SYCL-Bench Artifact Overview Document

Sohan Lal, Aksel Alpay, Philip Salzmann, Biagio Cosenza,
Alexander Hirsch, Nicolai Stawinoga, Peter Thoman,
Thomas Fahringer, and Vincent Heuveline

1 Getting Started

1. Download hipSYCL (hipSYCL 0.8.1-master(`12406c8c`)) from `https://github.com/illuhad/hipSYCL`. Follow the instructions in the README file to build hipSYCL.

   Note: When installing hipSYCL, make sure that hipSYCL is configured with the cmake variable CLANG_EXECUTABLE_PATH pointing to *clang++* instead of just *clang* binary.

2. Download ComputeCpp 1.3.0 from `https://www.codeplay.com/products/computesuite/computecpp`. and unpack using

   ```
   tar -xvf computecpp-ce-1.3.0-ubuntu.16.04-64bit.tar.gz
   ```

   Run computecpp_info in the bin directory to obtain tool chain info. Detailed instructions on how to get started with ComputeCpp can be found at the following link. As such detailed instructions are not needed for reproducing results.
   `https://developer.codeplay.com/products/computecpp/ce/guides`

3. Download SYCL-Bench from `https://github.com/bcosenza/sycl-bench`.

```
# Follow the below steps to build SYCL-Bench
cd sycl-bench
mkdir build # Recommended to build each configuration in a separate build dir
cd  build

# cmake for hipSYCL CPU build (cmake minimum version required 3.5)
# Install cmake if not available
sudo apt-get install cmake

cmake -DCMAKE_PREFIX_PATH=/path/to/hipSYCL/lib -DHIPSYCL_PLATFORM=cpu
↪  -DhipSYCL_DIR=/path/to cmake/inside hipSYCL/lib/cmake -DSYCL_IMPL=hipSYCL
↪  -DCMAKE_SYCL_FLAGS_INIT="-march=native" /path/to/sycl-bench

# cmake for hipSYCL GPU build
cmake -DCMAKE_PREFIX_PATH=/path/to/hipSYCL/lib -DHIPSYCL_PLATFORM=cuda
↪  -DHIPSYCL_GPU_ARCH=sm_52  -DhipSYCL_DIR=/path/to cmake/inside
↪  hipSYCL/lib/cmake -DSYCL_IMPL=hipSYCL /path/to/sycl-bench

# Please specify HIPSYCL_GPU_ARCH that is suitable for your GPU architecture.
# We use sm_52 to match our GPU architecture (Maxwell).

# cmake for ComputeCpp CPU build
cmake -DComputeCpp_DIR=/path/to/computecpp/dir/containing/bin/directory/
↪  -DSYCL_IMPL=ComputeCpp  -DCMAKE_BUILD_TYPE=Release /path/to/sycl-bench

# cmake for ComputeCpp GPU build
```

```
cmake -DComputeCpp_DIR=/path/to/computecpp/dir/containing bin/
↪  -DSYCL_IMPL=ComputeCpp  -DCMAKE_BUILD_TYPE=Release
↪  -DCOMPUTECPP_BITCODE=ptx64 -DCOMPUTECPP_USER_FLAGS="-fno-addrsig
↪  -no-serial-memop -Wno-sycl-undef-func" /path/to/sycl-bench

make
make install
cd bin
```

4. Running benchmarks

```
# run-suite script is used to run all benchmarks
# It is suggested to use gpu-noverify, cpu-noverify for faster execution
# run-suite will generate results in a csv file that is used for plotting graphs.
# For example
./run-suite  gpu-noverify
```

2 Step-by-Step Instructions to Reproduce Main Results

Note: CPU execution may take several hours. Therefore, it is recommended to run GPU results if you are short on time. Hardware/software used to run our experiments.

Hardware: NVIDIA GTX TITAN X (Maxwell), Intel Xeon CPU E5-2699 v3 with 32 GiB DDR4
Software: Ubuntu 16.04, Linux 4.15, Clang 9.0.1, NVIDIA OpenCL 1.2, Intel OpenCL 2.0, CUDA 10.1

2.1 hipSYCL vs ComputeCpp runtime on NVIDIA TITAN X

To reproduce Figure 3 results presented in Section 4.3 in the paper, please follow the following steps.

1. Build SYCL-Bench using hipSYCL for GPU
2. Run run-suite script

```
./run-suite gpu-noverify
```

3. Build SYCL-Bench using ComputeCpp for GPU
4. Again run run-suite

```
./run-suite gpu-noverify
```

5. run-suite will write results in sycl-bench.csv. Rename to avoid conflicts.
6. Plotting Figure 3

```
cd artifact/scripts # submitted artifact folder
Copy hipSYCL and ComputeCpp GPU CSV files to scripts dir
./process_csvs.py 0 sycl-bench-hipSYCL-GPU.csv # Renamed csv file for hipSYCL GPU
./process_csvs.py 1 sycl-bench-ComputeCpp-GPU.csv # Renamed csv file for ComputeCPP GPU
./runTimehipSYCL-ComputeCpp-Stacked-GPU.py

# Install matplotlib if not available
sudo apt-get install python-matplotlib
```

2.2 hipSYCL and ComputeCpp runtime on Intel Xeon CPU

To reproduce Figure 4 results presented in Section 4.3 in the paper, please follow the following steps. As most of the steps are similar to producing Figure 3, a condensed form of steps is given below.

1. Build SYCL-Bench using hipSYCL for CPU

   ```
   ./run-suite cpu-noverify
   ```

2. Build SYCL-Bench using ComputeCpp for CPU

   ```
   ./run-suite cpu-noverify
   ```

3. run-suite will write results in sycl-bench.csv. Rename to avoid conflicts.

4. Plotting Figure 4

   ```
   cd artifact/scripts # submitted artifact folder
   Copy hipSYCL and ComputeCpp CPU CSV files to scripts dir
   ./process_csvs.py 2 sycl-bench-hipSYCL-CPU.csv # Renamed csv file for hipSYCL CPU
   ./process_csvs.py 3 sycl-bench-ComputeCpp-CPU.csv # Renamed csv file for ComputeCPP CPU
   ./runTimehipSYCL-ComputeCpp-CPU.py
   ```