Comparison of Python 3 Single-GPU Parallelization Technologies on Example of Charged Particles Dynamics Simulation Problem

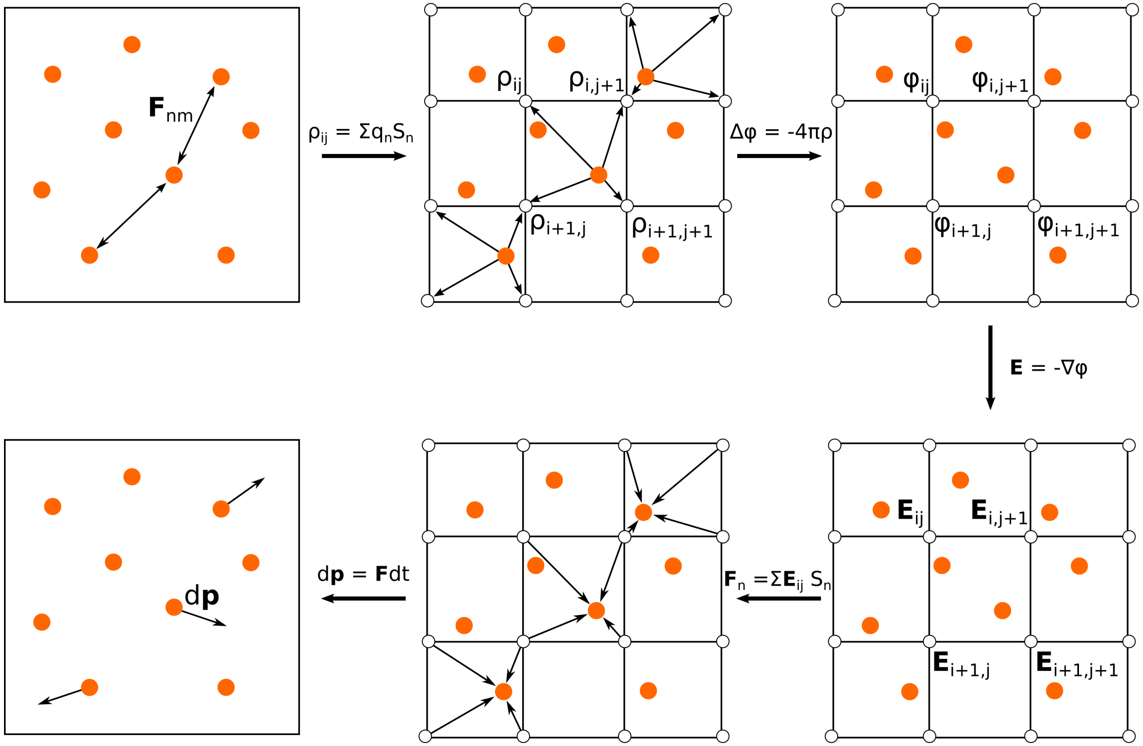
Alexey Boytsov, Ivan Kadochnikov, Maxim Zuev, Andrew Bulychev, Yaroslav Zolotuhin

General motivation

Ion source simulation

Concrete motivation

Particle-in-cell simulation



Python

Fast development

Slow computing

GPU

Slow development

Fast computing

Libraries

Python development

GPU computing

Candidate libraries

Numba

- CUDA (Nvidia)
- Python decorators

```
@jit
def sum2d(arr):
    ...
```

PyCUDA

- CUDA (Nvidia)
- Compiled string of CUDA code

```
mod = SourceModule("__global__ void region_update_vels_from_bin_interaction(float reg, float reg_5, …
```

PyOpenCL

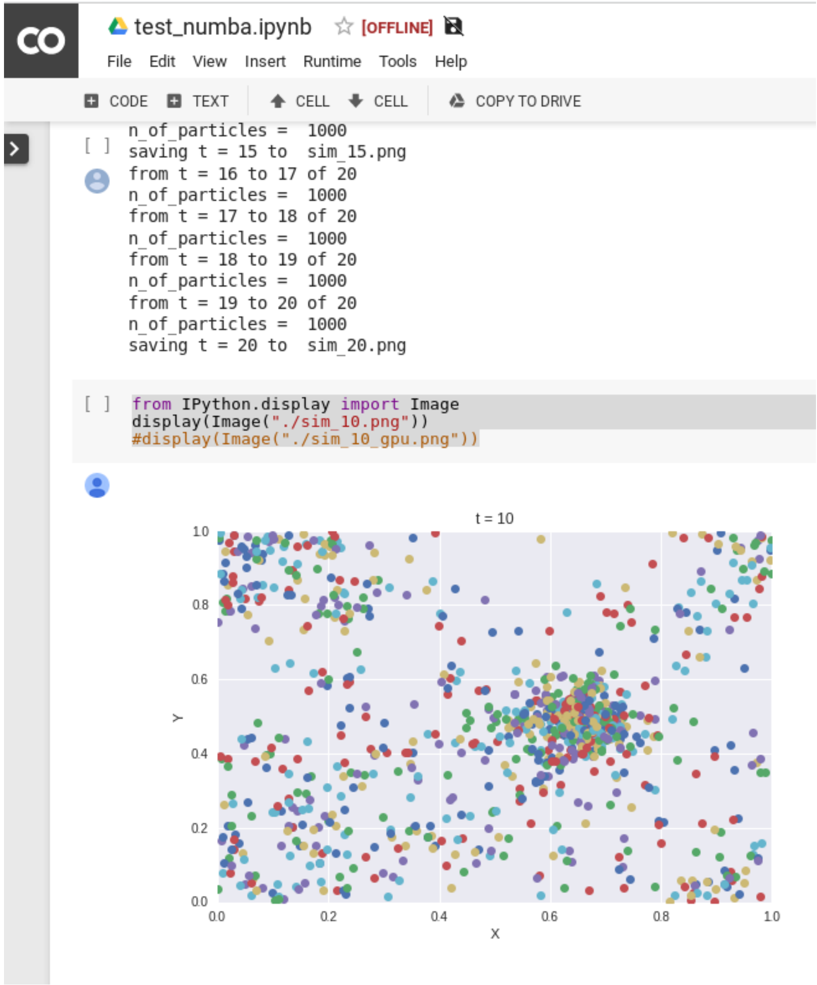
- OpenCL
  - on CUDA
- Compiled string of OpenCL code

```
program = cl.Program(reg["ctx"], """
    __kernel void update_position(
    __global const unsigned int *id,
    ...
```

Testing platform

Google Colaboratory

- NVidia Tesla K80
- Shared IPython notebooks
- Easy collaboration
- Common environment



```
n_of_particles =  1000
saving t = 15 to  sim_15.png
from t = 16 to 17 of 20
n_of_particles =  1000
from t = 17 to 18 of 20
n_of_particles =  1000
from t = 18 to 19 of 20
n_of_particles =  1000
from t = 19 to 20 of 20
n_of_particles =  1000
saving t = 20 to  sim_20.png
```

```
from IPython.display import Image
display(Image("./sim_10.png"))
#display(Image("./sim_10_gpu.png"))
```

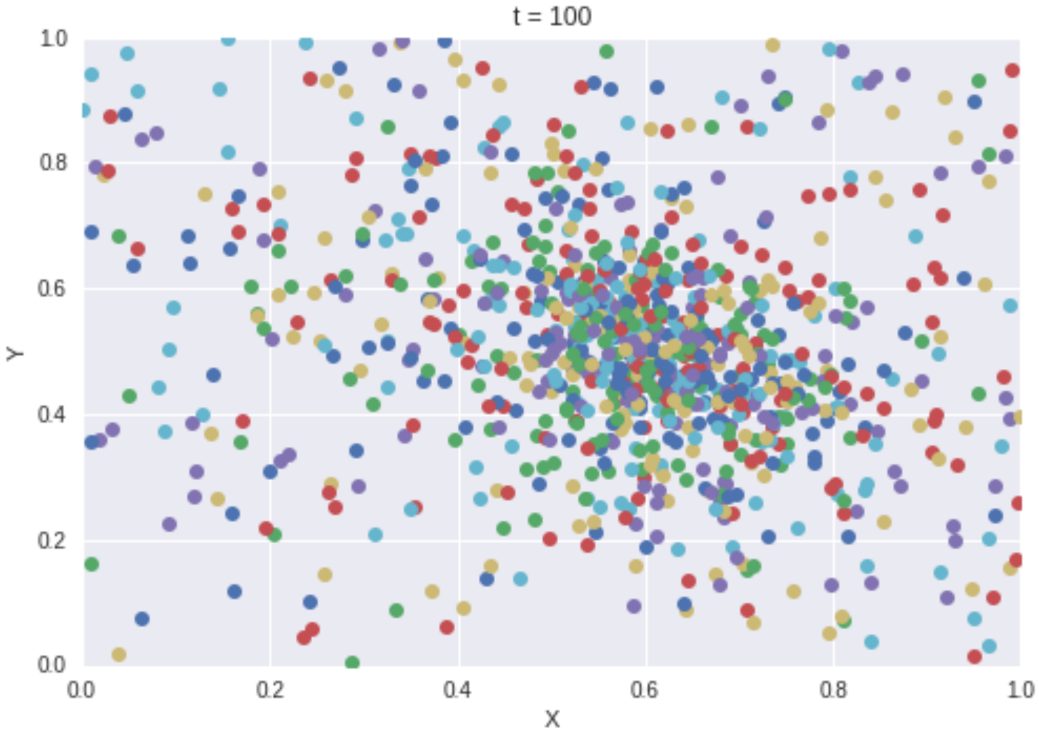
Sample task

2d simulation

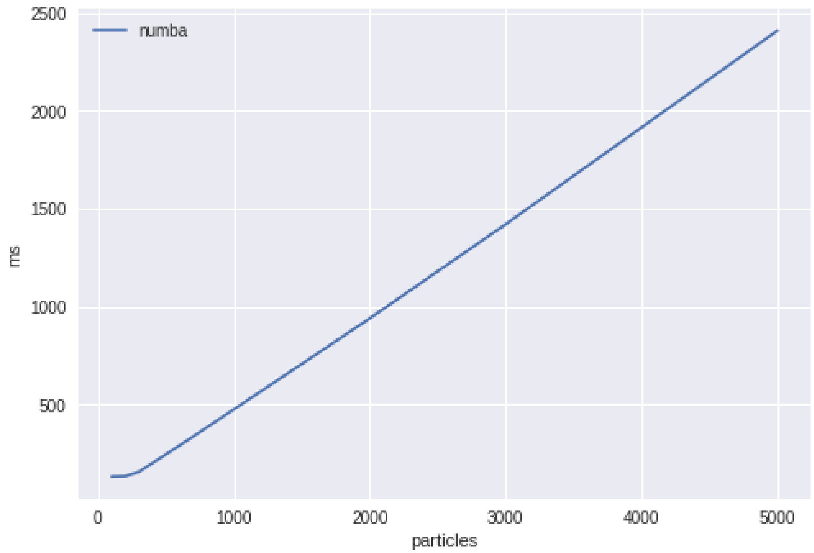
Pair interaction (n
PIC!)

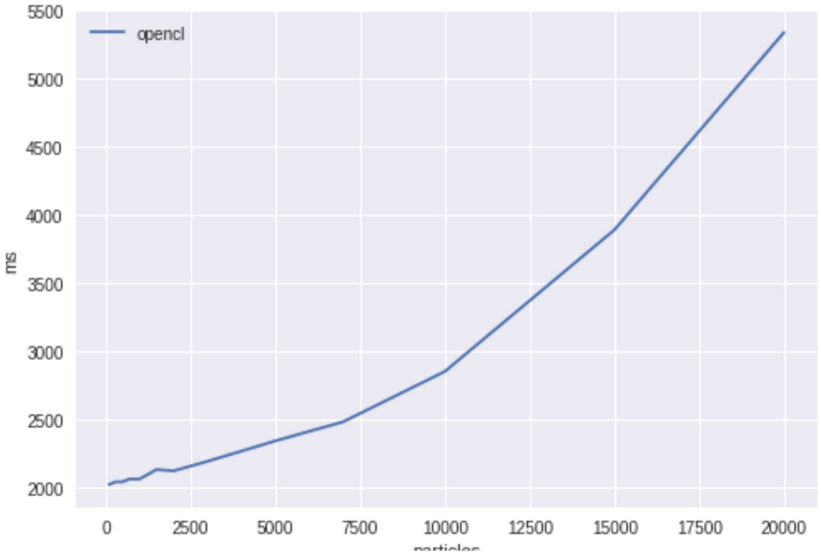
100 time steps

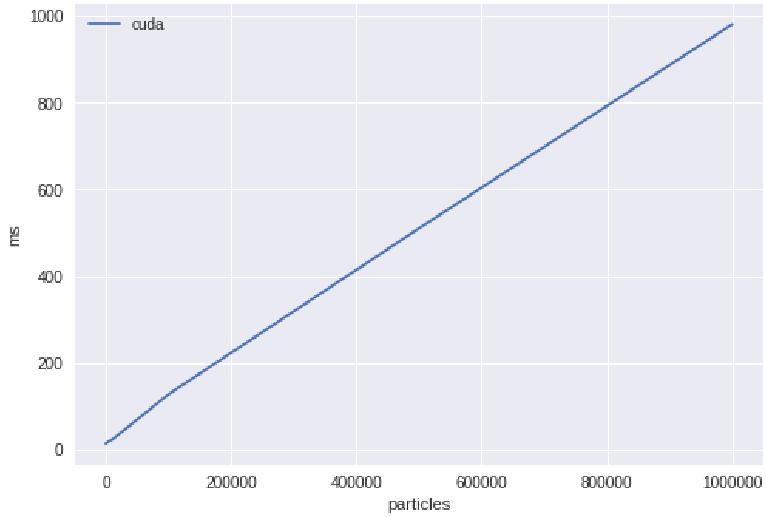
10 to 1 000 000 particles

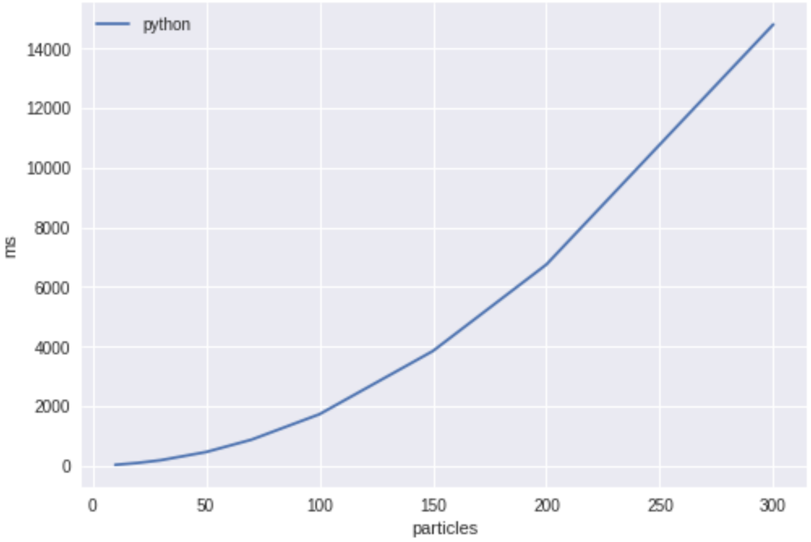


Results


numba
ms
2500
2000
1500
1000
500
0
1000
2000
3000
4000
5000
particles


opencl
ms
5500
5000
4500
4000
3500
3000
2500
2000
0
2500
5000
7500
10000
12500
15000
17500
20000


cuda
ms
1000
800
600
400
200
0
0
200000
400000
600000
800000
1000000
particles


python
ms
14000
12000
10000
8000
6000
4000
2000
0
0
50
100
150
200
250
300
particles

Conclusions

Performance is very dependant on library

- Todo: review implementation details
- Todo: implement and compare PIC

Google Colaboratory is convenient

- Can we test on local hardware
- Can we test on HybriLIT

GPUs are orders of magnitude faster

- Our problem can parallelize
- Implement it into the simulation project as soon as possible

Acknowledgements

This work was supported by RFBR grant 18-32-00239 мол_а