Eiron: Short status

Oskar Lappi

November 2024

Status at a glance: software development

- Eiron: software development phase done
  - Domain decomposition implemented
  - Pipeline for running simulations/benchmarks on Mahti
  - Public dashboard for looking at benchmark results
  - Ready to release version 0.1.0
- KDMC prototype: implemented
  - New kinetic-diffusive simulation scheme
  - Simple time stepped estimation scheme
  - Globally constant BGK operator for testing purposes
  - WIP: Thijs Steel is working on a fluid estimator for the KDMC scheme

Status at a glance: what's missing

- Eiron: missing features
  - Smart load balancing
  - Different memory layouts for grids
  - Some charon parameters can't be tuned at runtime

I think these can be left out of scope from the first paper, and we can come back to them later (unless we really really need them in the performance paper).

Status at a glance: research

- Eiron: performance study started
    - Only started seriously benchmarking last week
    - Memory effects in domain decomposition identified
    - Some issues with scalability, will present on this in main Eiron presentation
- KDMC prototype: comparisons started
    - We've run some new experiments, and feel like the results are promising
    - Still need to run bigger experiments
    - The isotropic rotation scattering might be an issue
    - We therefore also want to run new experiments with a BGK operator

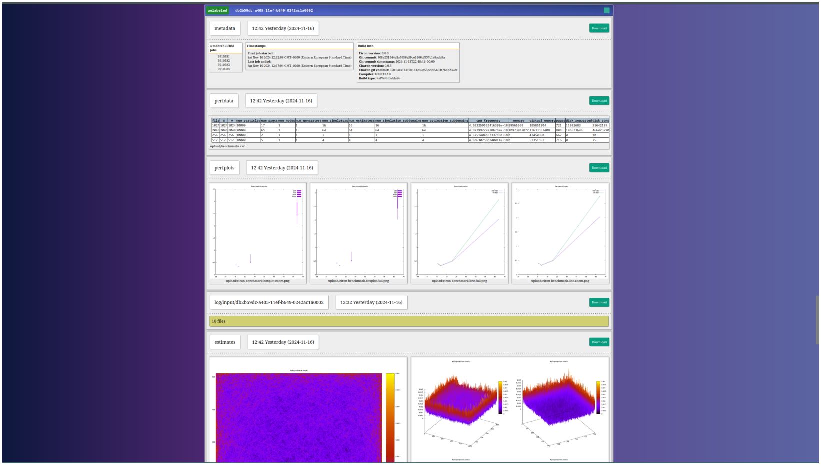
Research pipeline

To make life easier for myself, I've set up a rough research pipeline to run these experiments.

- I upload an experiment to CSC's cloud storage
- The experiment is scheduled on CSC's supercomputers
- The results are uploaded to cloud storage
- A public web dashboard displays the results

Research pipeline dashboard

The dashboard lets me check on multiple jobs and compare them, and also allows me to share results quickly with collaborators.



unlabeled db2b59dc-a405-11ef-b649-0242ac1a0002

metadata 12:42 Yesterday (2024-11-16) Download

4 mahti SLURM jobs
3910581
3910582
3910583
3910584

Timestamps
First job started: Sat Nov 16 2024 12:32:08 GMT+0200 (Eastern European Standard Time)
Last job ended: Sat Nov 16 2024 12:37:04 GMT+0200 (Eastern European Standard Time)

Build info
Eiron version: 0.0.0
Git commit: 9f8a231944e5a5656e59ce1966cf837c1e8ada8a
Git commit timestamp: 2024-11-13T22:48:41+00:00
Charon version: 0.0.3
Charon git commit: 5503983373190144259b55ec095634f76ab2328f
Compiler: GNU 13.1.0
Build type: RelWithDebInfo

perfdata 12:42 Yesterday (2024-11-16) Download

file	x	y	num_particles	num_procs	num_nodes	num_generators	num_simulators	num_estimators	num_simulation_subdomains	num_estimation_subdomains	cpu_frequency	memory	virtual_memory	pages	disk_requested	disk_cons
1024	1024	1024	10000	17	1	1	16	16	16	16	4.693259533416399e+18	99565568	185851904	721	11023683	21642125
2048	2048	2048	10000	65	1	1	64	64	64	64	4.693992297786769e+18	10973007872	11633553408	808	146523646	466423290
256	256	256	10000	2	1	1	1	1	1	1	4.675140483733783e+18	0	43450368	662	0	10
512	512	512	10000	5	1	1	4	4	4	4	4.686302588340011e+18	0	51351552	716	0	25

upload/benchmarks.csv

perfplots 12:42 Yesterday (2024-11-16) Download

upload/eiron-benchmark.boxplot.zoom.png

upload/eiron-benchmark.boxplot.full.png

upload/eiron-benchmark.line.full.png

upload/eiron-benchmark.line.zoom.png

log/input/db2b59dc-a405-11ef-b649-0242ac1a0002 12:32 Yesterday (2024-11-16) Download

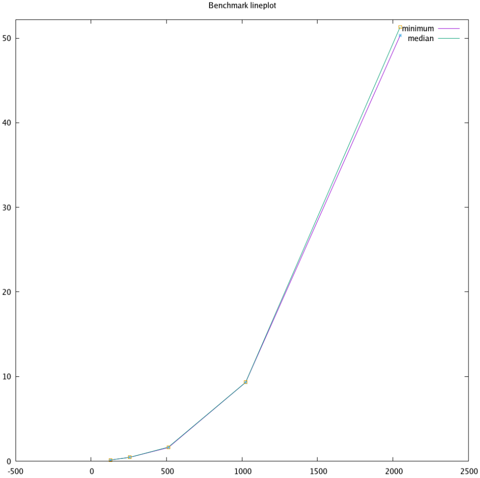
18 files

estimates 12:42 Yesterday (2024-11-16) Download

Selected plots (Y-axis is runtime)

Serial performance: box size effect

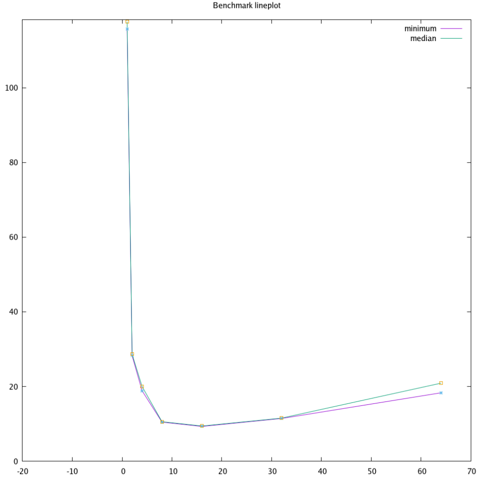
Increase box size in steps of 4x: 128^2, 256^2, 512^2, 1024^2, 2048^2,



The runtime increase is ∼3.5x for the first two steps, ∼5.5x for the last two.

2048x2048 grid: domain decomposition effect

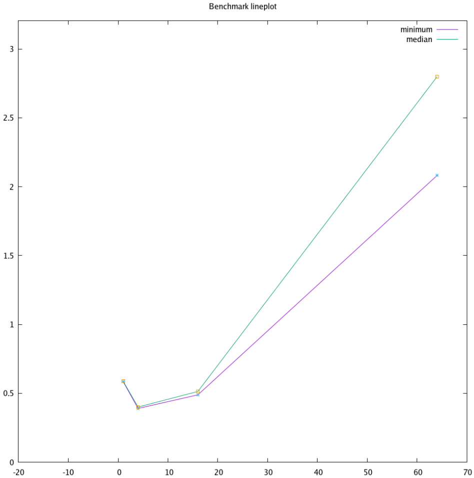
All points on this graph use the same amount of memory and cores, but a different number of subdomains.



4x4 subdomains of size 512^2 is best. One 2048^2 domain 12x slower.

Performance study: weak scaling

Weak scaling is causing some trouble.



Up to 8 or 16 procs, runtime is flattish, but then we slow down. Most likely a communication cost.

The plan

Now it's a matter of tuning the performance parameters in charon and finding a case that scales. The phase space of the performance configuration is quite large, and I've only started exploring it.

After that, we'll design a few scaling experiments that we can publish, comparing OpenMP paralellization and MPI parallelization.

The experiments can be replicated and shared easily, which should make reviewing the paper easier.

We're now at the stage where the paper will be written as soon as I'm happy with the results.

Thank you

Questions/Comments

Live dashboard