

# High Performance CPU/GPU Multiresolution Poisson Solver

Wim M. van Rees, Babak Hejazialhosseini, Diego Rossinelli, Panagiotis Hadjidoukas and Petros Koumoutsakos  
Chair of Computational Science, ETH Zurich, Switzerland

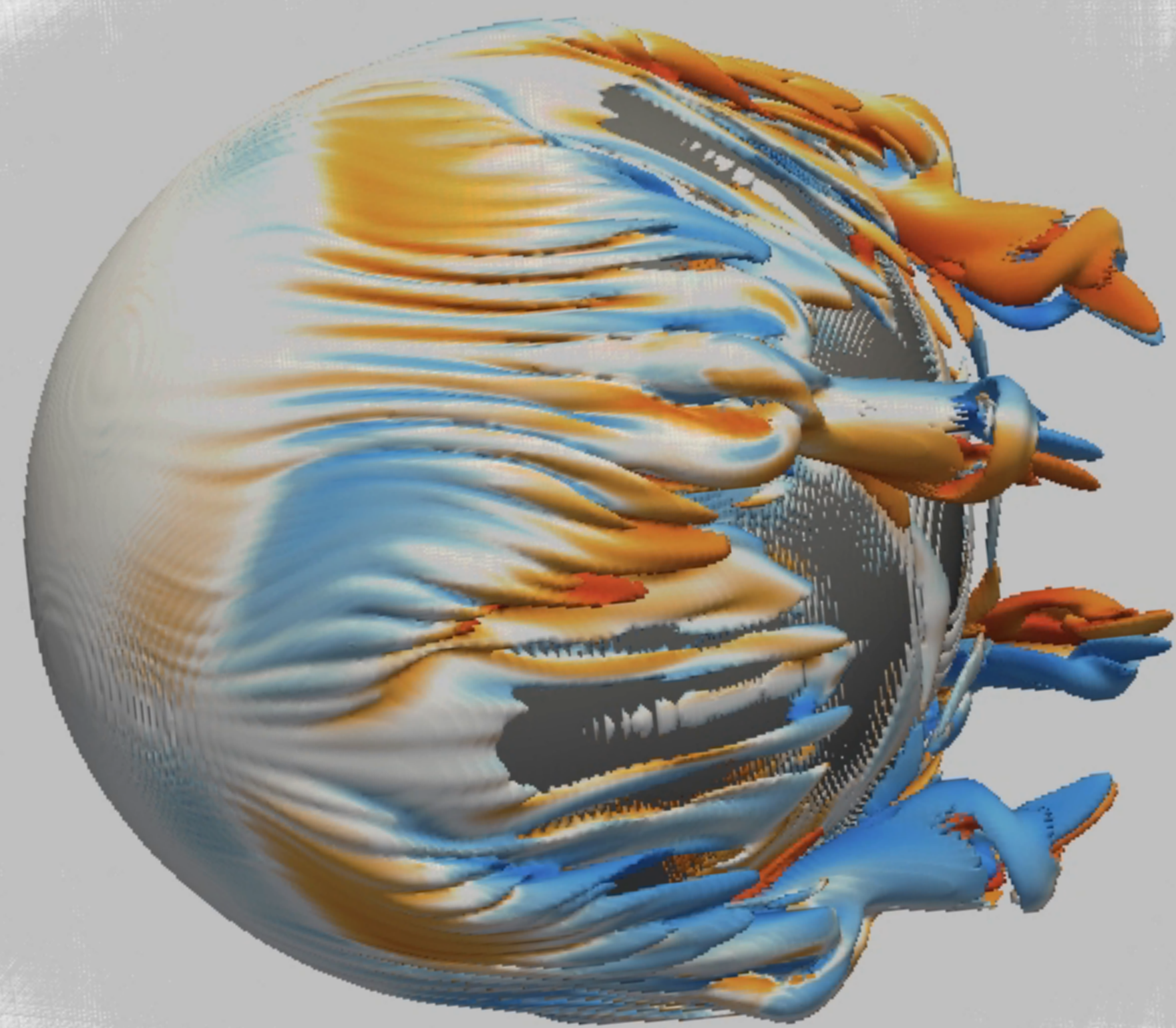
## Introduction

We simulate 3D incompressible bluff body flows using a wavelet adapted, remeshed vortex method. The Poisson equation is solved on *wavelet-adapted block-structured grids* using the *multipole method*.

$$\nabla^2 \vec{\Psi} = -\vec{\omega}$$

streamfunction      vorticity

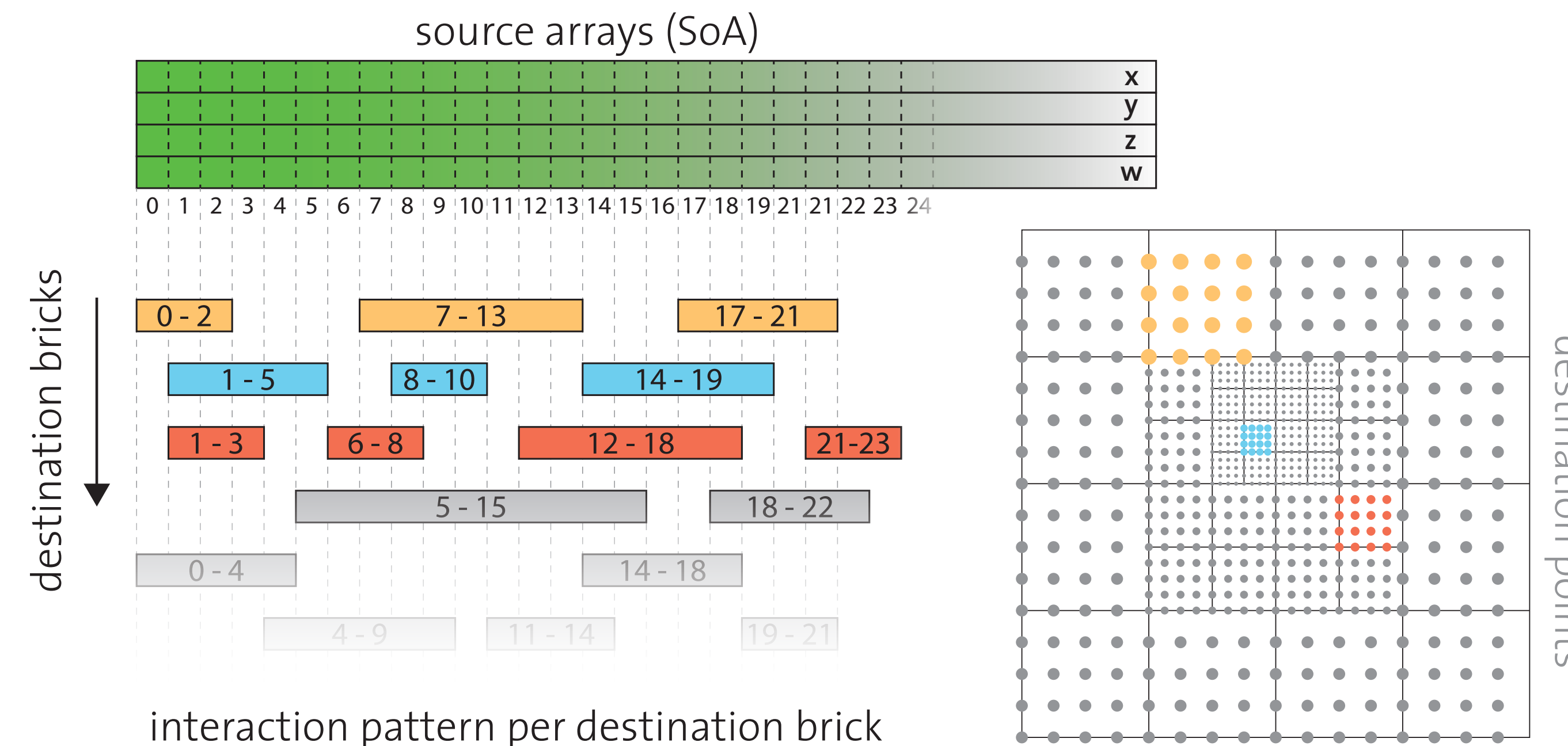
The *vorticity source points* have a compact support, whereas the *streamfunction* is computed on the entire grid. The velocity field is obtained by taking the curl of the streamfunction.



Flow past an impulsively started sphere at  $Re=5000$ , simulated using a remeshed vortex method

## Wavelet-Adapted Grids

The wavelet-adapted grid is composed of structured, non-overlapping blocks with a constant number of grid points, existing at different levels of resolution. We subdivide the grid blocks into smaller destination *bricks*. All the grid points in a destination brick share the same logical interaction patterns during the evaluation phase.

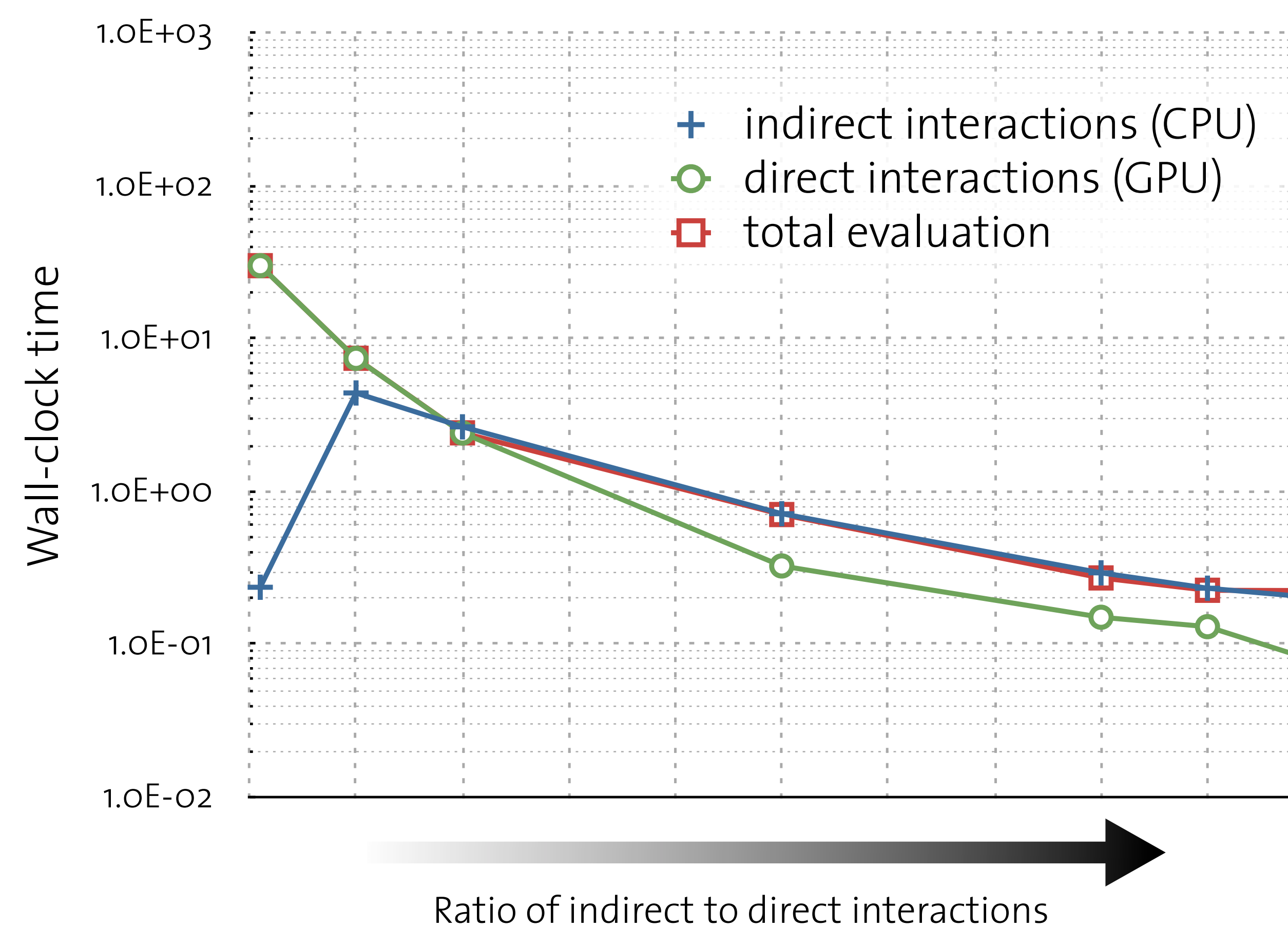


For the particle-particle interactions, destination points are grouped into bricks and evaluated according to a shared interaction pattern

## GPU implementation

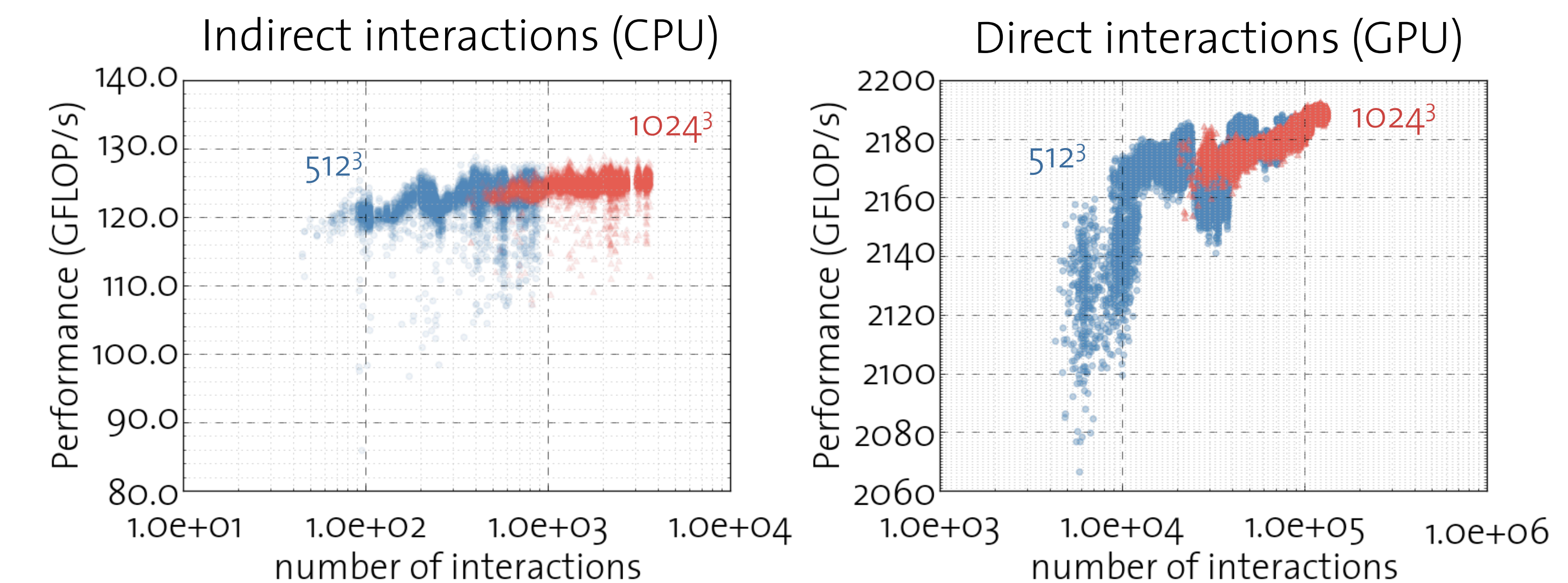
The multipole method distinguishes direct *particle-particle* and long-range *particle-multipole* interactions. The compute-intensive direct interactions are offloaded to the GPU.

Each destination brick is processed by one SM unit, so that each CUDA thread computes the interactions for one destination point. Simultaneously the multicore CPU evaluates the particle-multipole interactions. With this approach, the computation of the long-range interactions on the CPU determines the total evaluation time.



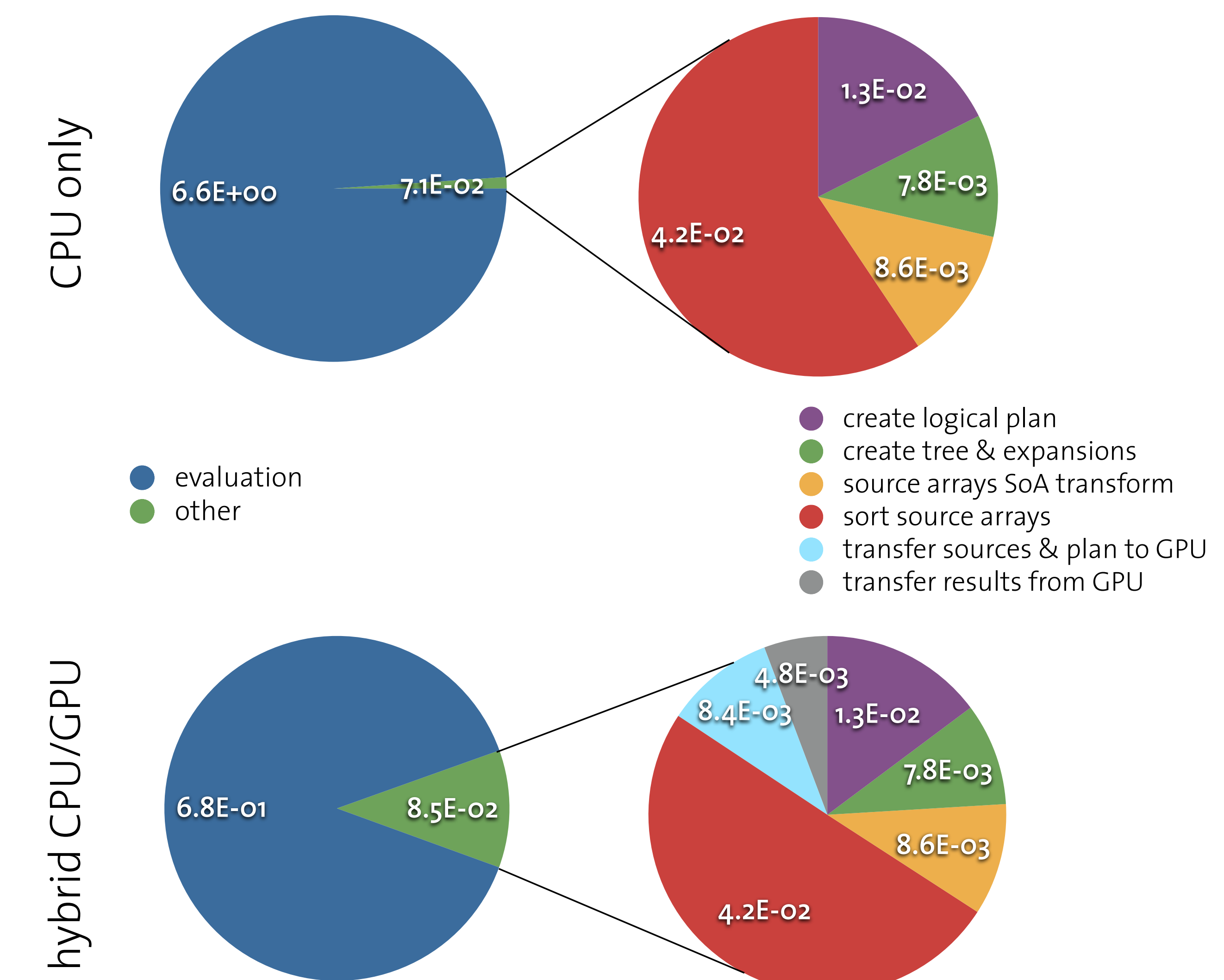
## Performance on Cray XK7

The software achieves **125 GFLOP/s** and **2.19 TFLOP/s** on the CPU and GPU, respectively. These correspond to 80% (CPU) and 97% (GPU) of the nominal peak performance, scaled according to the FMA instruction density of the kernels.



## CPU vs CPU/GPU

Data transfer to and from the GPU is mostly hidden by the computation on the CPU. Compared to a CPU-only evaluation using AVX/SSE intrinsics, a speedup factor of **10** is achieved.



## Acknowledgment

The authors thank Dr. Peter Messmer of NVIDIA for insightful discussions.