기말 프로젝트

1 Accelerating Text Classifier

1.1 Acheived Performance

```
Model : Classifier
====================================
Number of inputs : 8192
Validation : OFF
------------------------------------
Loading inputs ... DONE
Initializing classifier ... DONE
Classifying 8192 articles ... (a04|rank=0) Start classifier: 0.000000
(a06|rank=2) Start classifier: 0.000000
(a06|rank=2) Init mem: 0.000017
(a06|rank=2) Init CE: 0.000034
(a05|rank=1) Start classifier: 0.000000
(a05|rank=1) Init mem: 0.000019
(a05|rank=1) Init CE: 0.000033
(a06|rank=2) Scatter input: 0.000215
(a04|rank=0) Compute engines initialized: 0.000197
(a04|rank=0) Scatter input: 0.000217
(a07|rank=3) Start classifier: 0.000000
(a07|rank=3) Init mem: 0.000017
(a07|rank=3) Init CE: 0.000032
(a07|rank=3) Scatter input: 0.000206
(a05|rank=1) Scatter input: 0.000196
(a04|rank=0) GPU 0 Inference started: 0.000236
(a04|rank=0) (a04|rank=0) (a04|rank=0) GPU 2 Inference started: 0.000305
GPU 3 Inference started: 0.000307
GPU 1 Inference started: 0.000293
(a05|rank=1) GPU 0 Inference started: 0.011271
(a06|rank=2) GPU 0 Inference started: 0.012651
(a07|rank=3) GPU 0 Inference started: 0.013110
(a05|rank=1) GPU 1 Inference started: 0.019089
(a06|rank=2) GPU 1 Inference started: 0.021532
(a07|rank=3) GPU 1 Inference started: 0.021773
(a05|rank=1) GPU 2 Inference started: 0.026873
(a07|rank=3) GPU 2 Inference started: 0.030757
(a06|rank=2) GPU 2 Inference started: 0.030780
(a05|rank=1) GPU 3 Inference started: 0.034620
(a07|rank=3) GPU 3 Inference started: 0.039857
(a06|rank=2) GPU 3 Inference started: 0.039902
(a04|rank=0) Scatter input done: 0.098183
(a05|rank=1) Scatter input done: 0.268357
(a07|rank=3) Scatter input done: 0.273624
(a06|rank=2) Scatter input done: 0.292203
(a04|rank=0) Computation complete: 0.326870
(a05|rank=1) Computation complete: 0.338256
(a05|rank=1) Gathered output: 0.338319
(a05|rank=1) Finish classifier: 0.338321
(a07|rank=3) Computation complete: 0.364581
(a07|rank=3) Gathered output: 0.364737
(a07|rank=3) Finish classifier: 0.364744
(a06|rank=2) Computation complete: 0.366941
(a06|rank=2) Gathered output: 0.367084
(a06|rank=2) Finish classifier: 0.367092
(a06|rank=2) [LATENCY] Compute: 0.366907
(a07|rank=3) [LATENCY] Compute: 0.364549
(a04|rank=0) Gathered output: 0.367086
(a04|rank=0) Finish classifier: 0.367093
DONE
====================================
Elapsed time : 0.367244 s
Throughput   : 22306.697053 input(s)/sec
(a04|rank=0) [LATENCY] Compute: 0.326673
(a05|rank=1) [LATENCY] Compute: 0.338222
```

Figure 1: The best record of my implementation.

NODE	00(root)	01	02	03
Classifier 시작	0.000000	0.000000	0.000000	0.000000
메모리 초기화	0.000000	0.000016	0.000017	0.000017
Compute Engine 초기화	0.000197	0.000032	0.000034	0.000032
Input 분산 시작	0.000217	0.000215	0.000215	0.000206
GPU0 연산 시작	0.000236	0.005239	0.012651	0.013110
GPU1 연산 시작	0.000305	0.017486	0.021532	0.021773
GPU2 연산 시작	0.000307	0.029747	0.030780	0.030757
GPU3 연산 시작	0.000293	0.041901	0.039902	0.039857
Input 분산 종료	0.098183	0.268357	0.292203	0.273624
Compute Engine 종료	0.326870	0.338256	0.366941	0.364581
Output을 root에 전달	0.367086	0.338319	0.367084	0.364737
Classifier 종료	0.367093	0.338321	0.367092	0.364744

Table 1: Latency breakdown of my implementation, in seconds.

총 8192개의 input을 4개 노드에서 연산하도록 하여 최고 **22306.697053 input(s)/sec**의 성능을 달성하였다[Fig. 1].

1.2 Implementation

Root 노드는 MPI를 이용하여 나머지 노드에 input을 분배한다. 각 노드는 여러 input을 batch로 묶어 처리하며, classifier의 모든 연산을 GPU에서 수행하고 최종 결과만을 CPU로 전송한다. 각종 최적화 기법들을 사용하여 연산에 걸리는 시간을 매우 줄일 수 있었으며, 이에 MPI로 input을 분산하는 과정을 fine-grained하게 쪼개어 compute와 interleave되도록 하였다. 또한 Host와 Device간의 메모리 접근으로 인한 latency를 최소화하기 위해 event를 이용하여 메모리 접근과 커널의 수행을 overlap하였다.

이러한 fine-grained interleaving으로 인해 classifier의 성능은 slurm에 의해 배정되는 노드의 종류나 그 사이의 네트워크 topology등의 요인으로 **측정하는 순간의 서버 클러스터의 네트워크 환경에 따라 약간의 편차가 있을 수 있다.** 하지만 대부분의 경우 20000 input(s)/sec 이상의 성능을 얻을 수 있었으며, 몇 번의 반복을 통해 얻은 최고 성능은 22306.697053 input(s)/sec이었다. **최종 성능 채점시에 이와 같은 점이 충분히 고려되어야 한다.**

1.3 Latency Breakdown

자체 구현한 디버그 모드로 실행하여 Tab. 1과 같이 latency를 세부적으로 분석하였다. 단위는 초(sec)이다. 측정시에 전체 소요시간은 **0.367244 sec** 이었으며, throughput은 **22306.697053 input(s)/sec** 이었다.

1.4 Optimization History

다음과 같은 순서로 최적화를 진행하였으며, 그 각 과정에서 얻은 성능을 측정하였다.

1. Baseline: 2.12 input(s)/sec
2. Synchronous offload: 8.33 input(s)/sec
3. Naively batched computation: 7.86 input(s)/sec
4. Naive CUDA conv1d: 12.76 input(s)/sec
5. Replace every conv1d with conv1d_cuda, fuse relu: 165.00 input(s)/sec
6. Use multiple GPUs: 555.00 input(s)/sec
7. Naive CUDA linear: 727.20 input(s)/sec
8. Replace every linear with linear_cuda, fuse relu: 1152.75 input(s)/sec
9. Merged maxpool1d and relu: 1290.74 input(s)/sec

10. conv1d_k3 square blocking: 1505.14 input(s)/sec
11. conv1d_k3 rectangular blocking: 1550.79 input(s)/sec
12. conv1d hyperparameter tuning: 2537.34 input(s)/sec
13. conv1d_k7 rectangular blocking: 3013.50 input(s)/sec
14. Batched processing: 3501.90 input(s)/sec
15. linear rectangular: 3644.37 input(s)/sec
16. conv1d_k3, conv1d_k7 avoid bank conflict: 3753.42 input(s)/sec
17. Naive linear normalization: 4241.36 input(s)/sec
18. Naive maxpool1d: 5266.67 input(s)/sec
19. Memory cleanup: 5865.32 input(s)/sec
20. No more Tensor type: 6175.81 input(s)/sec
21. Scatter into Scatterv: 5924.65 input(s)/sec
22. Networking & offloading interleaved: 8587.53 input(s)/sec
23. Fine-grained interleaving: 9303.47 input(s)/sec
24. Asynchronous MPI: 13077.40 input(s)/sec
25. Fine-grained layernorm_cuda: 15872.02 input(s)/sec
26. Split gpu stream into mem and compute: 16359.99 input(s)/sec
27. Avoid bank conflict + miscellaneous skills: 17077.22 input(s)/sec
28. layernorm_cuda vectorized mem access: 19267.45 input(s)/sec
29. H2D & D2H access interleaved properly: **22306.697053** input(s)/sec