Special thanks to Radim Janalič and Juraj Kardos for all their help in this project.

The simulation of particle systems has become essential for visualizing the behavior of relevant physical systems where the computational complexity of performing simulations grows with the number of particles in the system.

Forces involved in this particle simulation are, in essence, a second-order ordinary differential equation (ODE), which can be stated as follows:

\[ F(x(t), \dot{x}(t), t) = m \ddot{x}(t) \]  

(1)

This has created a need to effectively speed up the performance of simulations in a simple, yet effective manner.

Given the relevance of parallel computation in particle simulations, our study aims to assess the scalability and performance of OpenACC parallelization in our model with an increasing number of particles.

The ICS cluster hosted 7 of these GPU nodes, which were used extensively to collect the results gathered in this project.

The NVIDIA GeForce GTX 1080 uses a Pascal architecture, has a frame buffer of 8GB GDDR5X, has 2560 NVIDIA CUDA cores, has a memory speed of 16GB/s, and has 3044 maximum threads per block.

The ICS cluster hosted 7 of those GPU nodes, which were used extensively to collect the results gathered in this project.

This study has been performed as a part of “Software Atelier: Simulation, Data Science, and Supercomputing 2018” course in the Università della Svizzera Italiana (USI).

The simulation of N-body particle simulation was executed on the ICS Cluster having CPU Intel X5-2650 and GPU NVIDIA GeForce GTX 1080, situated at the Institute of Computational Science, Università della Svizzera Italiana (USI). The NVIDIA GeForce GTX 1080 uses a Pascal architecture, has a frame buffer of 8GB GDDR5X, has 2560 NVIDIA CUDA cores, has a memory speed of 16GB/s, and has 3044 maximum threads per block.

The ICS cluster hosted 7 of those GPU nodes, which were used extensively to collect the results gathered in this project.

This study has been performed as a part of “Software Atelier: Simulation, Data Science, and Supercomputing 2018” course in the Università della Svizzera Italiana (USI).

The execution time for the update particles function remains constant with the varying number of time integration steps (N).

Approximately 127x speedup improvement in overall runtime occurs when the code is executed with OpenACC.

Special thanks to Radim Janalič and Juraj Kardos for all their help in this project.

The execution time for the update particles function remains constant with the varying number of time integration steps (N).

Approximately 127x speedup improvement in overall runtime occurs when the code is executed with OpenACC.

The performance was calculated by considering the execution time of the computationally most expensive unit, i.e., the update particle states function. The arithmetic intensity was computed as follows:

Floating point operations

\[ \frac{128}{529} = 0.29 \text{ Flop/Byte}. \]

The arithmetic intensity was computed by considering the most computationally expensive part, i.e., the update particle states function. The arithmetic intensity was computed as follows:

Floating point operations

\[ \frac{128}{529} = 0.29 \text{ Flop/Byte}. \]

The performance was calculated by considering the execution time of the computationally most expensive unit, i.e., the update particle states function and its double-precision floating-point operations, resulting in final performance of 0.015 TFlop/s.

The parallelization of N-body particle simulation was executed on the ICS Cluster having CPU Intel X5-2650 and GPU NVIDIA GeForce GTX 1080, situated at the Institute of Computational Science, Università della Svizzera Italiana (USI). The NVIDIA GeForce GTX 1080 uses a Pascal architecture, has a frame buffer of 8GB GDDR5X, has 2560 NVIDIA CUDA cores, has a memory speed of 16GB/s, and has 3044 maximum threads per block.

The ICS cluster hosted 7 of those GPU nodes, which were used extensively to collect the results gathered in this project.

This study has been performed as a part of “Software Atelier: Simulation, Data Science, and Supercomputing 2018” course in the Università della Svizzera Italiana (USI).

The execution time for the update particles function remains constant with the varying number of time integration steps (N).

Approximately 127x speedup improvement in overall runtime occurs when the code is executed with OpenACC.

The performance was calculated by considering the execution time of the computationally most expensive unit, i.e., the update particle states function. The arithmetic intensity was computed as follows:

Floating point operations

\[ \frac{128}{529} = 0.29 \text{ Flop/Byte}. \]