COLFAX RESEARCH | SOFTWARE EXPERTISE EVALUATION (SEE)

Batch FFTs in HBM

Download lab file

File list:

- Makefile
- main.cc
- worker.cc*

(* files need to be submitted)

Instructions:

In this lab, you will be working with an application that performs Fast Fourier Transforms (FFTs) on multiple large complex datasets. We will be using Intel's MKL library for computing FFTs:

```
void runFFTs(
      const size_t fft_size, // size of a dataset
      const size_t num_fft,  // number of datasets
      MKL_Complex8 *data,    // Input data
      DFTI_DESCRIPTOR_HANDLE *fftHandle // for MKL
     ) {
  for(size_t i = 0; i < num_fft; i++) {
    // Intel MKL FFT
    DftiComputeForward (*fftHandle,
                        &data[i*fft_size]);
  }
}
```

Note that the Intel MKL FFT has multi-threading and vectorization built-in and pre-optimized.

FFTs are bandwidth-bound, and will benefit greatly from the use of High

Bandwidth Memory (HBM) on-board Intel Xeon Phi processor (often referred to as MCDRAM). Unfortunately in the case of this application, we cannot put the entire input data into HBM because the required application memory is greater than the size of the available HBM (16GB). Therefore, the input data must be stored in the regular DDR4 system memory.

However, for large FFT sizes such as is the case here, the difference in performance between using HBM and using regular memory (DDR4) can be enough that you benefit from copying the dataset from DDR4 to a scratch buffer on HBM, doing the FFT in the scratch buffer, and copying the results back. The scratch buffer can be created using either memkind library or hbwmalloc library. Modify "worker.cc" and implement this strategy.

Hints:

We recommend using the hbwmalloc library, which is a higher level wrapper around the memkind library. Both the hbwmalloc library and memkind libraries have Linux *man* pages. If you have access to the Colfax cluster you can view them with the following qsub commands.

```
% echo "man -P cat hbwmalloc" | qsub -l nodes=1:knl
% echo "man -P cat memkind" | qsub -l nodes=1:knl
```

The following code is for creating an 4096-byte aligned buffer for MKL_Complex8 data with the hbwmalloc library.

```
const long buff_size = 1000;
MKL_Complex8 *buff;
hbw_posix_memalign((void**) &buff, 4096,
                   sizeof(MKL_Complex8)*buff_size);
```

Once you have the buffers set-up, remember that you need multi-threaded copy in order to get maximum bandwidth.

Running app:

The grading script uses the following command to run the application.

```
% KMP_HW_SUBSET=1t ./app
```

Grading:

1. Compile: The code compiles without error (0 points)
2. Verification: The code generates the correct output (0 points)
3. Performance: The application completes in under 6.5 seconds on the Xeon Phi 7210 system (1.0 points)

Last transaction TID: 29144
We have graded your submission, and your grade is shown in the table below

Solution: 选取文件 未选择文件

Resubmit

TID	Grade	Feedback	Submitted file	Submission time
29144	100 *,x	Compilation: PASSED Verification: PASSED Performance: PASSED	Submitted solution	2020/12/17 10:28:53
29143	0	Compilation: PASSED Verification: PASSED Performance: FAILED The computation must be completed under 6.5s. Submitted code completed in 18.682055s	Submitted solution	2020/12/17 10:21:42
29142	0	Compilation: PASSED Verification: FAILED The result did not match the reference result	Submitted solution	2020/12/17 10:10:18
29141	0	Compilation: PASSED Verification: FAILED The result did not match the reference result	Submitted solution	2020/12/17 09:58:54
		Compilation: PASSED		

29140	0	Verification: FAILED The result did not match the reference result	Submitted solution	2020/12/17 09:52:52
29139	0	Compilation: PASSED Verification: FAILED The result did not match the reference result	Submitted solution	2020/12/17 09:49:32
29138	0	Compilation: FAILED make command did not produce an executable	Submitted solution	2020/12/17 09:42:20
29137	0	Compilation: FAILED make command did not produce an executable	Submitted solution	2020/12/17 09:39:17

Legend:

* - your best grade for this lab so far

x - not yet submitted to the course platform. Grade transfer may take up to 5 minutes. To re-check the status, reresh this page

Coursera Learners:

- Coursera only remembers your highest score. If you re-attempt the lab and get a lower score, it will not be reflected in Coursera
- If your best grade does not get reflected in Coursera for more than 10 minutes, please let the system administrators know at labs@colfaxresearch.com