Final Project

2022-23020 Junyeol Ryu

1. 최적화 기법

1.1. MPI + OpenMP

`run.sh` 스크립트를 수정하여 총 4개의 노드에서 MPI 프로세스를 생성하였고, 각 MPI 프로세스마다 4개씩의 OpenMP 스레드를 만들어 병렬 처리를 할 수 있도록 하였다.

1.2. 모델 파라미터 initialization (Data preloading)

`namegen_initialize`에서 MPI root process가 모델 파라미터를 바이너리 파일로부터 읽은 후, 이를 다른 MPI 프로세스들로 broadcast하였다. 특히, GRU layer의 r, z, n 연산의 경우 kernel fusion을 할 수 있기 때문에 해당 파라미터들은 하나의 tensor로 합쳤다.

1.3. Kernel fusion

`namegen`에서 GRU layer의 r, z, n 연산의 경우 내부 구조가 비슷하기 때문에 kernel fusion을 하여 여러 연산을 한번의 큰 연산으로 합쳐서 처리하였다.

1.4. Batching

여러 name을 batching하여 처리할 수 있도록 관련 tensor들의 shape을 수정하였고, batch된 tensor를 루트 프로세스가 다른 프로세스들로 scatter해서 처리한 후 최종적으로 gather를 하도록 수정하였다.

1.5. Iteration 수정

이름 최대 길이보다 작은 이름이 생성되더라도, 해당 batch에 속한 모든 이름들은 정해진 iteration을 모두 돌도록 `namegen`의 for loop를 수정하였다. 직전 output이 EOS일 경우 해당 Iteration의 연산은 하되, output에는 EOS를 넣는 형태로 코드 수정을 하였다.

1.6. Matmul 최적화

Matrix multiplication의 경우 최대 사용 가능한 GPU device에게 모두 동일한 크기의 일을 분산해 하도록 작성했던 HW6의 Matmul 코드를 porting 하여 사용하였다. Matmul kernel에서는 shared memory를 활용한 tiling 및 vector type(float4)를 사용하는 최적화를 하였다.

1.7. 하이퍼 파라미터 튜닝

가장 좋은 성능을 내도록 batch size를 수정하였다. 실험은 N = 65536으로 고정하고, batch size를 16부터 2배씩 늘려가며 가장 좋은 성능을 내는 batch size를 찾았다.

2. 실험 결과

아래 기술한 최적화 기법들을 순차적으로 적용하였고, 해당 단계에서의 성능 결과를 첨부하였다.

2.1 완전 naive

Elapsed time: 9.463088 seconds
Throughput: 13.526 names/sec

2.2 Matmul porting

Elapsed time: 61.538724 seconds
Throughput: 8.320 names/sec

2.3. Kernel fusion

Elapsed time: 45.985762 seconds
Throughput: 11.134 names/sec

2.4. Batching

Elapsed time: 2.006858 seconds
Throughput: 510.250 names/sec

2.5. 모델 파라미터 initialization(Data preloading)

Elapsed time: 1.878802 seconds
Throughput: 545.028 names/sec