OpenCL GEMM Optimization

과제 코드 오류 교정 내용 정리

2021-01 인공지능 플랫폼 최적화

HW/SW Optimization for Machine Learning

박영준

과제 코드 오류 교정 내용 요약

- 교정 사항
  - [HW_3], [HW_4], [HW_5] 의 host program에 input array 초기화가 누락된 부분을 교정
  - [HW_5] SGEMM with vectorization
    - host 코드의 에러 체크 코드 변경
    - kernel 코드의 global_work_item layout을 수정
    - kernel 코드에 vectorize 변수 반복문의 오타를 교정

[HW_3], [HW_4], [HW_5] 공통 수정사항

- opencl_host_HW{3, 4, 5}.c
  - line 130: input B에 대해 초기화하도록 수정

```
109
110     printf("HW3: Naive SGEMM, not uniform dims \n");
111
112     // Create the two input vectors
113     int GEMM_M = 2048;
114     int GEMM_N = 1536;
115     int GEMM_K = 1024;
116
117     float *A = (float *)malloc(sizeof(float) * GEMM_M * GEMM_K);
118     float *B = (float *)malloc(sizeof(float) * GEMM_K * GEMM_N);
119
120     int i, j, k;
121
122     for(i = 0; i < GEMM_M; i++) {
123         for(j = 0; j < GEMM_K; j++) {
124             A[i * GEMM_K + j] = (rand() / (float)RAND_MAX) * (0.5 - 0) + 0.5;
125         }
126     }
127
128     for(i = 0; i < GEMM_K; i++) {
129         for(j = 0; j < GEMM_N; j++) {
130             B[i * GEMM_N + j] = (rand() / (float)RAND_MAX) * (0.5 - 0) + 0.5;
131         }
132     }
133
134     // Load the kernel source code into the array source_str
135     FILE *fp;
136     char *source_str;
137     size_t source_size;
138
139     fp = fopen("matmul_HW3.cl", "r");
140     if (!fp) {
141         fprintf(stderr, "Failed to load kernel.\n");
142         exit(1);
143     }
144     source_str = (char*)malloc(MAX_SOURCE_SIZE);
145     source_size = fread( source_str, 1, MAX_SOURCE_SIZE, fp);
146     fclose( fp );
147
```

[HW_5] 수정사항

- opencl_host_HW5.c
  - line 252~261: 에러 체크 코드 변경
    - cpu-gpu 간 FP 연산 fraction 차이 때문에 발생할 수 있는 미세한 오차로 문제가 생기는 경우는 제외할 수 있도록 수정

```
251
252     int res_count = 0;
253     for(i = 0; i < GEMM_M * GEMM_N; i++) {
254         float cmp_abs = C_ref[i] - C[i] > 0 ? C_ref[i] - C[i] : (C_ref[i] - C[i]) * (float)(-1);
255
256         if( cmp_abs / C_ref[i] >= 0.001f )
257             res_count += 1;
258     }
259
260     float res_check = (res_count / (float)(GEMM_M * GEMM_N)) * 100.0f;
261     printf("Performance: %.9lf sec, result: %s (Over 0.0001 percentage error ratio: %.2f) \n", end_time - start_time, res_check <= 5.0f ? "PASSED" : "FAILED", res_check);
262
```

  - Desired output (FAILED일 경우 실패):

```
cass@cass-gpu-server:~/aiplatform_course_HW/HW_5$ ./opencl_host_HW5.exe
HW5: SGEMM with vectorization
Performance: 0.000097990 sec, result: PASSED (Over 0.0001 percentage error ratio: 0.00)
```

[HW_5] 수정사항

- matmul_HW5.cl
  - line 10~11: global_work_item layout을 이전 과제와 동일하게 수정
  - line 19: vlen 만큼의 vectorize를 위한 for문에서의 에러 수정

```
1  // HW_5
2  __kernel void matmul_HW5(
3          const int M,
4          const int N,
5          const int K,
6          const __global float *A,
7          const __global float *B,
8          __global float *C)
9  {
10     int tidx = get_global_id(0); // i
11     int tidy = get_global_id(1); // j
12
13     int vlen = 4;
14
15     if (tidx < M && tidy < N)
16     {
17         float Csub = 0.0f;
18
19         for(int k = 0; k < K; k += vlen) // k
20         {
21             float /* fill here */
22
23             for (int l = 0; l < vlen; ++l)
24             {
25                 /* fill here */
26             }
27         }
28
29         C[tidx * N + tidy] = Csub;
30     }
31 }
32
```

Thank you!

교정사항에 대한 질문:

dydtmd1991@hanyang.ac.kr

2021-01 인공지능 플랫폼 최적화

HW/SW Optimization for Machine Learning

박영준