Triton

▼ Model config

▼ Quy ước đặt tên:

tên trong tệp config phải phản ánh được đối số truyền vào model

<name>__<index>: Ví dụ: INPUT_0, INPUT_1

▼ reshape

thay đổi shape của input/output.

Nhược điểm: số lượng các số là cố định

Ví dụ: truyền vào 2×3 thì nó chuyển thành 6×1, 3×2,... chứ không thể chuyển thành 2×4

▼ is_shape_tensor

Cho phép truyền vào kích thước động và shape của tensor.

tiết kiệm bộ nhớ và có phép tính toán nhanh hơn.

Tại sao không dùng dim [-1]?

- dim -1 chỉ xác định chiều nào của tensor là động thôi
- is_shape_tensor truyền cả thông tin shape của vector

Tại sao dùng is_shape_tensor vẫn phải xác định dim?

- is_shape_tensor chỉ hỗ trợ trong lúc chạy mô hình. Có thêm shape giúp model dễ dàng hiểu hơn. Thay vì xử lý tổng quát như dim -1

Chỉ hỗ trợ TensorRT

▼ *preserve_ordering trong dynamic_batching:*

trả về output theo đúng thứ tự input nhận vào. Tức là request A đến trước request B thì response A phải được trả về trước.

Sử dụng khi nào? khi cần trả về đúng thứ tự. mô hình theo thời gian hoặc logic có tính thứ tự.

Khi nào output trả không đúng thứ tự?

- Có thể bị chia thành nhiều thread khác nhau để cho các phiên bản khác của model tính toán ⇒ nếu bật dù request B đến sau, xử lý xong rồi nhưng vẫn phải đợi request A xử lý ⇒ trả về a→b

▼ response_cache:

lưu lại cache để truy xuất cho lần sau ⇒ cải thiện hiệu suất.

Tuy nhiên chỉ hỗ trợ 2 loại default và redis.

Phải config ở model và phải có flag --cache-config khi run. Nếu thiếu thì sẽ không chạy.

▼ allow_ragged_batch

Không cần phải thêm pad. Nối lại thành tensor 1 chiều

> When the client sends 3 requests of shapes [1, 3], [1, 4], [1, 5]. To exploit dynamic batching, the straight-forward way to implement this model would expect INPUT shape [-1, -1] and assume that all inputs were padded to same length so that all requests become shape [1, 5] and thus Triton can batch and send them to the model as a single [3, 5] tensor.

> With triton ragged batching, the model will be implemented to expect INPUT shape [-1] and an additional batch input, INDEX, shape [-1] which the model should use to interpret the batch elements in INPUT. For such model, the client requests don’t need to

> be padded and they can be sent as they are (with shapes [1, 3], [1, 4], [1, 5]). The backends discussed above will batch the input into a tensor of shape [12] which contains the 3 + 4 + 5 concatenation of the requests. Triton also creates the batch input tensor of shape [3] with value [3, 7, 12] which gives the offset into the input tensor where each batch element ends.

▼ Metric

```
curl localhost:8002/metrics
```

nv_inference_request_success	
nv_inference_request_failure	
nv_inference_count	Số lượng yêu cầu thực thi trên model. (Một batch n phần tử thì tính là n, Không tính trong cache)
nv_inference_exec_count	Số lượng batch thực thi
nv_inference_pending_request_count	Số lượng infer đang chờ
nv_inference_request_duration_us	Tổng thời gian xử lý yêu cầu
nv_inference_queue_duration_us	Tổng thời gian yêu cầu chờ trong hàng đợi lập lịch
nv_inference_compute_input_duration_us	Thời gian khi 1 yêu cầu gửi từ client, qua các bước xử lý (kiểm tra vector, ...) đến model. Sau đó cộng dồn tổng thời gian đó lại
nv_inference_compute_infer_duration_us	Mô hình tính toán mất bao nhiêu thời gian
nv_inference_compute_output_duration_us	Thời gian từ khi model tính toán xong đến lúc gửi về cho client. Sau đó cộng dồn lại
nv_gpu_power_usage	Công suất tức thời của GPU, tính bằng watt
nv_gpu_power_limit	Giới hạn công suất GPU
nv_energy_consumption	năng lượng tiêu thụ trên GPU

nv_gpu_utilization	Tỷ lệ sử dụng GPU (0,0 - 1,0)
nv_gpu_memory_total_bytes	Bộ nhớ GPU
nv_gpu_memory_used_bytes	Bộ nhớ GPU đã sử dụng, tính bằng byte

▼ Log

▼ log_verbose_level

3 cấp độ: 0, 1, 2

L1: chỉ lưu lại một số thông tin của server, input, output

L2: Ghi chi tiết hơn cấp độ 1

Chi tiết như nào? Config của mỗi model, một số thông tin tiền xử lý như bộ lập lịch, số lượng request, input, output, ...

▼ log_file

▼ log_format: có 2 loại: default và ISO8601. (Một cái tính thời gian chuẩn một cái tính thời gian tương đối)

▼ trace

▼ Model Warmup

Chạy trước mô hình để input ban đầu không phải đợi quá lâu.

▼ Thống kê

```
GET http://localhost:8000/v2/models/wav2vec_py/versions/1/sta
```

```
{
    "model_stats":[
        {
            "name":"wav2vec_py",
```

```
            "version":"1",
            "last_inference":1729175990604,
            "inference_count":12,
            "execution_count":2,
            "inference_stats":
                {
                    "success":{"count":12,"ns":18881524207},
                    "fail":{"count":0,"ns":0},
                    "queue":{"count":12,"ns":138585},
                    "compute_input":{"count":12,"ns":5296336}
                    "compute_infer":{"count":12,"ns":18866711
                    "compute_output":{"count":12,"ns":8127412
                    "cache_hit":{"count":0,"ns":0},
                    "cache_miss":{"count":0,"ns":0}
                },
            "batch_stats":[
                {
                    "batch_size":4,
                    "compute_input":{"count":1,"ns":469864},
                    "compute_infer":{"count":1,"ns":616435628
                    "compute_output":{"count":1,"ns":366943}
                },
                {
                    "batch_size":8,
                    "compute_input":{"count":1,"ns":427110},
                    "compute_infer":{"count":1,"ns":205012108
                    "compute_output":{"count":1,"ns":832455}
                }
            ],
            "memory_usage":[]
        }
    ]
}
```