Università della Svizzera italiana

Facoltà di scienze informatiche

Particle Simulations with OpenACC: Speedup and Scaling

Overview of mathematical models, simulation used, and OpenACC

Samuel A. Cruz Alegría, Alessandra M. de Felice, Hrishikesh R. Gupta

(University of Lugano)

March 27, 2018

Status Update

Our tasks for this week were the following:

- Develop serial code.
- Investigate visualization tools.
- Investigate parallelization methods.

Serial Code

The serial code is divided into the following three main sections:

1. Tracing particles (trails or no trails).
2. Drawing particles.
3. Updating particle details such as position and velocity.

For the moment, particle movement doesn't strictly abide to any well-established physics. For instance, particle collisions with each other are not calculated yet.

Serial Code

Demo...

Visualization Tools

- For the time being, the simulation is done in two dimensions. This makes it relatively straightforward to paint the particles on the canvas.
- In three dimensions, we would need to add behaviour for the third dimension and would need to change the way in which particles are currently being drawn.
- In order to minimize time spent in building code for rendering in more than two dimensions, we can choose to use visualization tools.

Visualization Tools

An option for visualization is *ParaView*.

- Used at the CSCS (Swiss National Supercomputing Centre).
- Open source, used for visualizing two and three-dimensional data sets.
- Platforms supported range from single-processor workstations to multiple-processor distributed-memory supercomputers or workstation clusters.

References

- Farber, R., 2016.Parallel programming with OpenACC. Newnes.
- Gonzales, R., Martin, M., Mittow, N., and Rasmuss, R.,2016, An Introduction to OpenAcc.ECS 158 Final Project.
- Li, X., Shih, P.C., Overbey, J., Seals, C. and Lim, A., 2016. Comparing programmer productivity in OpenACC and CUDA: an empirical investigation.International Journal of Computer Science, Engineering and Applications (IJCSEA),6(5), pp.1-15.
- Memeti, S., Li, L., Pllana, S., Kołodziej, J. and Kessler, C., 2017, July. Benchmarking OpenCL, OpenACC, OpenMP, and CUDA: programming productivity, performance, and energy consumption. InProceedings of the 2017 Workshop on Adaptive Resource Management and Scheduling for Cloud Computing(pp. 1-6). ACM.
- Urbanic, J., 2013. Introduction to Directive Based Programming.
- OpenACC Programming and Best Practices. Guide `http://www.openacc.org/sites/default/files/inline-files/OpenACC_Programming_Guide_0.pdf`
- `http://www.nvidia.com/object/what-is-gpu-computing.html`
- Allen, M.P., 2004. Introduction to molecular dynamics simulation. Computational soft matter: from synthetic polymers to proteins, 23, pp.1-28.
- Eijkhout, V., 2014. Introduction to High Performance Scientific Computing. Lulu. com.