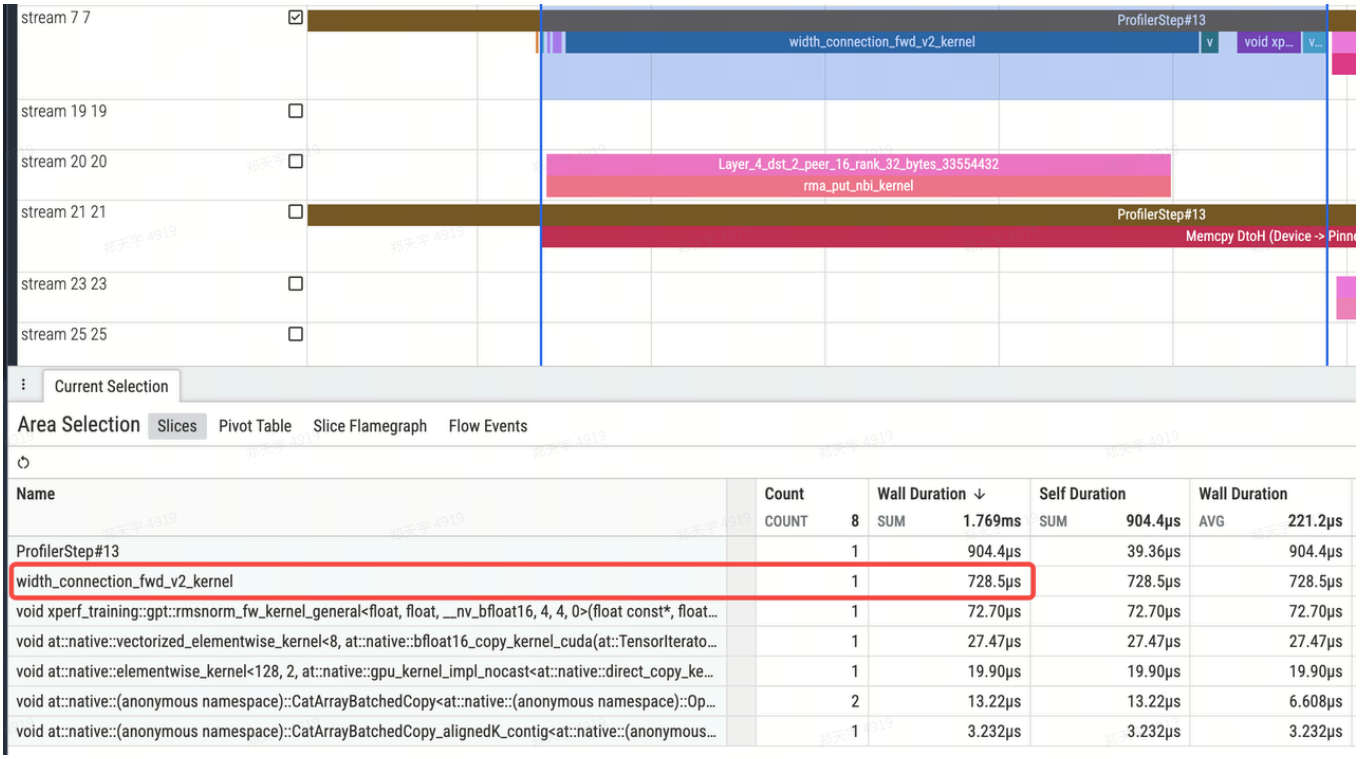
1.229->0.904 ms

before:



Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 9	SUM 2.436ms	SUM 1.229ms	AVG 270.7µs
ProfilerStep#13	1	1.229ms	21.44µs	1.229ms
_vwn_full_fwd_kernel	1	508.1µs	508.1µs	508.1µs
void xperf_training::gpt::rmsnorm_fw_kernel_general<float, float, __nv_bfloat16, 4, 4, 0>(float const*, float...	1	312.1µs	312.1µs	312.1µs
void cuApplyRMSNorm<float, float, float>(float*, float*, float const*, int, int, float, float const*)	1	289.6µs	289.6µs	289.6µs
void at::native::elementwise_kernel<128, 4, at::native::gpu_kernel_impl_nocast<at::native::bfloat16_copy_...	1	85.60µs	85.60µs	85.60µs
void at::native::(anonymous namespace)::CatArrayBatchedCopy<at::native::(anonymous namespace)::Op...	2	6.240µs	6.240µs	3.120µs

after:

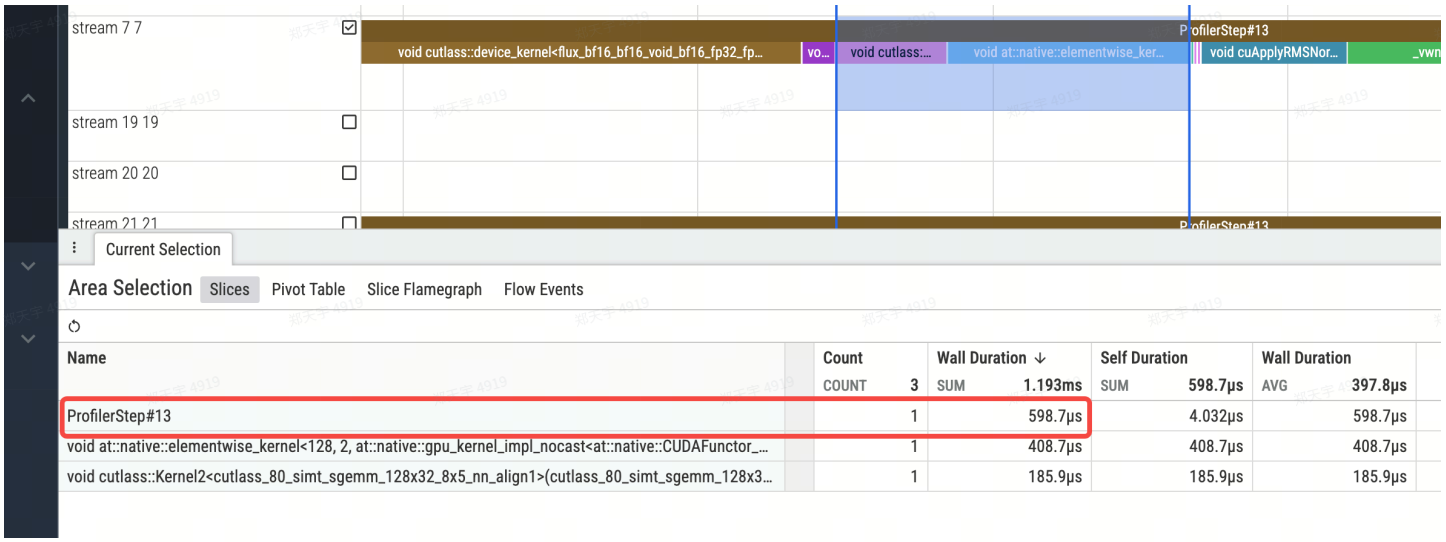


Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 8	SUM 1.769ms	SUM 904.4µs	AVG 221.2µs
ProfilerStep#13	1	904.4µs	39.36µs	904.4µs
width_connection_fwd_v2_kernel	1	728.5µs	728.5µs	728.5µs
void xperf_training::gpt::rmsnorm_fw_kernel_general<float, float, __nv_bfloat16, 4, 4, 0>(float const*, float...	1	72.70µs	72.70µs	72.70µs
void at::native::vectorized_elementwise_kernel<8, at::native::bfloat16_copy_kernel_cuda(at::TensorIterato...	1	27.47µs	27.47µs	27.47µs
void at::native::elementwise_kernel<128, 2, at::native::gpu_kernel_impl_nocast<at::native::direct_copy_ke...	1	19.90µs	19.90µs	19.90µs
void at::native::(anonymous namespace)::CatArrayBatchedCopy<at::native::(anonymous namespace)::Op...	2	13.22µs	13.22µs	6.608µs
void at::native::(anonymous namespace)::CatArrayBatchedCopy_alignedK_contig<at::native::(anonymous...	1	3.232µs	3.232µs	3.232µs

- depth_connection

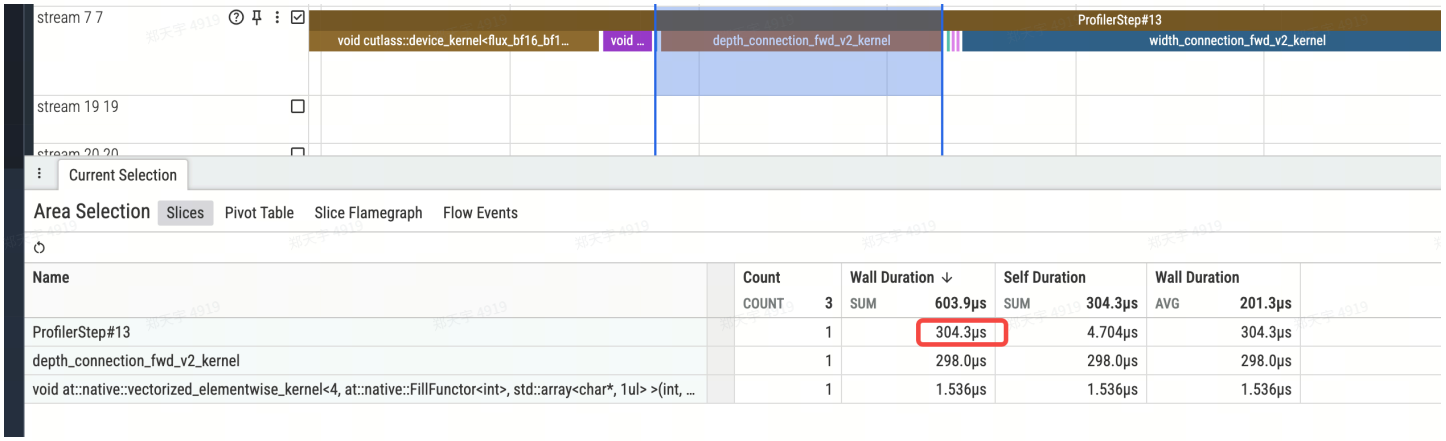
598us->304us

before:



Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 3	SUM 1.193ms	SUM 598.7µs	AVG 397.8µs
ProfilerStep#13	1	598.7µs	4.032µs	598.7µs
void at::native::elementwise_kernel<128, 2, at::native::gpu_kernel_impl_nocast<at::native::CUDAFunctor_...	1	408.7µs	408.7µs	408.7µs
void cutlass::Kernel2<cutlass_80_simt_sgemm_128x32_8x5_nn_align1>(cutlass_80_simt_sgemm_128x3...	1	185.9µs	185.9µs	185.9µs

after:

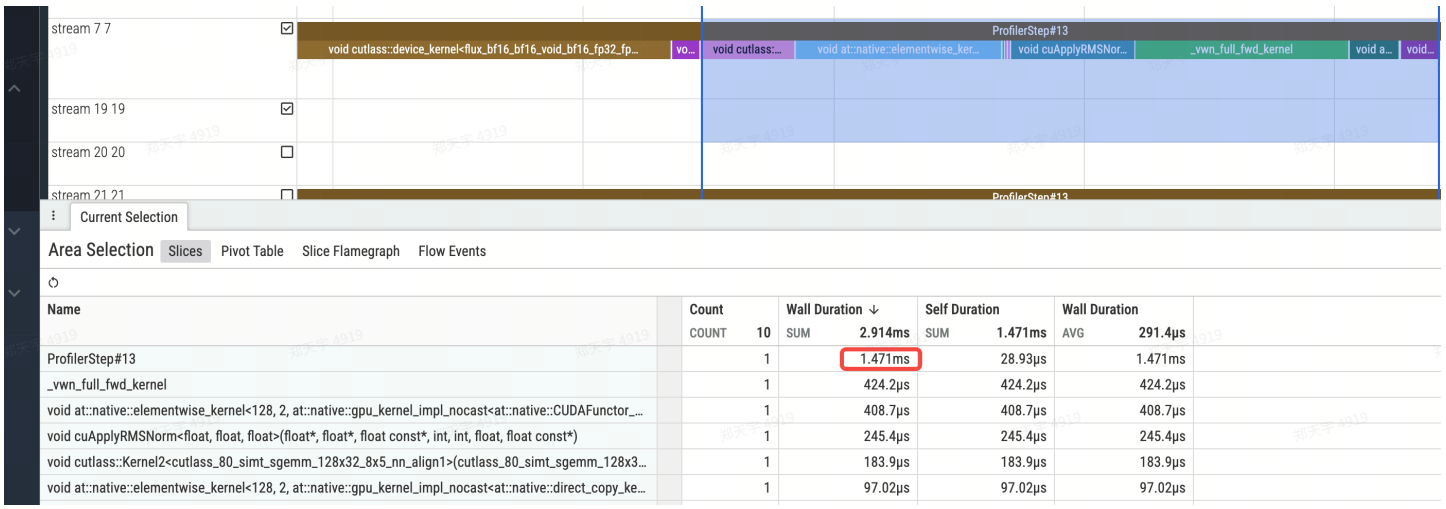


Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 3	SUM 603.9µs	SUM 304.3µs	AVG 201.3µs
ProfilerStep#13	1	304.3µs	4.704µs	304.3µs
depth_connection_fwd_v2_kernel	1	298.0µs	298.0µs	298.0µs
void at::native::vectorized_elementwise_kernel<4, at::native::FillFunctor<int>, std::array<char*, 1ul> >(int, ...	1	1.536µs	1.536µs	1.536µs

- 第一个depth_connection+第二个width_connection+rmsnorm

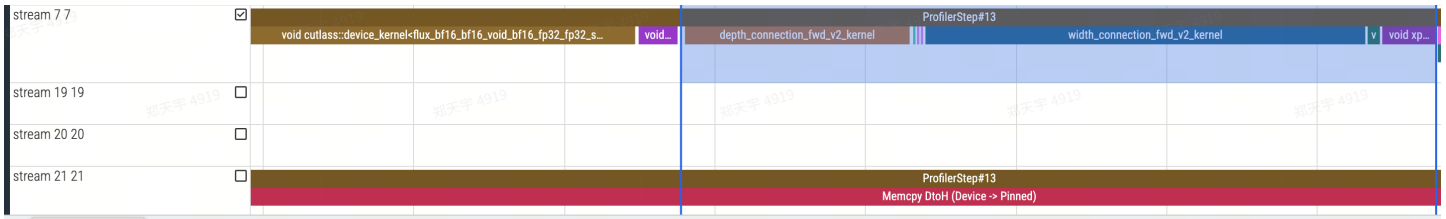
1.47ms->1ms

before：



Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 10	SUM 2.914ms	SUM 1.471ms	AVG 291.4µs
ProfilerStep#13	1	1.471ms	28.93µs	1.471ms
_vwn_full_fwd_kernel	1	424.2µs	424.2µs	424.2µs
void at::native::elementwise_kernel<128, 2, at::native::gpu_kernel_impl_nocast<at::native::CUDAFunctor_...	1	408.7µs	408.7µs	408.7µs
void cuApplyRMSNorm<float, float, float>(float*, float*, float const*, int, int, float, float const*)	1	245.4µs	245.4µs	245.4µs
void cutlass::Kernel2<cutlass_80_simt_sgemm_128x32_8x5_nn_align1>(cutlass_80_simt_sgemm_128x3...	1	183.9µs	183.9µs	183.9µs
void at::native::elementwise_kernel<128, 2, at::native::gpu_kernel_impl_nocast<at::native::direct_copy_ke...	1	97.02µs	97.02µs	97.02µs

after:



Current Selection

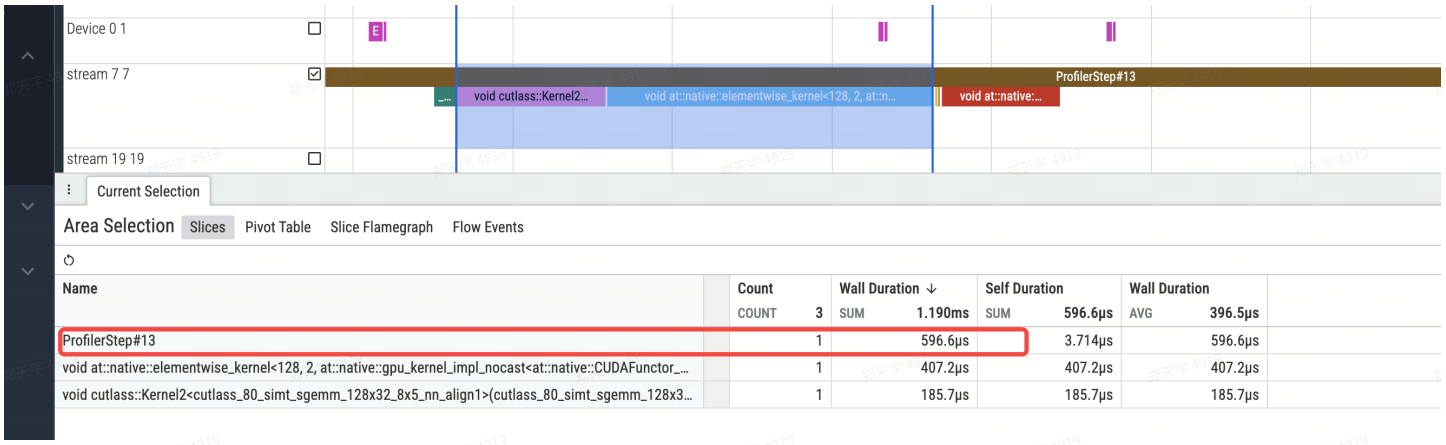
Area Selection | Slices | Pivot Table | Slice Flamegraph | Flow Events

Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 9	SUM 1.982ms	SUM 1.003ms	AVG 220.2µs
ProfilerStep#13	1	1.003ms	23.87µs	1.003ms
width_connection_fwd_v2_kernel	1	583.9µs	583.9µs	583.9µs
depth_connection_fwd_v2_kernel	1	298.0µs	298.0µs	298.0µs
void xperf_training::gpt::rmsnorm_fw_kernel_general<float, float, __nv_bfloat16, 4, 4, 0>(float const*, float...	1	71.30µs	71.30µs	71.30µs
void at::native::elementwise_kernel<128, 2, at::native::gpu_kernel_impl_nocast<at::native::direct_copy_ke...	1	16.19µs	16.19µs	16.19µs

- 第二个depth_connection

596us->224us

before:

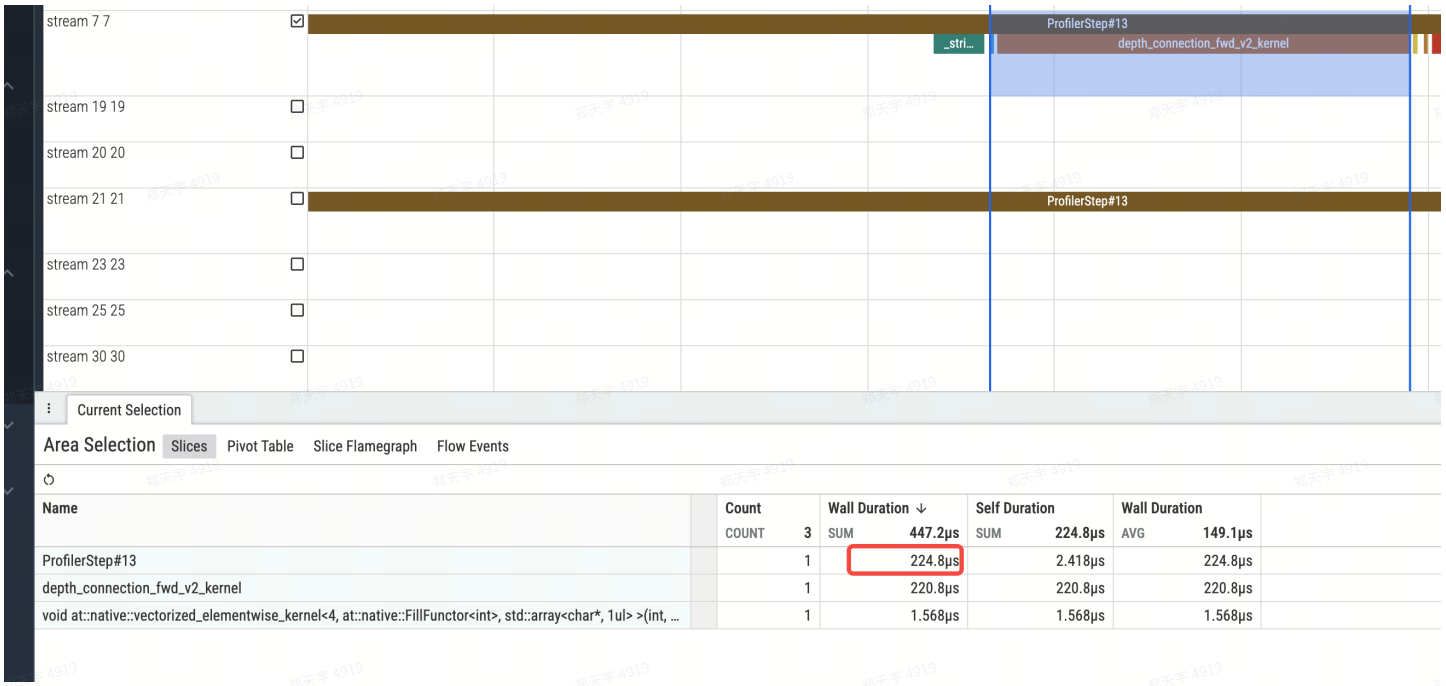


Current Selection

Area Selection | Slices | Pivot Table | Slice Flamegraph | Flow Events

Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 3	SUM 1.190ms	SUM 596.6µs	AVG 396.5µs
ProfilerStep#13	1	596.6µs	3.714µs	596.6µs
void at::native::elementwise_kernel<128, 2, at::native::gpu_kernel_impl_nocast<at::native::CUDAFunctor_...	1	407.2µs	407.2µs	407.2µs
void cutlass::Kernel2<cutlass_80_simt_sgemm_128x32_8x5_nn_align1>(cutlass_80_simt_sgemm_128x3...	1	185.7µs	185.7µs	185.7µs

after:



Current Selection

Area Selection | Slices | Pivot Table | Slice Flamegraph | Flow Events

Name	Count	Wall Duration ↓	Self Duration	Wall Duration
	COUNT 3	SUM 447.2µs	SUM 224.8µs	AVG 149.1µs
ProfilerStep#13	1	224.8µs	2.418µs	224.8µs
depth_connection_fwd_v2_kernel	1	220.8µs	220.8µs	220.8µs
void at::native::vectorized_elementwise_kernel<4, at::native::FillFunctor<int>, std::array<char*, 1ul> >(int, ...	1	1.568µs	1.568µs	1.568µs

vwn_kernel更新后大约节省1.2ms

4.重计算

depth_connection_bwd_kerenl在执行前，需要先执行2个width_connection_fwd以及一个depth_connection。

链式法则比较好理解，并且从参数上看：depth_connection_bwd2需要beta2，而beta2由width_connection_fwd2计算得到，而width_connection_fwd2的输入即经过残差连接的4h输入，由depth_connection_fwd1得到，而depth_connection_fwd1又需要width_connection_fwd1的beta1，所以必须要做这些计算。

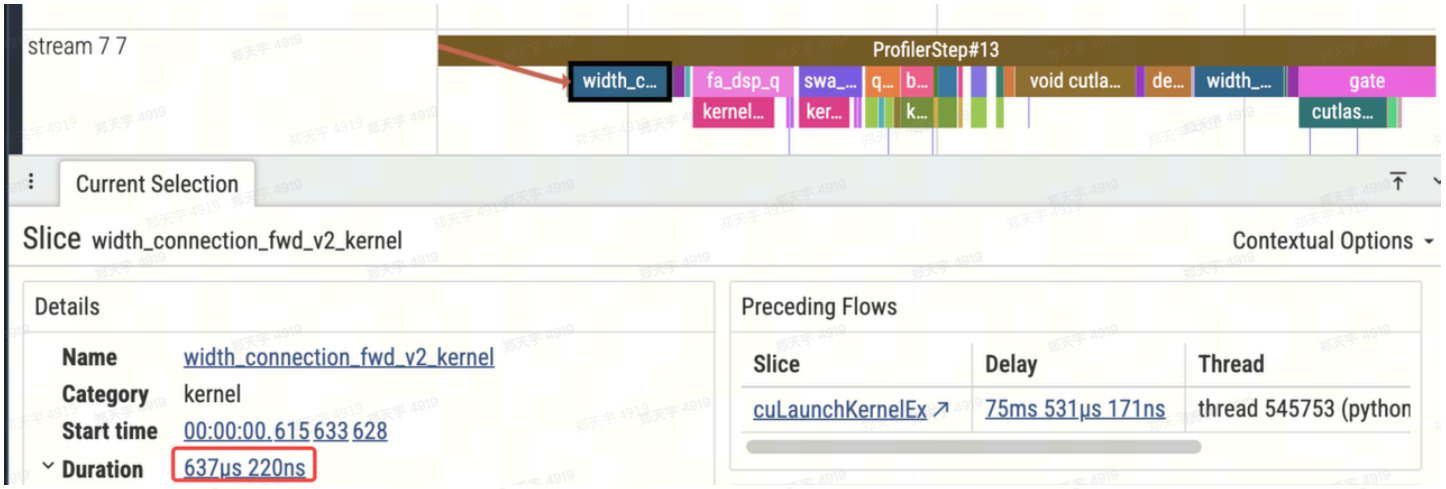
5.记录一下v2的使用(同时记录 triton 3.5版 deterministic的实验)

为了满足m=8的矩阵乘，得用triton>=3.5以上的版本，之前的版本都不支持，最低shape m，n，k都为16。

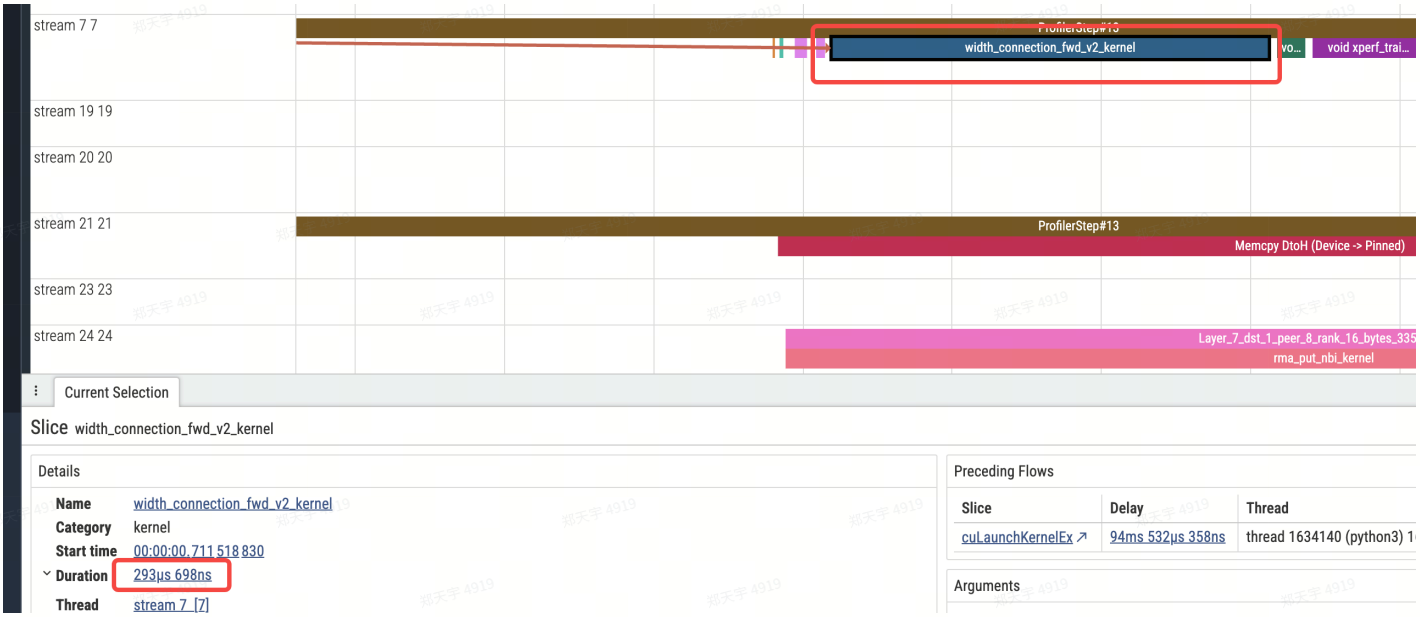
data/aml/mlsys/triton_dist_pt29_cu131_tt35	1.0.0.1	/opt/tiger/triton_dist	/opt/tiger/triton_dist/python

SCM链接：https://cloud.bytedance.net/scm/detail/532532/versions?x-resource-account=public&x-bc-region-id=bytedance

如果padding到16，width_connection的时间会增加：



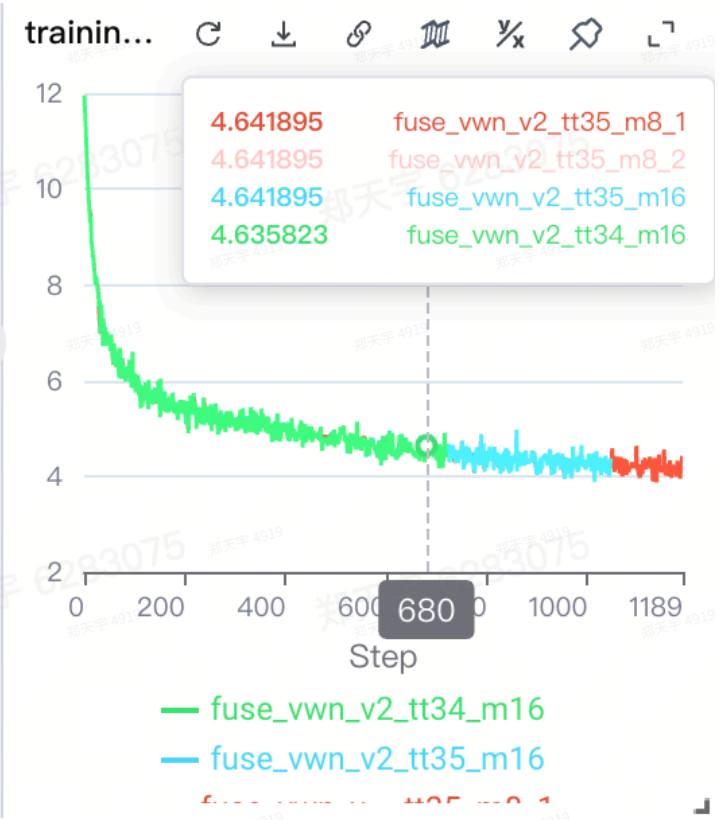
更换triton后，保持BLOCK_M=8，则width_connection的时间缩减到原来的1/2，符合预期：



- triton3.5测试:

测试链接：https://seed.byteintl.net/experiment/tracking/detail?Id=project_20260317_6645be31&selectedTrial=&tab=CHART&viewId=view_20260510_6e15bdce

loss：符合预期，blockm=8/16都是bitewise对齐的



mfu：看起来差不多

memory：峰值显存不变