Parallel programming

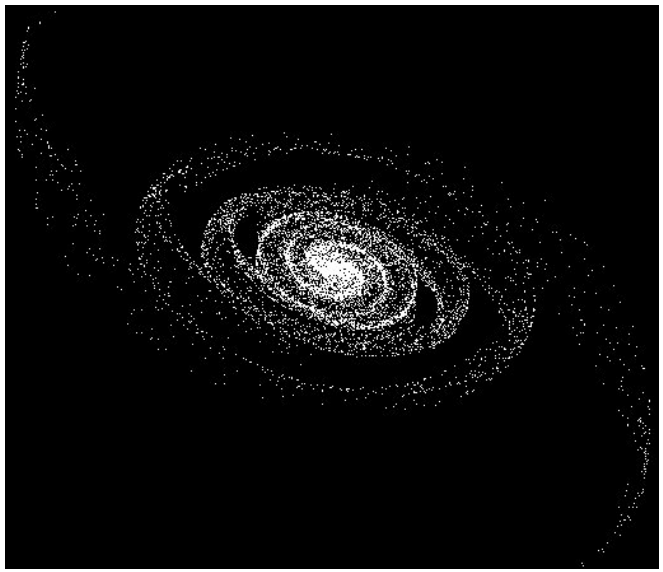
Semestral work
2D Gravitational N-body simulation

Libor Bukata a Jan Dvořák

FAKULTA
ELEKTROTECHNICKÁ
ČVUT V PRAZE

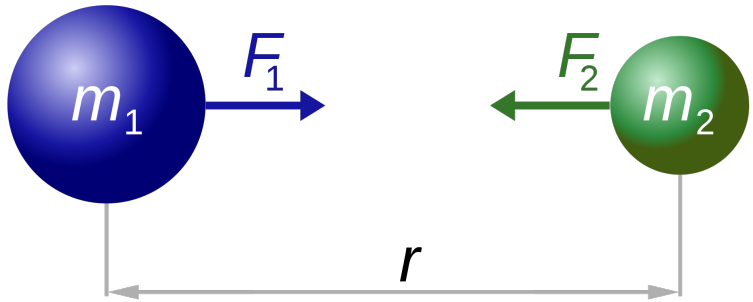
What is n-body simulation

- N-body simulation
  - simulation of a **dynamical** system of **particles**
  - usually under the influence of **physical forces**, such as gravity
  - **iterative** algorithm
    - computing **differences** between two discrete consecutive time instants



Hmm, Gravity... I do remember that from somewhere...

- Gravity
  - natural phenomenon by which all things with energy are brought toward
  - described by gravity force F
  - the force is applied in a given direction!
- Particle system
  - Each object (planet) is represented by one particle
  - Each particle interacts with each other



$$F_1 = F_2 = G\frac{m_1 \times m_2}{r^2}$$

$F = m \times a$
$s = vt + \frac{1}{2}at^2$

Input instance - example

```
3
10.235 18.654 33.55564
6.15 65.12 34.5064
198.654 0.215 15567.324
```

- 1. line – integer **number of stars** in the instance
- 2. line
  - **x-coordination** of the star (double)
  - **y-coordination** of the star (double)
  - **weight** of the star (double)
- You can use the provided generator

Semestral work assignment

- Task
  - Implement a sequential and parallel version of the algorithm for 2D Gravitational N-Body simulation
    - working on CPU/XeonPhi/Metacentrum
  - Export resulting simulation to gif
    - Simple library for gif creation can be found here: *https://github.com/ginsweater/gif-h*
  - Prepare the **presentation** to show achieved results

Evaluation of your semestral work

- Functional code (2 points)
  - Show an animation of the resulting simulation.
  - Code is tested by the teacher before the handover.
- Profile the bottlenecks (2 points)
  - Profiler outputs, detection of bottlenecks, analysis.
- Parallel code executed on Metacentrum + Xeon Phi (2 points)
  - Functional parallel code (C++11, OpenMP, OpenMPI).
  - Scalability and performance graphs and other performance metrics.
  - Metacentrum: PBS scripts, hardware info.
- Reproducibility of your experiments (2 points)
  - Benchmark scripts, description of used hardware and software.
  - A text file briefly describing how to carry out experiments.
- Presentation of your work (2 points)
  - Does your presentation satisfy formal requirements?
  - Is your presentation gripping to others (oral speech + slides)?
- General impression (4 points)
  - Have you used an algorithm with a better algorithmic complexity?
  - Is the performance of the optimized code outstanding?
  - Other aspects that may improve the general impression.

Requirements for your presentation

- **Base structure**:
  - Introduction
  - Profiling/bottleneck analysis
  - Scalability graph
  - Performance graph
  - Validity of the results
  - Discussion and conclusion
- The presentation should have at max **10 slides**.

Presentation - Introduction

- Describe the algorithm you used for simulation
- Mention the used technology

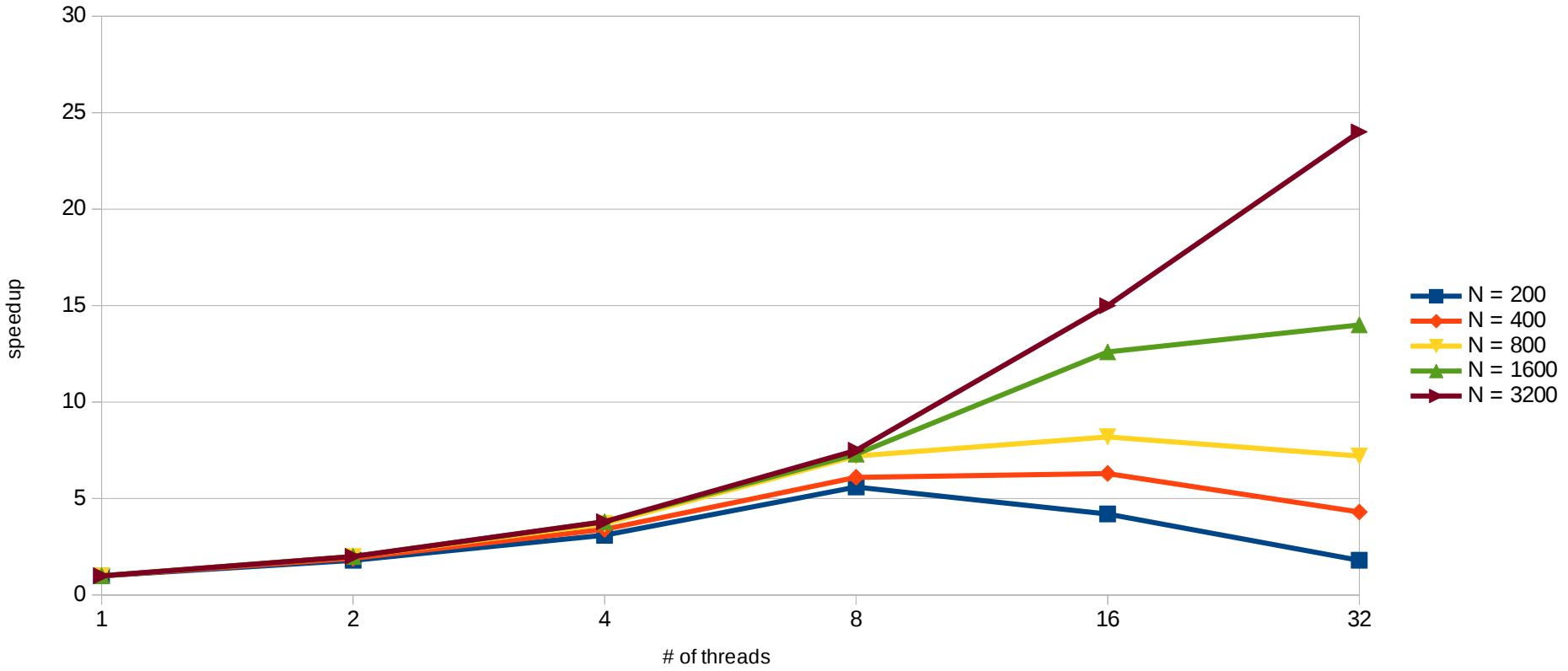
Presentation - Profiling

- Describe
  - The bottleneck of the sequential algorithm
  - Include a profiler output (graphs, excerpts from text files) indicating the bottleneck
  - The parts of the algorithm which is possible to parallelize
  - The way how you profiled the code (which tool you used)

Presentation – Scalability graphs

- Shows how the algorithm scales with respect to the number of used threads
- Scalability graph for up to 256 threads
- Each graph should have a title, legend, and an appropriate format of axes (+units).
- Description of the hardware and software.
- Graphs:
  - 1) Speedup of Parallel CPU version (e.g., Xeon Phi) vs Sequential version (scalability graph)
  - 2) Graph showing the algorithm runtime based on the size of an input instance (performance graph).

Presentation – Scalability graph

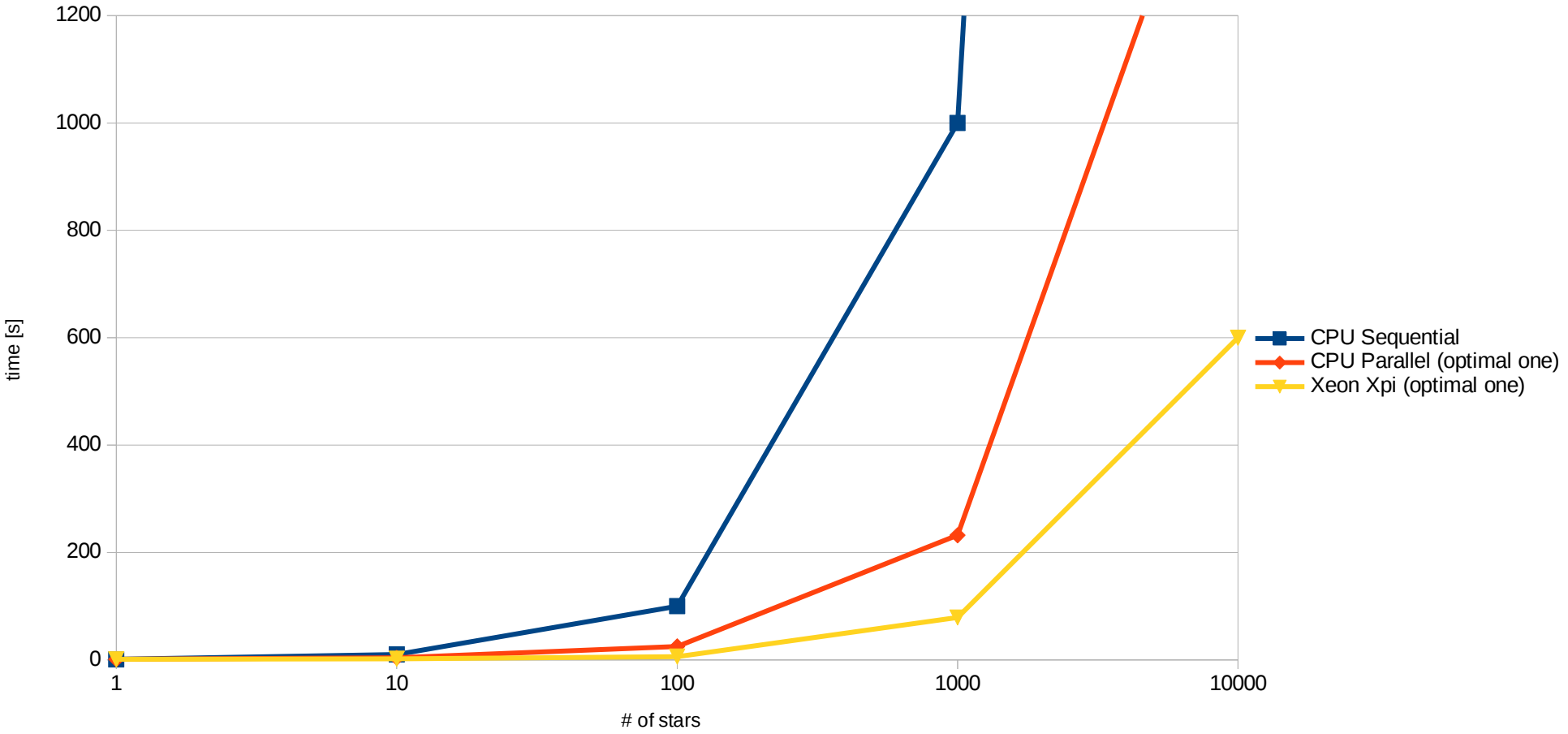


- N – number of stars
- For constant number of iterations (e.g. 100 iterations)

Presentation – Performance graph

- Shows how much time the algorithm takes to finish the computation depending on the number of stars, threads, ...
- Set the reasonable upper threshold for measuring (e.g., 5 minutes)

Presentation – Performance graph



Presentation

Validity of the results

- Show the resulting gif image.
- To be able to place the gif to the presentation shorten the animation to 5-10 seconds.

Discussion and conclusion

- Discuss the achieved results.
- Explain what was the most complicated part and why the results are as provided.
- What is the limiting factor of the parallelism in your algorithm.