

# First Programming and Optimizing for Performance (POP'16) Competition

## Overview:

HIMENO is the most famous benchmark in whether forecast. The benchmark consists of a kernel which is a solver of Poisson's equation<sup>1</sup> and represents a single iteration of the Jacobi iterative method<sup>2</sup>, using a typical 19-point Laplacian stencil code<sup>3</sup>.

HIMENO includes five standard grid sizes (XS/S/M/L/XL), but in the POP'16 competition we are only considering the large grid size (L).

## Goal

The goal is to accelerate the execution of the function *jacobi* in the ANTAREX target machine using only multicore CPUs (see the characteristics of the machine below). To accelerate the code, one possibility is to use a multithreaded implementation (by using POSIX threads<sup>4</sup> or OpenMP directives<sup>5</sup>).

The function to be accelerated has the following prototype:

```
void jacobi(Matrix* a, Matrix* b, Matrix* c, Matrix* p, Matrix*
bnd, Matrix* wrk1, Matrix* wrk2)
```

And the implementation can be found in the file *jacobi.c*. The file contains two additional functions, which are used to initialize the input data. Since we are targeting a NUMA machine, the way memory is accessed for the first time can impact the performance of the kernel.

The benchmark provides a score at the end of execution. A score of 100 represents the execution time on the ANTAREX machine for that size, when compiled without optimizations. Higher scores represent faster implementations, and the objective is to submit a version that gets the best score when using the L grid size. Please note that while the evaluation will use the L grid size, other sizes will be considered in the case of an unclear winner.

## Example

Below we present the results of executing the code using the L grid size.

```
> himeno.exe L
Now, start the actual measurement process
Wait for a while...
Time : 3.809950e+03 ms
Score : 100
Gosa : 4.882812e-04
Verification successful!
```

<sup>1</sup> [https://en.wikipedia.org/wiki/Poisson%27s\\_equation](https://en.wikipedia.org/wiki/Poisson%27s_equation)

<sup>2</sup> [https://en.wikipedia.org/wiki/Jacobi\\_method](https://en.wikipedia.org/wiki/Jacobi_method)

<sup>3</sup> [https://en.wikipedia.org/wiki/Stencil\\_code](https://en.wikipedia.org/wiki/Stencil_code)

<sup>4</sup> <https://computing.llnl.gov/tutorials/pthreads/>

<sup>5</sup> <http://openmp.org/wp/>

At the end of execution, the program prints the computation time of a single execution of the kernel, as well as the corresponding score. For reference, the value of Gosa (one of the outputs) is also printed. Finally, the benchmark will inform you whether the produced outputs are correct.

## Target machine

The ANTAREX machine represents a supercomputing center node and the properties of the machine relevant to this contest are the ones below (note that in this first edition of the POP competition we are not mapping computations to the GPU).

ANTAREX Machine	
CPU	2x INTEL XEON E5-2630V3 2.4 GHZ 20MB CACHE 8 cores per CPU, and 2 threads per core (total of 16 cores, and 32 threads)
GPU	AMD Radeon R9 Nano
RAM	4x 16GB DDR4-2133MHZ REG ECC MODULE (64GB total, 32GB per NUMA node)

The compiler and relevant libraries that will be used in the machine are the following:

- GCC 5.4.0
- OpenMP 4.0 (GNU libgomp<sup>6</sup>)

The machine has Ubuntu 16.04 LTS installed. Intel® Turbo Boost is disabled, to provide more consistent results (maximum frequency of 2.4GHz).

## How to Evaluate your Modifications:

You can download the packages containing the source code and the expected outputs from the website [specs.fe.up.pt/pop16](https://specs.fe.up.pt/pop16). The source code package contains all the necessary code as well as a *Makefile* (tested in Linux and Windows) which emulates how the application will be compiled after the final submission. You are free to make your modifications to the *jacobi.c* file and change the *Makefile* to add custom flags (use the variable `JACOBI_FLAGS`). While you can manually change other source files and compilation options, this is discouraged as these will not be reflected in the final results. You will only be able to submit the file *jacobi.c* and the compilation flags.

The outputs packages have binary files which are to be extracted to a directory named 'outputs', in the same directory as the source code. If these exist, the program performs an automatic verification of the outputs, so that you can check if you have correct results.

On a Linux machine, you can do the following in order to compile, run and test the application, starting from the source and outputs packages:

```
> mkdir contest_dir
> tar -zxvf himeno_sources.tar.gz -C contest_dir
> tar -zxvf himeno_outputs.tar.gz -C contest_dir
```

---

<sup>6</sup> <https://gcc.gnu.org/onlinedocs/libgomp/>

```
> cd contest_dir
> make run
```

You can use your own machine to develop and test solutions, and submit your *jacobi.c* file and compilation flags to the POP'16 website in order to perform the same evaluation procedure on the ANTAREX machine. While the results are not binding, you can get a reference value on target machine where the final evaluation will be conducted.

To use the website you will need a code that is unique to each team, please ask the people in the organization for this code when you arrive. If we detect abusive use of the website (e.g., intentional denial-of-service attempts), we will make the responsible team code invalid.

## Submission and Final Evaluation:

You should submit your *jacobi.c* file and team code by email, using the address *specs@fe.up.pt* and the subject "POP16", before the 23h of the 19<sup>th</sup> of October. The *jacobi.c* contains a comment section which must be filled with the names of the participants, the gcc flags for your solution, scores in the ANTAREX machine and in your test machine, as well as the characteristics of the test machine. You can also provide additional comments and overall description of your optimization strategy, we would be happy to hear about your ideas.

The final evaluation of the optimizations on the *jacobi* kernel will be performed after the contest is over and the results should be expected within a week. We will be using the score metric and will evaluate the submissions based on their performance when executing with the L grid size. In the case of an unclear winner, we will consider the score using other sizes as well as the overall strategy.

The application code will be compiled under the GNU11 standard (C11 with GNU extensions) and without optimizations, as shown on the provided Makefile. The exception is the submitted *jacobi.c* file, which will be compiled with the options defined by each team.

The code of the top-rated submissions will undergo manual inspection on top of the automatic verification. Teams that submit obfuscated code, or whose Jacobi function does not perform the needed computation shall be disqualified.

## Sponsors:

The POP contest is sponsored by the ANTAREX project (H2020 FET-HPC project with Reference: 671623, <http://www.antarex-project.edu>) and is organized by the SPeCS research group<sup>7</sup> (<http://www.fe.up.pt/~specs/>) and by the IEEE Student Branch of FEUP (<http://nuieee.fe.up.pt/>).



---

<sup>7</sup> SPeCS is a research group located at the DEI (Department of Informatics Engineering) Computing Systems Lab (FEUP, room J204) and associated with INESC TEC.