

Performance and Reliability of VSTRAP – a Snapshot

28th Spacecraft Plasma Interaction Network in Europe meeting 2021

June 8th - 10th 2021

R. Bouziane, S. Rouwette, U. Siems, L. Basov, S. Chibani, B. Mockel, F. Zhu, R. Schmit, D. Petkow,

F. Taccogna,

D. Feili, and G. Deprez

Agenda



Software Design

Reliability

Performance

Outlook

Software Design | Nutshell



Modular programming

Objective: ease of addition (new solvers, new features, user requirements, etc.).

Agile : efficient development cycles within a team.

Maintainability : **efficient** refactoring enhanced by CI & modular framework

Reusability : less redundant code.



Example of the modular implementation structure

Reliability | Objective & Ingredients



High quality code.

- Tools: GIT, CI, different compilers, code analyses (static and dynamic),...
- Rules: Branching system, code reviews, change logs, code styling,...
- Tests: Unit tests & Regression tests,...
- Metrics: Unit test code coverage, s/w metrication of code reliability,...

Reliability | Objective & Ingredients



High quality code.

- Tools: GIT, CI, different compilers, code analyses (static and dynamic),...
- Rules: Branching system, code reviews, change logs, code styling,...
- Tests: Unit tests & Regression tests,...
- Metrics: Unit test code coverage, s/w metrication of code reliability,...

Reliability | Tools & Rules



- CI Continuous Integration
- GITlab: Code versioning, bug tracking, ...
- Branching system (master branch, development branch, feature branches)
- 2 important rules:
 - 1. The master branch is the production branch and must always be deployable
 - 2. Feature branches are rebased & merged during development only once test suites in CI pass successfully.
- Code review for every change in a feature branch (assigned responsibilities)
- Change logs (condensed change info going way beyond (technical) code commit info)
- Event-triggered test cases: Unit tests + regression tests executed on each branch at each push operation
- Time-triggered test cases: compilation, unit tests, integration tests and regression tests

Reliability | Metrics



Unit test code coverage

- Automated weekly reports sent by Email
- CPU code: Quantification using Lines of Code & Number of Functions
- GUI code: Quantification using Lines of Code

Reliability | Final Touch



Deployment

- Containered with Docker & Singularity
- VSTRAP is stable *and* independent of user's system configuration
- Guaranteed user-friendly: after installing Docker or Singularity, just execute VSTRAP

Performance | Two Physics Engines



VSTRAP comes with two feature-identical physics engines

- CPU-only code: OMP parallelized, i.e., shared memory machines
- CPU/GPU code:
- Parts of CPU-only code are GPU-parallelized for CUDA-compatible GPUs (NVIDIA)
- OMP for the remaining CPU-only code

Reason:

- CPU for long serial processes | Few high frequency cores | High memory size.
- GPU for high parallel processes | Many lower frequency cores | High local memory throughput.

Performance | CPU-only Physics Engine



Optimization strategy

- Profiling to identify bottlenecks
- Increase performance in two ways:
 - Optimize algorithms
 - OMP parallelization
- Measure progress by set of test cases

How do we select the code parts / algorithms for GPU parallelization?

Performance | CPU/GPU Physics Engine



Identification of GPU-parallelizable Engine code

- Computational expensive
- Little data transfer required
- NVIDIA GPU: CUDA framework. How does it work?









Porting Process

- Define device data that are required for GPU kernels
- Study the transfer overhead (minimize data transfer for better performance gain)
- Make device memory allocations
- Setup the device data on the host or on the device
- Define the GPU kernel and configure it with the proper execution configurations:
 - number of blocks
 - number of threads per blocks
- Study the synchronization between host and device
- After the GPU kernel call and synchronization, copy the result from device to host
- Unit testing and evaluating performance



Performance gains (today) just by porting to GPU

• CPU : Intel(R) Xeon(R) CPU E5-2697 v2 @ 2.70GHz

Tesla V100 :

- Speedup factor of 5 for FMM
- Speedup factor of 3.5 for 2D pusher
- Speedup factor of 2 for 2D PIC



Optimization strategy

Roofline Analysis : NVIDIA Nsight compute:

- Arithmetic intensity (FLOP/byte)
- Performance (FLOP/s)





Roofline analysis memory statistics

Memory statistic -L1/L2 caches Memory access patterns -uncoalesced access



VSTRAP | Outlook



Reliability

- Extend s/w metrication
- Add unit test GPU-code coverage

Performance

Further increase by

- GPU Algorithm optimization
- Data transfer between host and device
- Hardware: NVIDIA A100 GPU 40MB + Utilize tensor cores
- Multi-GPU utilization



Product available 06/22.

Beta available now.

If you want to influence further VSTRAP development – contact us.

SPARC Industries

We deliver.The future.



www.sparc-industries.com info@sparc-industries.com