ASO LAB Seminar #week6

Triton Server

엄소은

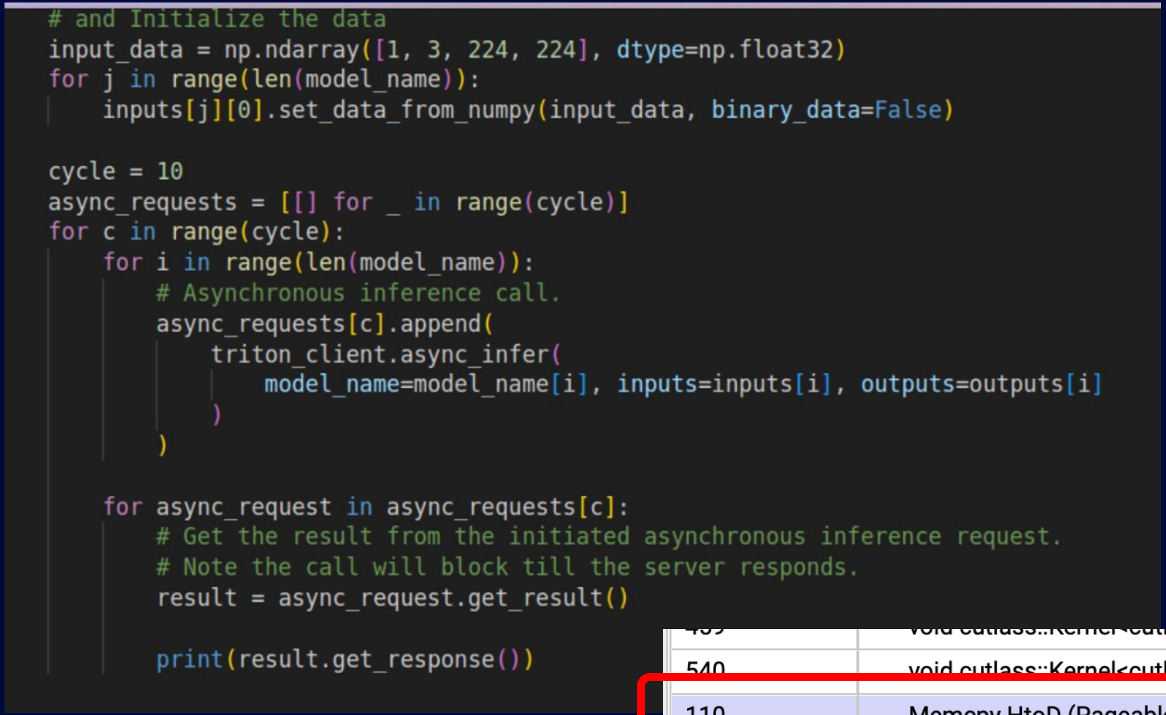
CONTENTS

01. Triton - Multiple Request

02. Local vs. Triton

1. Triton

Multiple Request - Memcpy



```
# and Initialize the data
input_data = np.ndarray([1, 3, 224, 224], dtype=np.float32)
for j in range(len(model_name)):
    inputs[j][0].set_data_from_numpy(input_data, binary_data=False)

cycle = 10
async_requests = [[] for _ in range(cycle)]
for c in range(cycle):
    for i in range(len(model_name)):
        # Asynchronous inference call.
        async_requests[c].append(
            triton_client.async_infer(
                model_name=model_name[i], inputs=inputs[i], outputs=outputs[i]
            )
        )

    for async_request in async_requests[c]:
        # Get the result from the initiated asynchronous inference request.
        # Note the call will block till the server responds.
        result = async_request.get_result()

        print(result.get_response())
```

- 3개의 다른 모델에 100번 request 를 보내는 경우

Nsight 분석 결과

Memcpy from Host to Device (model parameters..)

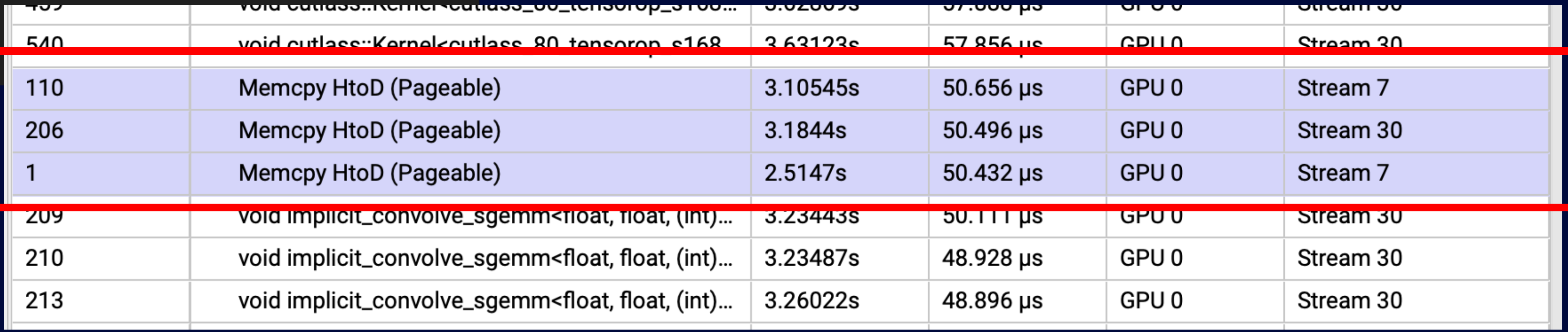
input = […]

for i in range(100):

    Memcpy from Host to Device

    infer(input, …)

    Memcpy from Device to Host



[illegible]	void cutlass::Kernel<cutlass_80_tensorop_s168…	[illegible]	[illegible]	GPU 0	Stream 30
540	void cutlass::Kernel<cutlass_80_tensorop_s168	3.63123s	57.856 µs	GPU 0	Stream 30
110	Memcpy HtoD (Pageable)	3.10545s	50.656 µs	GPU 0	Stream 7
206	Memcpy HtoD (Pageable)	3.1844s	50.496 µs	GPU 0	Stream 30
1	Memcpy HtoD (Pageable)	2.5147s	50.432 µs	GPU 0	Stream 7
209	void implicit_convolve_sgemm<float, float, (int)…	3.23443s	50.111 µs	GPU 0	Stream 30
210	void implicit_convolve_sgemm<float, float, (int)…	3.23487s	48.928 µs	GPU 0	Stream 30
213	void implicit_convolve_sgemm<float, float, (int)…	3.26022s	48.896 µs	GPU 0	Stream 30

1. Triton

Multiple Request - 개선여지?

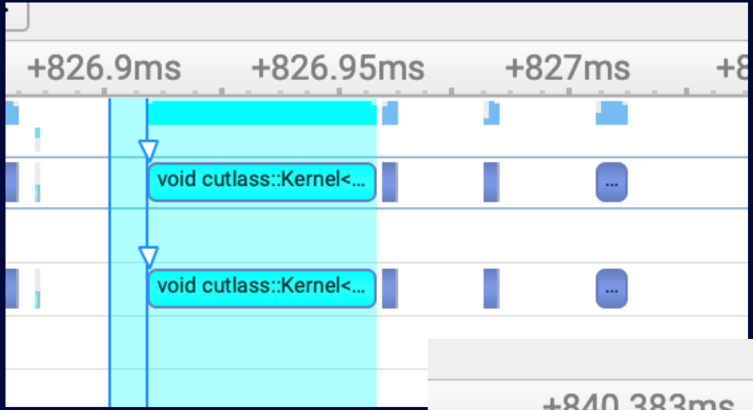
```
input = […]
Memcpy from Host to Device
for i in range(100):
        infer(input, …)
        Memcpy from Device to Host
```

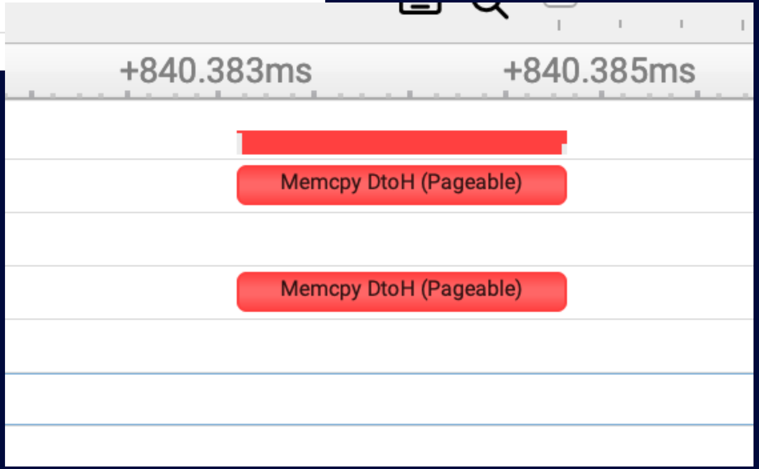
1. Triton

Multiple Request - Synchronization

- 3개의 모델에 100번 infer request 를 보냈을 때

(M1: Stable-Diffusion, M2 : Bert-base-uncased(110M), M3 : VIT-base-patch

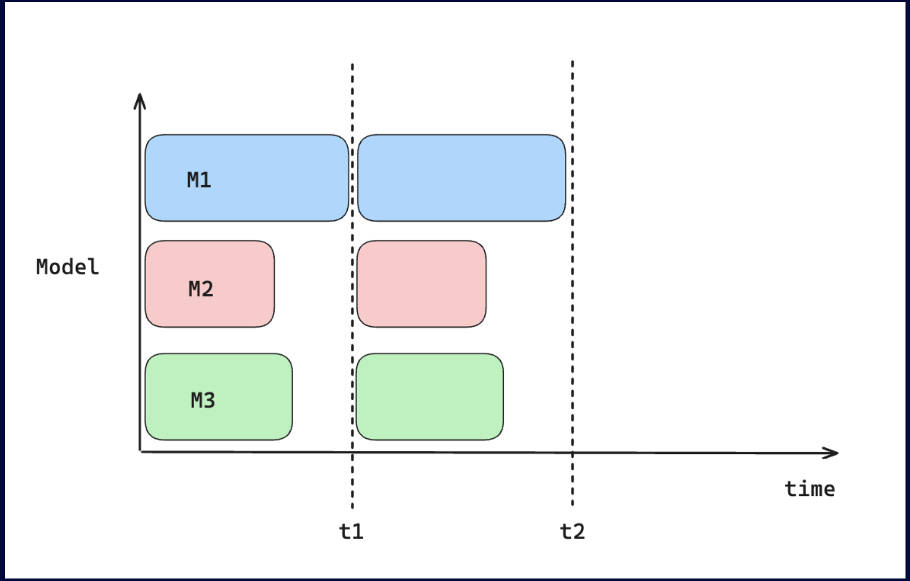




가장 긴 모델의

연산이 끝나고 나서야

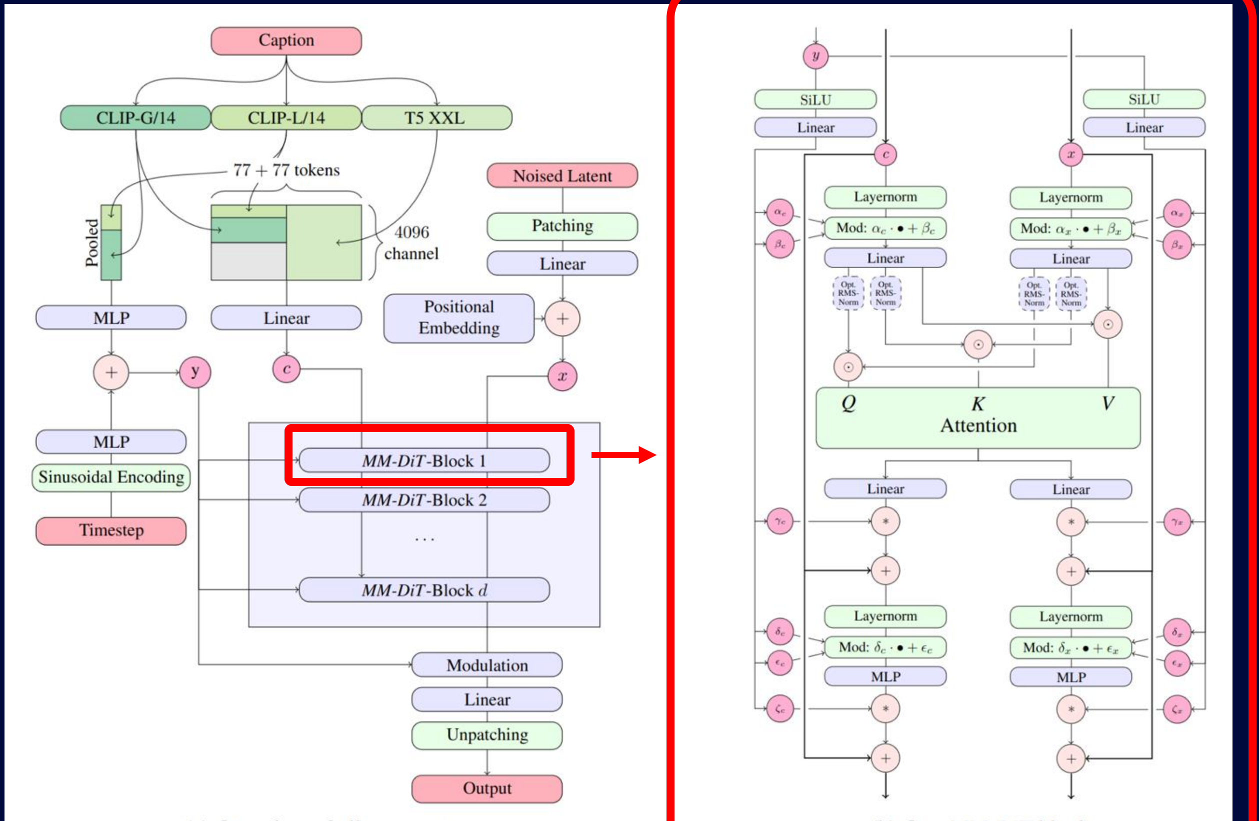
Memcpy(DtoH) 진행



2. Nsight local vs. Triton

사용한 모델

stabilityai / stable-diffusion-3-medium



(a) Overview of all components.

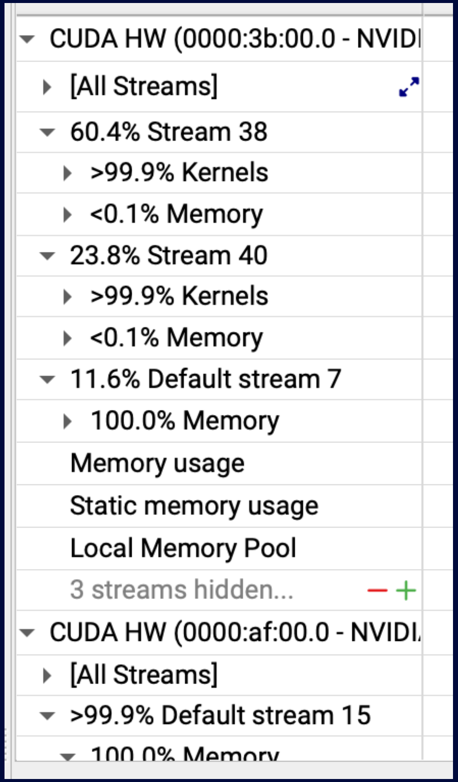
(b) One *MM-DiT* block

- text to image model

2. Nsight local vs. Triton

GPU Utilization

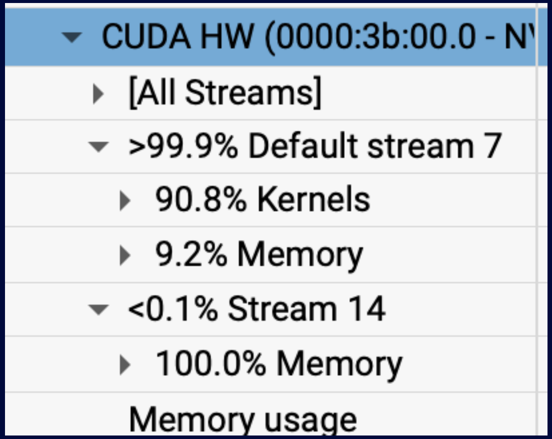
- Triton Server -> 2 streams 사용
- Local



```
▾ CUDA HW (0000:3b:00.0 - NVIDI
  ▸ [All Streams]
  ▾ 60.4% Stream 38
    ▸ >99.9% Kernels
    ▸ <0.1% Memory
  ▾ 23.8% Stream 40
    ▸ >99.9% Kernels
    ▸ <0.1% Memory
  ▾ 11.6% Default stream 7
    ▸ 100.0% Memory
  Memory usage
  Static memory usage
  Local Memory Pool
  3 streams hidden...
▾ CUDA HW (0000:af:00.0 - NVIDI
  ▸ [All Streams]
  ▾ >99.9% Default stream 15
    ▾ 100.0% Memory
```

Triton Server Profiling

-> 2개의 Stream이 kernel 을 나누어서 작동



```
▾ CUDA HW (0000:3b:00.0 - N\
  ▸ [All Streams]
  ▾ >99.9% Default stream 7
    ▸ 90.8% Kernels
    ▸ 9.2% Memory
  ▾ <0.1% Stream 14
    ▸ 100.0% Memory
  Memory usage
```

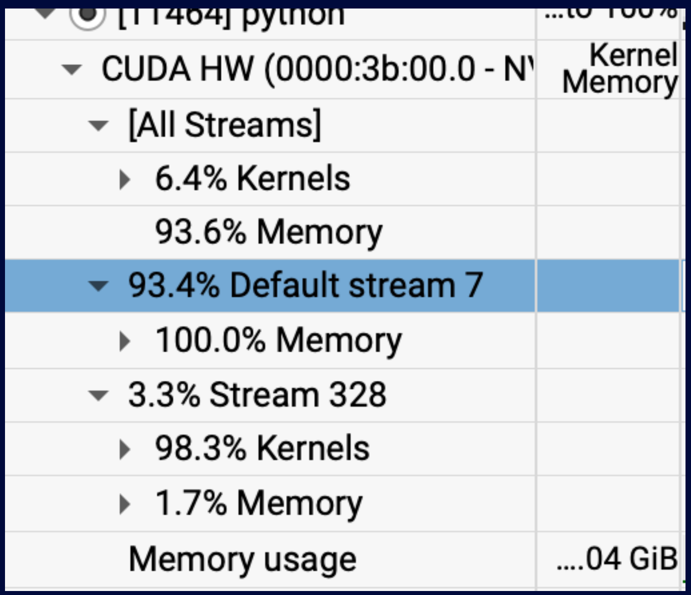
Local Profiling

-> Default stream 에서 커널 + 메모리 복사

2. Nsight local vs. Triton

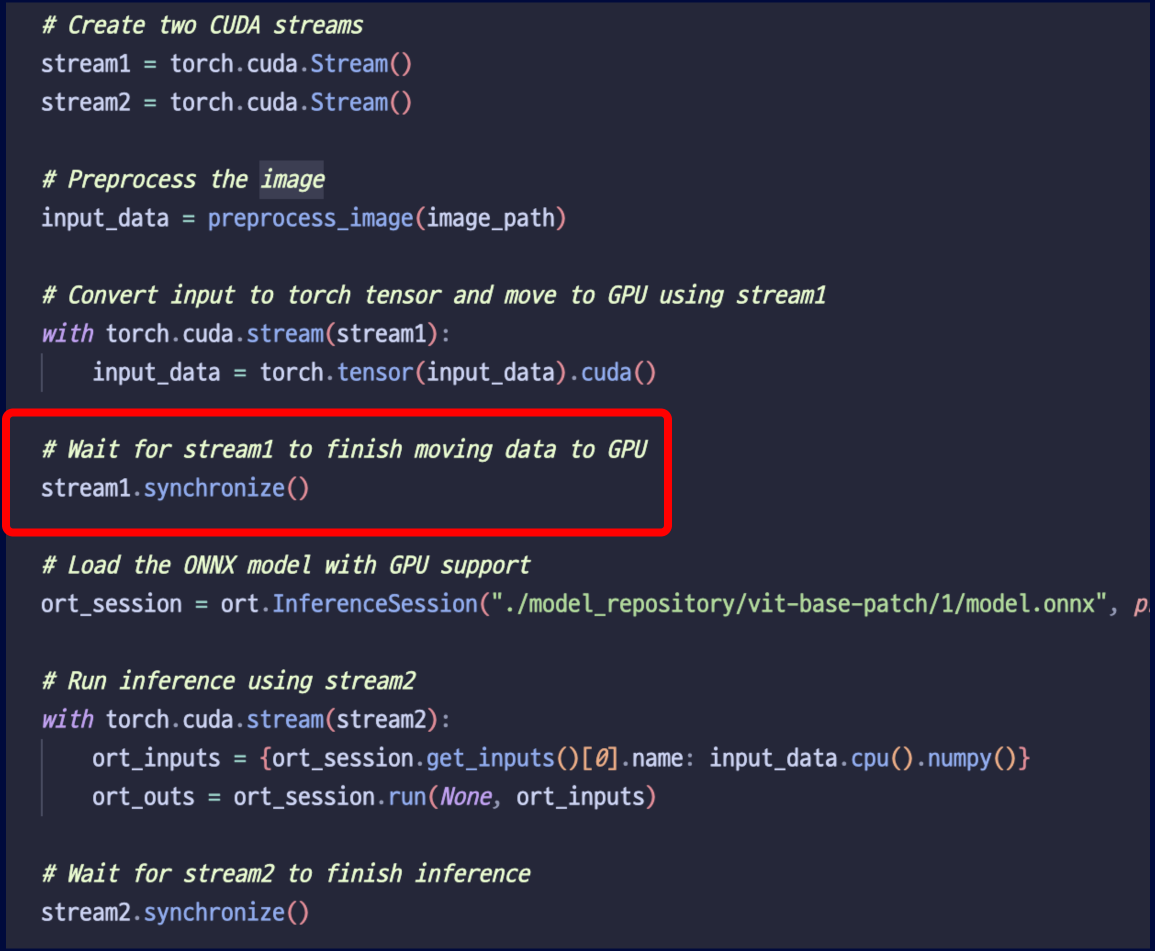
GPU Utilization

- Local -> 2 stream to 2 device



```
[11464] python                        ...to 100%
  CUDA HW (0000:3b:00.0 - N\          Kernel
                                      Memory
    [All Streams]
      6.4% Kernels
      93.6% Memory
    93.4% Default stream 7
      100.0% Memory
    3.3% Stream 328
      98.3% Kernels
      1.7% Memory
    Memory usage                      ....04 GiB
```

Local Profiling

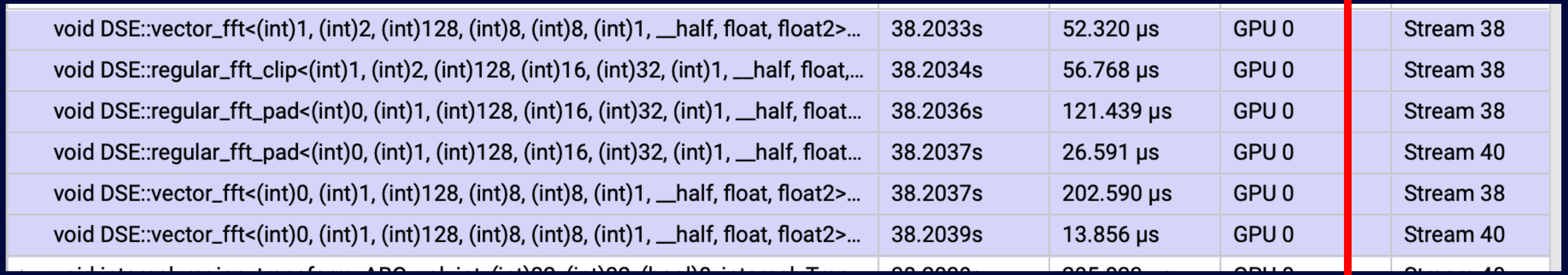


```python
# Create two CUDA streams
stream1 = torch.cuda.Stream()
stream2 = torch.cuda.Stream()

# Preprocess the image
input_data = preprocess_image(image_path)

# Convert input to torch tensor and move to GPU using stream1
with torch.cuda.stream(stream1):
    input_data = torch.tensor(input_data).cuda()

# Wait for stream1 to finish moving data to GPU
stream1.synchronize()

# Load the ONNX model with GPU support
ort_session = ort.InferenceSession("./model_repository/vit-base-patch/1/model.onnx", p

# Run inference using stream2
with torch.cuda.stream(stream2):
    ort_inputs = {ort_session.get_inputs()[0].name: input_data.cpu().numpy()}
    ort_outs = ort_session.run(None, ort_inputs)

# Wait for stream2 to finish inference
stream2.synchronize()
```

2. Nsight local vs. Triton

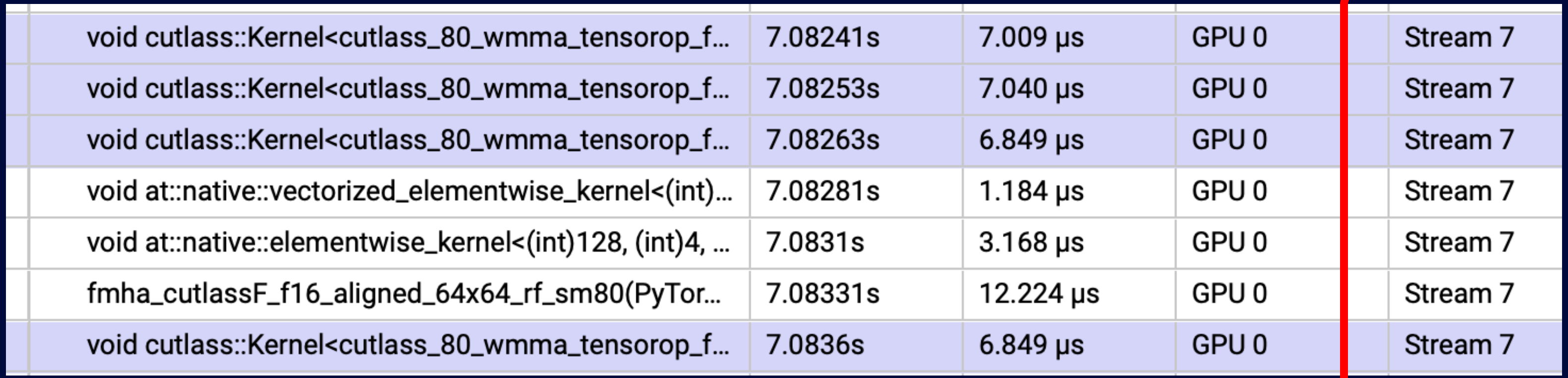
Kernel Execution Patterns

- regular_fft_pad (Fast Fourier Transform (FFT) process - 커널 연산 중 시간이 가장 오래 걸리는 연산)



void DSE::vector_fft<(int)1, (int)2, (int)128, (int)8, (int)8, (int)1, __half, float, float2>...	38.2033s	52.320 µs	GPU 0		Stream 38
void DSE::regular_fft_clip<(int)1, (int)2, (int)128, (int)16, (int)32, (int)1, __half, float,...	38.2034s	56.768 µs	GPU 0		Stream 38
void DSE::regular_fft_pad<(int)0, (int)1, (int)128, (int)16, (int)32, (int)1, __half, float...	38.2036s	121.439 µs	GPU 0		Stream 38
void DSE::regular_fft_pad<(int)0, (int)1, (int)128, (int)16, (int)32, (int)1, __half, float...	38.2037s	26.591 µs	GPU 0		Stream 40
void DSE::vector_fft<(int)0, (int)1, (int)128, (int)8, (int)8, (int)1, __half, float, float2>...	38.2037s	202.590 µs	GPU 0		Stream 38
void DSE::vector_fft<(int)0, (int)1, (int)128, (int)8, (int)8, (int)1, __half, float, float2>...	38.2039s	13.856 µs	GPU 0		Stream 40
[illegible]	[illegible]	[illegible]	[illegible]		[illegible]

Triton : 38, 40 stream 에서 concurrent 하게 계산



void cutlass::Kernel<cutlass_80_wmma_tensorop_f...	7.08241s	7.009 µs	GPU 0		Stream 7
void cutlass::Kernel<cutlass_80_wmma_tensorop_f...	7.08253s	7.040 µs	GPU 0		Stream 7
void cutlass::Kernel<cutlass_80_wmma_tensorop_f...	7.08263s	6.849 µs	GPU 0		Stream 7
void at::native::vectorized_elementwise_kernel<(int)...	7.08281s	1.184 µs	GPU 0		Stream 7
void at::native::elementwise_kernel<(int)128, (int)4, ...	7.0831s	3.168 µs	GPU 0		Stream 7
fmha_cutlassF_f16_aligned_64x64_rf_sm80(PyTor...	7.08331s	12.224 µs	GPU 0		Stream 7
void cutlass::Kernel<cutlass_80_wmma_tensorop_f...	7.0836s	6.849 µs	GPU 0		Stream 7

Local : 하나의 스트림에서만 계산 (동기화 문제..)

3. Plan

- Triton Server code Build + (오류 해결하기)
- 앞의 문제 관련 코드 찾고 고쳐서 실행해보기