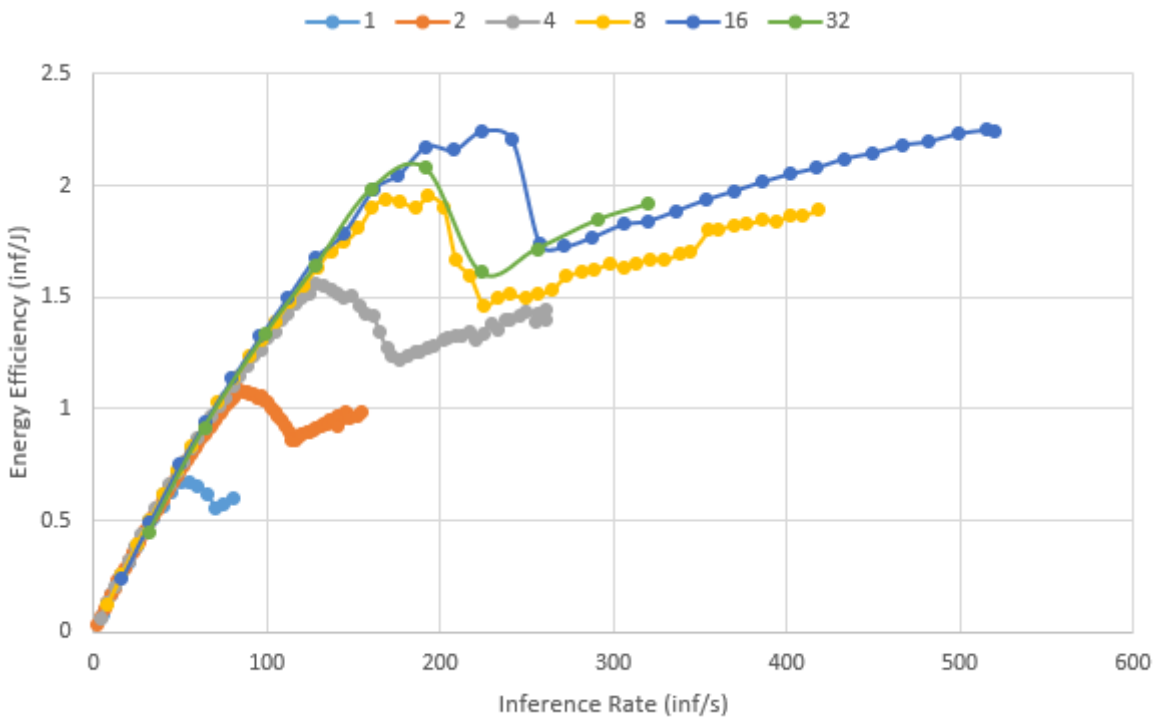
202202

22.02.14

실험환경

- Triton Inference Server 2.7.0
- Model - Inception-v3 FP32, TensorFlow



Batch size를 고정시킨 request를 보낼 때, 처리된 inference의 개수-에너지 효율성을 확인했다. 여기서 inference의 수는 image의 수이며 request의 수에 batch size를 곱한 값이다. (Batch size가 커질수록 compute time이 증가하므로 최대 'request' rate는 낮아지지만 처리된 image의 수는 증가함.)
이러면 그냥 request가 모일 때마다 계속 처리하는게 좋은 것 아닌가? Request rate에 상관없이 max batch size만 잘 결정하면 될 것 같은데.

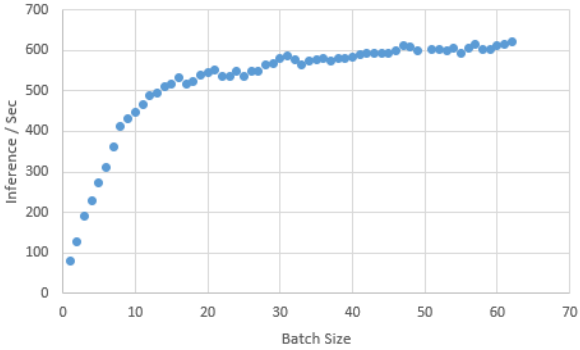
미팅 결과 정리

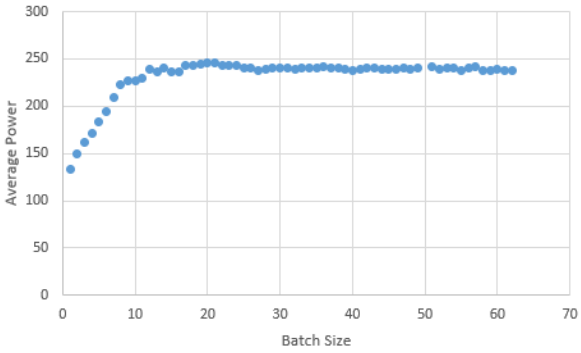
batch size가 증가할 때 energy와 throughput의 상관관계 확인

→ direct하게 상관관계가 있다면 batch size를 조절하는 연재 연구 주제와 같은게 아닌지?

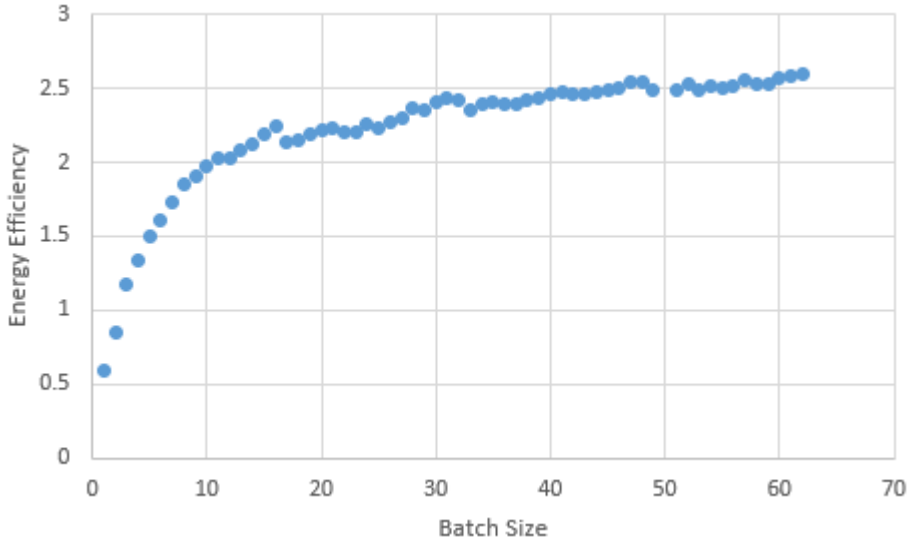
22.02.17

- Model - Inception-v3 FP32, TensorFlow
- Batch size를 점점 증가시키면서 max throughput, average power consumption을 측정

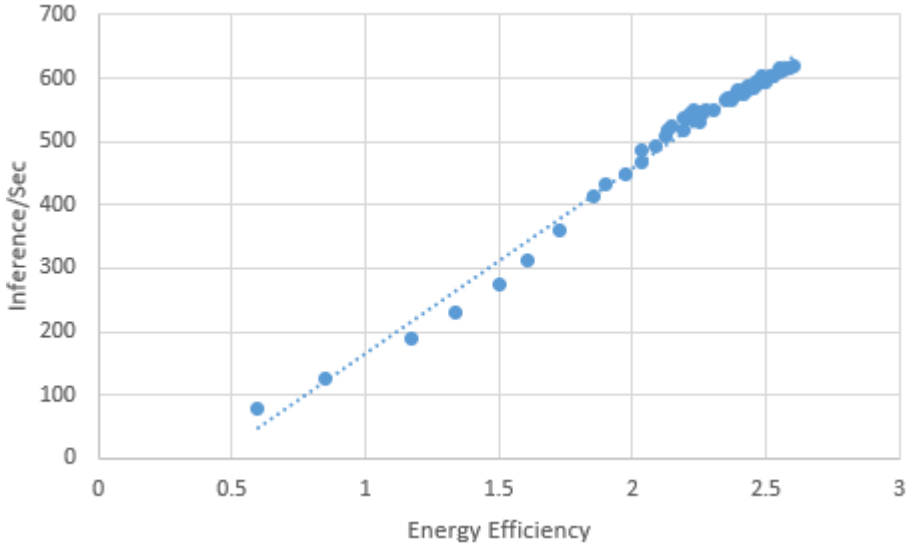




Batch size 증가에 따라서 throughput (inf/s)는 계속 증가한다. 큰 batch size에서는 throughput 증가가 더뎌진다. Power (W)는 특정 batch size를 넘어가면 일정하게 유지된다. 점점 낮아지는 건, 계속 측정하면서 높아진 온도로 쓰로틀링이 걸리기 때문이다. 실험에 사용한 TitanRTX는 power budget이 280W로 설정되어 있다.



위의 metric으로 에너지 효율성을 구해보면 batch size가 크면 클수록 power는 비슷하나 throughput이 조금씩 증가하기 때문에 효율성은 좋아진다.



Throughput과 energy efficiency의 관계를 그려본 결과, 선형관계에 있음을 알 수 있다.

진행중인 연구(연재)

Dynamic batching의 time out을 길게 설정해서 preferred batch size가 모여서 한 번에 처리될 수 있도록 세팅해서 latncy와 throughput을 측정했다. 다음에 request rate를 보고 적절한 batch size를 설정할 수 있는 프레임워크를 만들었고, 이것을 실제로 적용하기 위한 과정을 진행중.

22.02.18

---

1. 실험을 하다가 기존에 짜놓은 triton server에서 inference latency를 300ms까지만 리포팅하도록 되어 있어서 수정했음.
2. 지금은 모든 request의 서버측 latency를 얻기위해 triton에서 stat에다가 업데이트하고 있어 response에 담아서 보낼 수 있는지 확인.
    - Request response는 ProcessRequests(backend) → TRITONBACKEND_ResponseSend → Send → InferResponseComplete -> FinalizeResponse
    - 하지만 request_end_ns는 이 다음 과정에서 ProcessRequests(backend) → TRITONBACKEND_ModelInstanceReportStatistics → ReportStatistics 에 도달했을 때 기록한다.
    - FinalizeResponse를 먼저한 다음에 report. 즉 내가 원하는 기능을 구현하려면 backend도 수정해야함.