

과제 #7

M1522.006700 확장형 고성능 컴퓨팅 (001)

M3239.005400 데이터사이언스를 위한 컴퓨팅 2 (001)

Due: 2024년 12월 15일(일) 23:59:59

1 (200점) DeepSpeed

DeepSpeed의 ZeRO(Zero Redundancy Optimizer)는 대규모 모델 훈련을 효율적으로 수행하기 위한 기술로, GPU 메모리 사용을 최소화하고 성능을 최적화하는 데 중점을 둔다. ZeRO는 데이터 병렬화의 한계를 극복하기 위해 모델 parameter, gradient, 그리고 optimizer state를 여러 GPU에 분산 저장하고 처리한다. 이렇게 하면 각 GPU가 필요한 메모리만 사용하게 되어, 단일 GPU의 메모리 용량 제약을 초과하는 대규모 모델을 훈련할 수 있다.

ZeRO는 크게 3단계로 구성되어 있으며, 각 단계는 더 많은 메모리 최적화를 제공한다. ZeRO-1은 optimizer state를, ZeRO-2는 gradient를, ZeRO-3은 모델 parameter까지 완전히 분산시켜 메모리 사용을 줄인다. ZeRO Offload는 DeepSpeed의 ZeRO(Zero Redundancy Optimizer) 기술을 확장하여, GPU 메모리뿐만 아니라 CPU 메모리를 활용해 대규모 모델 훈련을 가능하게 하는 기술이다.

실습 서버 로그인 노드의 /shpc/skeleton/hw7 디렉토리에 뼈대 코드와 Slurm 작업 스케줄러에 작업을 제출하는 run.sh 스크립트가 제공된다. 스크립트를 통해 GPT2-Large를 ZeRO의 1-3단계로 학습할 수 있으며 실행결과는 sbatch_output directory에 있는 sbatch.out 파일을 통해 확인할 수 있다. 실행예시는 다음과 같다.

```
$ ./run.sh 3
sbatch sbatch_files/zero2.slurm
Submitted batch job 1090471
...

$ cat sbatch_output/zero2-sbatch.out
...
"train_batch_size": 512,
"train_micro_batch_size_per_gpu": 2,
"zero_optimization": {
  "stage": 2,
  "allgather_partitions": true,
  "allgather_bucket_size": 5.000000e+08,
  "reduce_scatter": true,
```

```

        "overlap_comm": true
    },
    "optimizer": {
        "type": "Adam",
        "params": {
            "lr": 0.0001
        }
    }
}
}
Epoch 1, Step 0, Loss: 10.4737548828125
Epoch 1, Step 10, Loss: 9.5732134413428
...

```

보고서에 다음 질문들에 대한 답을 서술하라

- (40점) DeepSpeed의 ZeRO Stage 1, 2, 3, Offload에 대해 설명하라. 각각의 Stage에서 optimizer state, gradient, parameter 값들이 forward, backward, optimizer step 단계에서 어떻게 관리되는지, 그리고 N 개의 GPU가 동일하게 갖고 있는지 또는 나누어 갖고 있는지를 기술해야 한다. 또한, forward, backward, optimizer step 단계에서 어떤 데이터(optimizer state, gradient, parameter)가 통신되고, 이 통신이 어떤 방식(예: collective communication 종류)으로 이루어지는지를 설명하라.
- (40점) ZeRO Stage 1, 2, 3 각각에 맞는 Deepspeed Configuration 값을 확인하고, 각 설정 값이 의미하는 바를 설명하라. 또한, FP16 및 CPU Offloading을 활성화하기 위해 필요한 추가 argument들을 명시하고, 각 설정이 어떤 역할을 하는지 설명하라.
- (40점) 실행 파일(run.sh)의 1번부터 6번까지의 command에 따라 ZeRO Stage에 따른 GPU 메모리 사용량을 확인하라. 각 실행 단계에서 forward, backward, optimizer step 동안의 메모리 사용량을 체크하고, 만약 중간에 문제가 발생한다면 그 이유를 설명하라. torch.cuda.memory_allocated()를 사용하면 GPU 메모리 사용량을 확인할 수 있다.
- (40점) ZeRO Stage에 따른 20 iteration 동안의 학습 시간을 측정하라. 실행 파일(run.sh)의 1번부터 6번까지의 command에 따라 학습시간을 측정하고, 중간에 문제가 발생하면 그 원인을 기술하라.

보고서 (160점) 명시된 질문들에 대한 답으로 평가.

코드 (40점) 실습 서버에서 FP16 및 CPU Offloading을 활성화하기 위한 코드를 deepspeed.training.py에 작성해서 제출.

2 제출 방법

- 과제 제출은 실습 서버에서 이루어진다.
- 보고서는 pdf 형식으로 만들어 report.pdf 이름으로 제출한다. 제출할 report.pdf 파일이 위치한 디렉토리에서 shpc-submit submit hw7 report.pdf 명령을 실행한다.
- 제출할 deepspeed_training.py 파일이 위치한 디렉토리에서 shpc-submit submit hw7 deepspeed_training.py

명령을 실행한다.

- 파일들이 잘 제출되었는지 확인을 위해 `shpc-submit status` 명령을 실행한다.
- 과제 마감 기한이 지난 뒤 다시 제출 명령을 실행하면 마지막 제출시간이 변경되므로 주의할 것.
- 과제 마감 기한이 지난 뒤 파일이 수정된 경우 `grace day` 를 사용한 것으로 간주한다.

3 주의 사항

- 뼈대 코드를 각자의 홈 디렉토리로 복사해 가 작업하도록 한다.
- 실습용 서버에서 과제를 수행하도록 한다. 소스 코드를 제출하는 과제의 경우 실습용 서버에서 작동하지 않으면 점수를 받을 수 없다.
- 보고서는 간략하게 필요한 내용만 적는다.