PPAR: GPU lab 1 and 2

Sylvain Collange

March 2018

You will be given a key `pparXX_id_rsa` to your account `pparXX` on parawell, where *XX* is a unique number. To log in, enter

```
ssh -i pparXX_id_rsa pparXX@parawell.irisa.fr
```

where `pparXX_id_rsa` is the complete path to the key. Once logged on your account on paramax, you will start from the template in: `gpu_lab1`.
In this lab assignment, we want to compute the following series:

$$\frac{1}{1} - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \frac{1}{5} \cdots,$$

or equivalently:

$$\sum_{i=0}^{n-1} \frac{(-1)^i}{i+1}$$

for large n.

1 Parallelization

1. Program this computation in sequential C in function `log2_series`. Note that in C, computing $(-1)^i$ can be done by testing whether the value of `i % 2` is 0 or 1.

2. Compare the result obtained when summing the terms by increasing order of indices (from 1 to n) and by decreasing order (from n back to 1). How do you explain your observations?

We will now offload this computation to the GPU. We could implement it with a reduction using n threads, but it would require too much memory. Instead, each thread will compute multiple terms of the series.
3. We want to split the computation to parallelize it on m threads ($m < n$). Give two possible strategies to split the n elements equally between the m threads.

For now, we return one result per thread, and finish the computation on the CPU.
4. Implement memory allocation, copy, and de-allocation for the results.

5. Program the computation in `summation_kernel`. Perform the final computation on the CPU.

6. Compare the result with the CPU-only versions and explain possible differences. Implement the other solution from question 3 and compare its result.

7. Compare the run-times obtained for different number of threads per block and number of blocks. What is the best block size and grid size for your GPU?

2 Reduction

8. We want to reduce the amount of data transferred back. How to adapt the summation algorithm to return one value per block? Implement it.

9. Perform the whole computation on the GPU.