

Parallel Programming 2013/14 - Lab exercise 4

You are invited to a programming challenge. This challenge presents an array of bitmaps, which are represented by integer square matrices of dimensions $64 * 64$. A black pixel within a bitmap is represented by 1 whereas a white pixel is represented by a 0. If a bitmap contains more black pixels than white pixels, the bitmap is classified as a B-bitmap, otherwise it is classified as a W-bitmap. Bitmaps with equal numbers of black and white pixels are also classified as B-bitmaps. The aim of the challenge is to come up with a program that can classify a vector of bitmaps into B-bitmaps and W-bitmaps as fast as possible (maximum throughput), the vector size n is such that: $n = 2^k, k \geq 10$.

The program committee has published the specification of the target computer on which all the solutions will be evaluated:

CPU	Intel Core i5-3570 @ 3.4GHz (Quad core)
RAM	4GB
Other processing units	NVIDIA GeForce GT610

They have also informed the participants that they are free to use a programming environment of their choice.

The challenge consists of three rounds:

- Download the challenge program code:

```
http://www.cs.bham.ac.uk/~hxt/2013/parallel-programming/lab4.zip
```

First, familiarize yourself with the code. The aim is to implement the `soln()` function within `soln_stub.c`, you are not allowed to modify any other source files.

- In the qualifying round, you will compete against two robots. The first robot models a first year undergrad. His code is available at:

```
http://www.cs.bham.ac.uk/~hxt/2013/parallel-programming/soln\_a.c
```

As expected, he has written the most trivial program to accomplish the task. You can compile and run this program with the commands:

```
> gcc soln_a.c main.c
> ./a.out
```

The second robot models a self-taught C “hacker” who has recently taken a university module on concurrent programming. His multi-threaded program is available at:

```
http://www.cs.bham.ac.uk/~hxt/2013/parallel-programming/soln\_b.c
```

You can compile and run this program with the commands:

```
> gcc -lpthread soln_b.c main.c
> ./a.out
```

In order to pass the qualifying round, you must come up with a program that beats both of these programs. If you only manage to beat the first year undergrad, you will receive a note-of-thanks for your participation in the challenge.

- In the second round you will compete against a veteran C programmer. His code is available at:

<http://www.cs.bham.ac.uk/~hxt/2013/parallel-programming/soln.c.c>

He also happens to be a GCC expert, and has informed the program committee to enable aggressive compiler optimizations on his program (-O3). You can compile and run this program with the following commands:

```
> gcc -O3 -lpthread soln.c.c main.c
> ./a.out
```

Fortunately for him, it seems the optimizations do not affect the correctness of the program. You have to beat this guy in order to qualify for the final round. All those participants qualifying from this round will receive official bragging rights on CUDA (engraved on an email, of course).

- In the final round, the programs of all those qualifying will be evaluated against one another (and ranked). The author of the program which reports the best runtime (best of 5 invocations) will be the winner.

Email your program submissions (with build instructions) to the demonstrator by the official dead-line (announced on the module page).