Manual: Automated Stencil Compilers Comparison

By Gsponer S., Guerrera D. and Maffia A.

In this document I will give a overview over the structure of this little framework and how to use it.

I. General Structure

The root directory ($BENCHROOT) contains three folders:

```
$BENCHROOT
├── StencilCompilers
│   ├── installAll.sh
│   ├── <CompilerName>
│   │   └── install<CompilerName>.sh
├── wave
│   ├── cleanAll.sh
│   ├── runAll.sh
│   ├── gatheringOutputs.sh
│   ├── <CompilterName>
│   │   ├── run
│   │   ├── output
│   │   └── src
├── manual
```

A. StencilCompilers

This folder contains all the compilers. The shell script *installAll.sh* call the corresponding *installCompiler.sh* in the sub folders. This scripts downloads the source, extract it and compile the files.

B. wave

Every stencil compiler needs it own prepared code for the comparison. The wave directory contains prepared code for Patus, Pochoir, Pluto, Halide and a naive approach. For each compiler the structure in the sub folder is the same and should be self explaining. Additionally to the compiler specific files some scripts are also provided.

- runAll.sh: Compiles the stencil with each compiler and run it for fixed number of threads.

- cleanAll.sh: Cleans up the sub folders from the last compilation.

- geatheringOutputs.sh: Collects the output of the different compilers and arrange them in the result.dat file.

C.   manual

In the manual folder all files for this document are stored.

II.   Usage

First thing we need to do is source the SetupEnv file. This set some environment variable for all the shell scripts like the root directory.

```
$ source SetupEnv
```

If the compilers are not installed yet switch to the *StencilCompilers* folder and run the *installAll.sh* script.

```
$ cd StencilCompilers
$ ./ installAll.sh
```

It is also possible to run the script only for a specific compiler. To achieve this simple give the name of the compiler as argument.
Each compiler comes with its own dependencies. At the end of this document I will provide a not final list for each compiler.
After installing all the compilers we can begin to run the comparison. For this change to the wave folder and execute the *runAll.sh*.

```
$ cd ../ wave
$ ./ runAll.sh
```

This step will take some time since it actually run the wave stencil with all the compiler for different number of threads. Per default this runs the stencil for 1, 2, 4, 8, 16 threads but this can be changed in the *SetupEnv* file.
When the computation is finished the *gatheringOutputs.sh* script can be executed which creates the result.dat file containing all the outputs. This file can afterwards be plotted with the *gnuplot.gp* file.

```
$ ./ gatheringOutputs.sh
$ gnuplot gnuplot.gp —— persist
```

After this step a plot sorted by number of threads shows up.

III.   Dependencies

A.   General

- wget

- Make

B.   Patus

- Java 7 or newer

- Maxima

- gcc

C.   Pochoir

- Intel C++ Compiler version 12.0.0 or later with Cilk Plus extension, up to v 13.0.0

- Glasgow Haskell Compilation System version 6.12.1 or later

- Parsec-2.1.0.1 or later

D.   Pluto

- gmp (GNU multiple precision arithmetic library)

E.   Halide

- llvm 3.2 or higher

- clang version matching to the llvm version