Profiling an Image Processing Implementation in Adaptive Optics

Nandhana Sakthivel

Exercise 2

Aim

The aim of this exercise is to profile and get a deeper understanding of the given code. Here, we are using a part of Image Processing implementation that is carried out in Adaptive Optics which is written in C. The above mentioned code uses **fftw** library and is profiled using **Valgrind**: where we used **Callgrind** and the call graph is visualised using **gprof2dot**.

Introduction

Profiling is a process of analysing the performance of a given code by measuring the frequency and duration of the function calls which helps us to understand the given code.Computer architects need such process to evaluate how well the programs will perform on new architectures. There so many tools available for profiling. Here we are using valgrind.

Valgrind

Valgrind was originally designed to be a free memory debugging tool for Linux on x86, but has since evolved to become a generic framework for creating dynamic analysis tools such as checkers and profilers. It has several other tools. Here, we are using callgrind to profile the code.

Callgrind

Callgrind is a profiling tool that records the call history among functions in a program's run as a call-graph. By default, the collected data consists of the number of instructions executed, their relationship to source lines, the caller/callee relationship between functions, and the numbers of such calls. Optionally, cache simulation and/or branch prediction (similar to Cachegrind) can produce further information about the runtime behavior of an application.

valgrind –tool=callgrind ./a.out

The output is stored in the file: **callgrind.out.28311**. The report below shows that it collected 1 billion events in order to generate the comprehensive report which is saved in the file callgrind.out.28311 — 28311 is the process id.

```
==28311==
==28311== Events    : Ir
==28311== Collected : 1096597016
==28311==
==28311== I   refs:      1,096,597,016
```

Figure 1: Callgrind

gprof2dot

gprof2dot.py is a Python script to convert the output from many profilers into a dot graph.Here we used the following commands to visualize the output of the callgrind.

gprof2dot -f callgrind -o img.dot callgrind.out.28311
dot -Tpng img.dot -o imgprocess.png

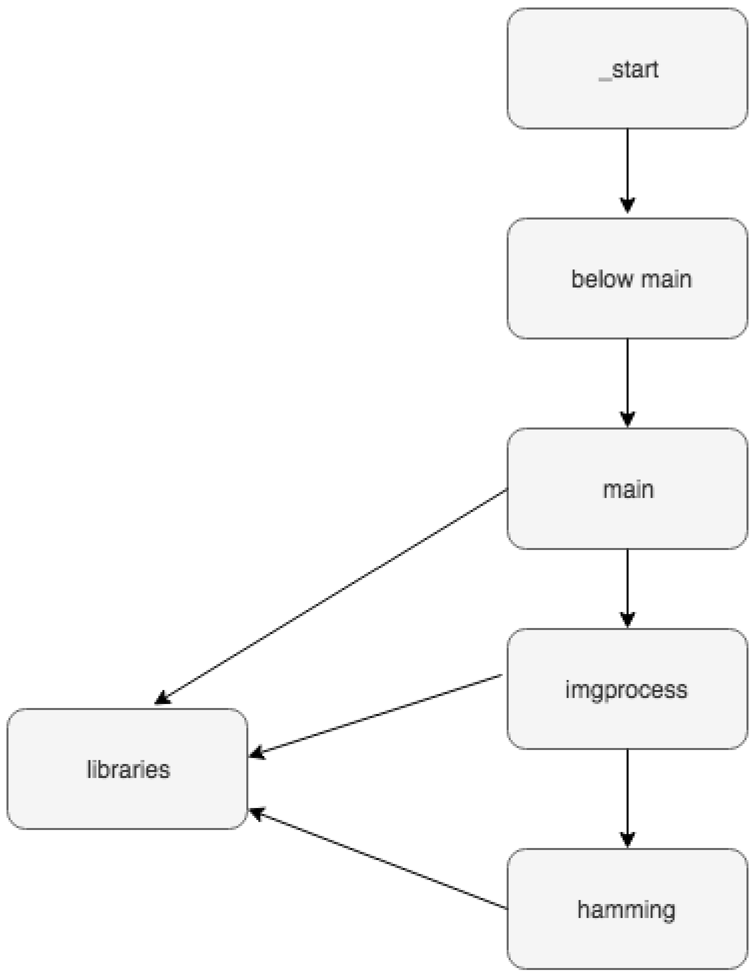

Figure 2: Consolidated Call graph

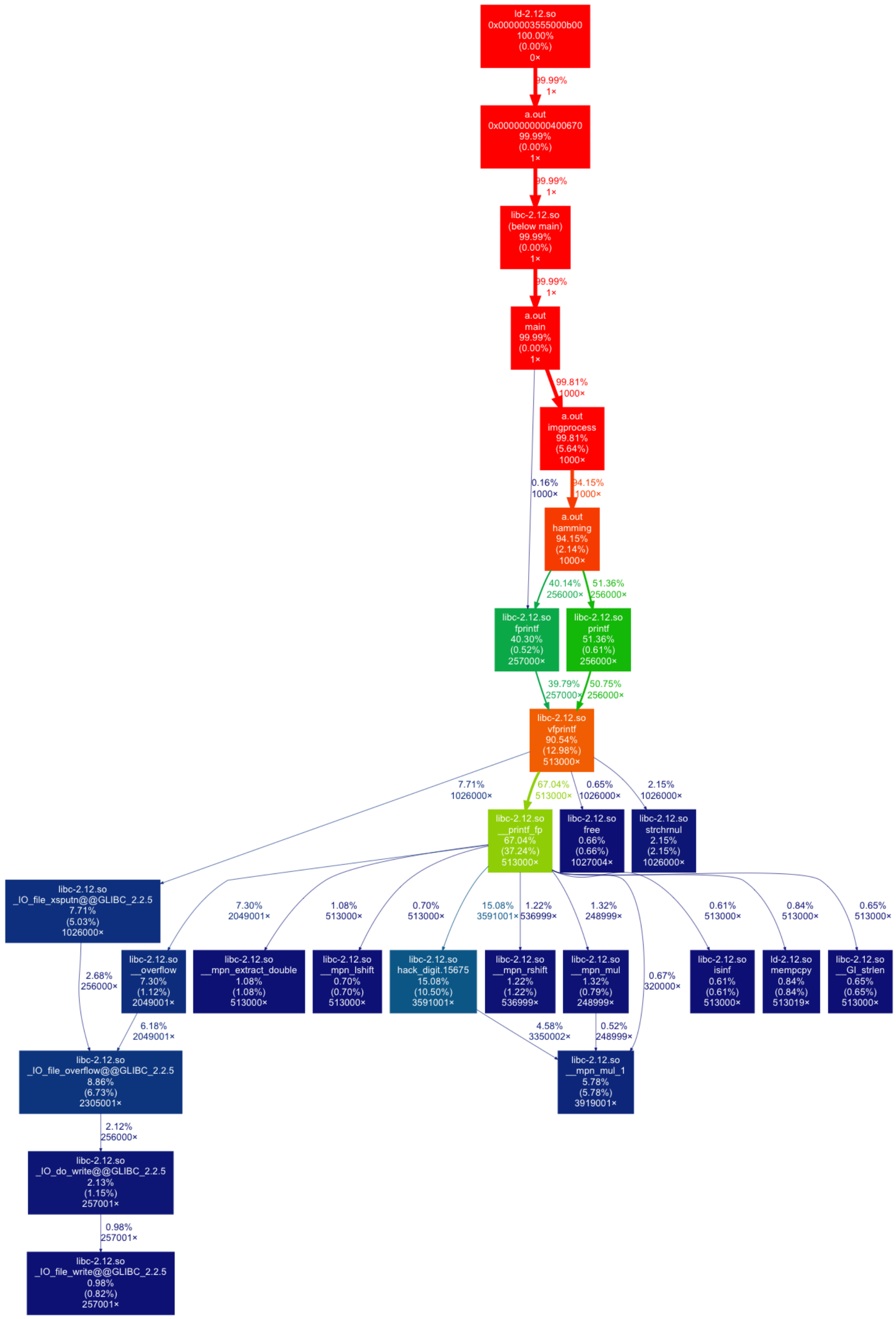

Figure 3: Callgrind Call graph

Conclusion

From the graph we can see that most of the time is spent in the main function where we are allocating 16*16 matrix using the fftw library when compared to the other functions present.